

Um Modelo Adaptativo Aplicado no Aprimoramento da Representação de Linhas Retas em Geometria Digital

Leoncio C. Barros Neto, André H. Hirakawa e Antonio M. A. Massola

Resumo—Apesar de representações digitais fundamentadas em linhas retas e arcos serem ferramentas de muitos estudos em ciência da computação, o seu uso não é tão popular em cenários que requerem estruturas flexíveis capazes de responder às imprecisões dos modelos ou às condições variáveis do ambiente. O objetivo deste trabalho é investigar uma proposta baseada na adaptatividade por meio do dispositivo denominado segmento digitalizado adaptativo (SLRDA), que incorpore o poder expressivo de representar segmentos em linha reta digitalizada (SLRD). Assim, partindo-se de primitivas, são concebidas as funções adaptativas correspondentes relacionadas a arcos digitais pelos quais *strings* estimulam, em apenas uma passada, a reação de determinado SLRDA conforme, por exemplo, as tolerâncias dos parâmetros envolvidos. Consequentemente, o método proposto torna-se uma alternativa versátil, relativamente simples e intuitiva às abordagens existentes, apresentando capacidade de aprendizagem, além de ser computacionalmente poderoso.

Palavras Chaves— Computação reconfigurável, Geometria e modelagem computacional, Reconhecimento de padrões, Teoria dos autômatos, Erro (Recuperação)

I. INTRODUÇÃO

PARTINDO de estudos na área de construção de compiladores por [1], surgiram aplicações da adaptatividade nas mais diversas áreas que resultaram em todo um acervo de recursos teóricos, conceituais e de ferramenta em aplicações variadas que foram sendo acumuladas, descritas nos levantamentos de [2] e [3]. No entanto, a despeito da relevância da representação computacional de linhas retas e arcos, inclusive sendo uma área ativa de pesquisas há quase meio século, citado por [4], esse assunto ainda não foi estudado sob o enfoque adaptativo com o potencial do mencionado acervo, até o momento.

Assim como o conceito de linha reta é importante na geometria Euclidiana, é dos objetos considerados fundamentais em computação, com vários pesquisadores tendo se interessando por esse assunto, pois, apesar da aparente simplicidade das linhas digitais, estas incorporam todas as dessemelhanças e disparidades entre o discreto e o contínuo.

No processo de digitalização, é inevitável que determinado segmento de reta contínuo no espaço Euclidiano seja afetado por transformações, deformações ou corrupção por ruído gerando um uma *string* com imperfeições (não seguindo o modelo ideal, afetada por erros).

“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” [5]. Os dispositivos adaptativos apresentam naturalmente a flexibilidade requerida para atuarem em cenários dinâmicos, por serem capazes de responder às variações nas condições ou situação momentânea do ambiente.

A hipótese inicial deste trabalho é quanto à viabilidade em modelar propriedades das retas digitais em imagens digitalizadas por um conjunto de regras a fim de aplicar a adaptatividade. Adicionalmente, caso as interferências espúrias possam também ser modeladas, mesmo que indiretamente por intermédio de tolerâncias permitidas, passa a ser viável que um dispositivo adaptativo represente as diferentes instâncias possíveis das estruturas.

Face ao exposto, esta pesquisa tem como principal objetivo investigar a representação de segmentos de linhas retas digitalizadas (SLRD¹¹) por meio da adaptatividade, cotejando com outras formas de representação ao mostrar aspectos positivos e negativos comparativamente.

O conceito de “linhas retas digitais adaptativas¹²” desta proposta utiliza o Autômato Finito Adaptativo (AFA), abrangendo incorporar a capacidade de representar parâmetros tais como as tolerâncias, a escalabilidade, os desvios em ângulo ou os desvios em comprimento dos mencionados segmentos em linha reta por intermédio de segmentos digitalizados adaptativos (SLRDA). Pela linearidade adaptativa, determinada *string*, relacionada a um arco digital qualquer, pode corresponder a um SLRD, mesmo que aproximado, após avaliação de parâmetros do arco em diversas escalas. Por outro lado, arcos que não corresponderem a retas, são representáveis pela concatenação de SLRD [7].

A. Fundamentos sobre Segmentos Digitalizados

Este tópico apresenta conceitos básicos sobre SLRD.

Entendendo-se pontos digitais como pontos cujas

¹¹ As siglas deste trabalho indicam tanto o singular como o plural do que se referem, com cada significado a ser interpretado em adequação com o contexto.

¹² [6] observa que a diferença entre linha reta digitalizada e linha reta digital é sutil. Entende-se que a primeira é o resultado da digitalização de uma reta Euclidiana específica. O foco principal desta pesquisa é quanto a retas digitalizadas, mas se aplica também ao caso de retas digitais.

coordenadas são inteiras resultantes de um esquema de digitalização; um arco digital qualquer S é um conjunto de pontos digitais interligados, componentes de uma imagem digital. A característica do conjunto de pontos pertencentes ao arco S é o seu posicionamento relativamente a uma grade, tal que cada ponto desse conjunto tem exatamente apenas dois vizinhos; exceto dois pontos, denominados extremos, que possuem apenas um vizinho em S [9].

Em [8] o chain code foi apresentado como um descritor de contornos, de apenas um pixel de espessura, e um modelo definidor de retas digitais foi também conjecturado por Freeman. Nesse modelo, dado um ponto digital, as vizinhanças principais e imediatas desse ponto estão mostradas na Fig. 1, que mostra o relacionamento dos símbolos do chain code com a vizinhança-4 e vizinhança-8.

Após seleção da vizinhança-4 ou vizinhança-8, o chain code é definido da seguinte maneira:

Definição 1: O chain code, ou o código da cadeia, é uma sequência de elementos com finalidade de representar um arco digital, onde cada elemento é um símbolo na vizinhança selecionada, que representa o vetor unindo dois pontos digitais vizinhos do arco em questão.

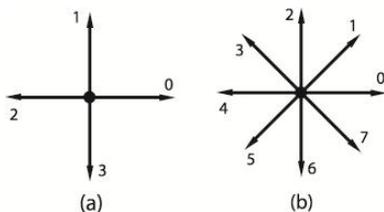


Fig. 1 À esquerda estão os símbolos de 0 a 3 do chain code para vizinhança-4. À direita, os símbolos de 0 a 7 do chain code para vizinhança-8.

Um arco digital é reto se for o resultado da digitalização de um segmento em linha reta. No modelo de Freeman, as *strings* que representam linhas retas obedecem a três propriedades, em uma codificação utilizando números de 0 até 7, em vizinhança-8:

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

--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 ocorre em grupos ou corridas (runs).

O significado de Prop1 a Prop3 é representar a linha por uma sequência de vetores com inclinações múltiplas de 45° e cujos comprimentos são unitários (se horizontal ou vertical) ou $\sqrt{2}$ (se diagonal).

Uma das questões fundamentais desta pesquisa é o poder computacional requerido para a análise sintática de SLRD.

B. Análise Sintática de Segmentos Digitalizados

Um dos procedimentos do modelo sintático inicia pela definição de uma gramática associada a algum tipo de dispositivo de reconhecimento [10]. Esse reconhecedor é

denominado parser por ser responsável pela análise sintática, decidindo (parsing) se uma dada *string* observada pertence ou não à classe representada pela gramática.

Porém, ruído e distorções complicam o processo computacional da análise sintática: além das distorções, primitivas espúrias são geradas, bem como primitivas reais podem não ser detectadas. Além disso, a própria natureza variável dos períodos dos símbolos dos SLRD, associado a comprimentos variáveis da escala das retas é um desafio para efetuar a análise sintática de SLRD. Dependências de contexto e alterações no ângulo de orientação em segmentos de comprimento arbitrário afetam a estrutura dos códigos das retas digitais forçando o parser a rever a sua análise [11]. 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 [12], as linguagens são classificadas em quatro classes diferentes: 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.

Entre as abordagens de pesquisa existentes, os métodos sintáticos são normalmente considerados pela literatura como não sendo convenientes para tarefas envolvendo SLRD. Isso porque a análise sintática de SLRD requer gramáticas poderosas, sensíveis ao contexto, inviabilizando a aplicação de formalismos simples, tais como AF [13] [4]. Relembre-se que uma Linguagem Regular é especificada por uma Gramática Regular. Os conceitos de Linguagem Regular e Autômato Finito (AF) são equivalentes no sentido de que para cada Linguagem Regular existe, pelo menos, um AF que a reconhece e vice-versa (ver [12] para o formalismo de gramáticas, linguagens e autômatos).

A maneira que o AFA aceita linguagens tipo 1 e 0 é apresentado por [14] pelo recurso do AFA alterar seu próprio conjunto de estados e de regras de transição [14].

Em princípio, a linguagem tipo 0 não faria parte do escopo desta pesquisa. Entretanto, visando estar de acordo com os comprimentos dos SLRD, os quais podem estar em várias escalas, teoricamente até infinito, implica em linguagens tipo 0 neste trabalho.

C. Codificação

Nesta pesquisa, os SLRD são codificados por meio de *strings* de caracteres, e não de números, bastando considerar $a = 1$ e $b = 0$ no segmento exemplificado pela Fig. 2 a fim de atender Prop1. Uma *string* é uma sequência de zero ou mais símbolos pertencentes ao alfabeto Σ . Esses símbolos são também denominados como primitivas, tokens, elementos do chain code ou simplesmente estímulos. O conjunto de todas as *strings* pertencentes a Σ é denotado por Σ^* . O comprimento de uma *string* qualquer S é denotado por $|S|$. A *string* vazia, de

comprimento zero, é representada por ϵ . O i -ésimo símbolo de uma *string* $S = s_1 \dots s_n$ é representado por s_i .

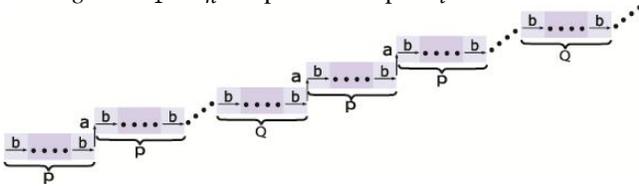


Fig. 2 Modelo de SLRD genérico no primeiro quadrante, com corridas de P e Q símbolos b , tão uniformemente espaçados quanto possível entre tokens a , que ocorrem isolados.

Caso nada em contrário seja especificado, sem qualquer redução em generalidade, neste trabalho é utilizada a vizinhança-4 como padrão, tal que os símbolos da propriedade Prop1 devem ser consecutivos, módulo quatro. Mais precisamente, os símbolos que compõem as *strings* pertencem a $\Sigma = \{a, b, c, d\}$ tal que, para atender Prop1, basta considerar módulo 4, juntamente com $a=1, b=0, c=3, d=2$, relativamente à vizinhança-4 da Fig. 1.

Comparando-se a vizinhança-4 com outras, por exemplo, vizinhança-8, a primeira apresenta vantagens como simplicidade nas implementações, além dos algoritmos de processamento digital considerarem normalmente apenas dois eixos x e y , porém tem a desvantagem de maior perda de informação. A escolha envolve também o aspecto da notação mais conveniente, pois autores utilizam uma e outra, por exemplo [15] e [16]. Uma *string* qualquer $S = s_1 \dots s_n$ pode também ser representada no formato da Expressão 1.

$$S: s_i; i = 1, 2, \dots, n.$$

(1)

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

As estruturas adaptativas deste texto são do primeiro quadrante. Caso seja necessário incorporar segmentos de outros quadrantes a um determinado SLRDA representativo de apenas um quadrante, basta manter a topologia inicial, alterando os símbolos convenientemente, obtendo o SLRDA homólogo do primeiro para o outro quadrante.

D. Organização do Texto

A seção II apresenta fundamentos sobre AFA. A seção III apresenta os trabalhos principais que embasam conceitualmente esta pesquisa. Na seção IV, avaliam-se os SLRD partindo-se de suas propriedades de ângulo de orientação e comprimento de segmentos. A seção V apresenta as estruturas que estão sendo propostas por esta pesquisa. A seção VI apresenta uma síntese dos conceitos principais envolvidos na proposta, bem como posiciona este trabalho com relação ao estado-da-arte ao mostrar suas vantagens mais relevantes. A seção VII apresenta as considerações finais deste relato, contribuições e sugestões de trabalhos futuros.

II. AUTÔMATO FINITO ADAPTATIVO (AFA)

A Definição 2 facilita as descrições dos autômatos desta pesquisa.

Definição 2: Uma 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. Na terminologia sintática esse conjunto é denominado como do tipo “head-to-tail concatenation”, de acordo com a página 3 de [17].

A Definição 3 apresenta o AFA.

Definição 3: O Autômato Finito Adaptativo (AFA) é um dispositivo auto-modificável da forma $AFA = (ND0, AM)$ onde ND0 é o mecanismo subjacente, e AM é a camada adaptativa associada, formalizada nos mesmos moldes que ND0. O AFA é caracterizado por ter o autômato finito (AF) como mecanismo subjacente.

A camada adaptativa da Definição 3 integra o conjunto de operações de auto modificação possíveis, denominadas ações adaptativas, entendidas como atividades que são executadas em resposta aos estímulos, alterando a configuração do dispositivo.

Definição 4: Ações adaptativas são chamadas paramétricas de funções adaptativas (FAD), que podem ser considerada como coleções de ações adaptativas elementares a serem aplicadas para definir as transições do autômato.

Um AFA é representado pela 7-tupla da Expressão 2.

$$AFA = (Q; AR_0; \Sigma; q_0; F; \mathcal{B}; \mathcal{A}) \quad (2)$$

tal que:

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

AR_0 é um conjunto de regras adaptativas:

$AR_0 \subset \mathcal{B} \times (Q \times (\Sigma \cup \{\epsilon\})) \times \mathcal{A}$, sendo que

$(Q \times (\Sigma \cup \{\epsilon\}))$ corresponde ao conjunto de regras δ do dispositivo subjacente;

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 ações adaptativas anteriores e posteriores, respectivamente. O conjunto \mathcal{B} é denominado anterior por ser executado antes da transição entre estados; enquanto o conjunto posterior \mathcal{A} é executado após a transição.

Por sua vez, as ações adaptativas componentes dos conjuntos \mathcal{B} e \mathcal{A} são formadas por conjuntos de ações adaptativas elementares ou primitivas a serem aplicadas para definir o processo adaptativo do autômato. Tais ações adaptativas elementares podem ser de três modalidades, de acordo com o prefixo representado na Tabela I, em que o padrão especifica uma transição em relação à qual as transições em uso devem ser testadas:

- 1) Ações de busca ou consulta, denotadas pelo prefixo “?”: efetuam uma busca por regras que casem com o padrão especificado, sem alterar o conjunto de regras;
- 2) Ações de remoção, denotadas pelo prefixo “-”: removem do conjunto as regras que casam com o padrão especificado;
- 3) Ações de inserção, denotadas pelo prefixo “+”: inserem o padrão especificado no conjunto de regras.

A Tabela I mostra as três modalidades de ações adaptativas

elementares ou primitivas com o padrão especificado entre colchetes. A maneira de especificar esse padrão é apresentada no tópico seguinte que mostra o formato das ações adaptativas elementares.

TABELA I
SIMBOLOGIA PARA AS AÇÕES ADAPTATIVAS PRIMITIVAS

PREFIXO	SIGNIFICADO	SIMBOLOGIA
?	consulta	?[padrão]
-	remoção	-[padrão]
+	inserção	+[padrão]

O padrão corresponde à regra a ser especificada

A. Formato das Ações Adaptativas Elementares

Na 7-tupla da Expressão 2, o conjunto de ações adaptativas AR_0 leva conta o formalismo do dispositivo subjacente por meio de δ .

Complementando a Definição 4, conjuntos de ações adaptativas elementares são abstraídas em FAD, as quais interconectam o dispositivo subjacente à contraparte adaptativa, em conformidade com a Fig. 3 pelas FAD R e S. Essa figura mostra uma representação gráfica estática de uma transição genérica do AFA do tipo: $(x, i) : R \rightarrow y : S$ 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; R e S são as FAD executadas respectivamente “antes” e “depois” da transição do estado x para o estado y .

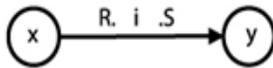


Fig. 3 Uma transição adaptativa genérica, onde R e S são FAD opcionais.

Consequentemente, o mecanismo adaptativo, composto de regras contidas em AM, tal que $AM \subseteq \mathcal{B} \times \delta_k \times \mathcal{A}$. Essas regras são aplicadas em cada passo operacional k em que δ_k é a função de transição do dispositivo subjacente no passo k .

Observe-se que, havendo mais de uma ação adaptativa elementar a ser executada, independentemente da ordem em que foram declaradas, têm precedência as consultas. Em seguida são efetuadas as remoções, e posteriormente as inserções. Possíveis transições em vazio são sempre executadas por último.

Ações elementares que referenciam elementos indefinidos não são executadas.

O formato das três modalidades de ações adaptativas elementares da Tabela I é indicado na Tabela II, com o padrão da Tabela I especificado entre colchetes.

TABELA II
FORMATO DAS AÇÕES ADAPTATIVAS PRIMITIVAS

MODALIDADE	SIMBOLOGIA
consulta	?[$x, i : R \rightarrow y : S$]
remoção	-[$(x, i) : R \rightarrow y : S$]
inserção	+[$(x, i) : R \rightarrow y : S$]

Detalhes sobre o formato das FAD são descritos a seguir.

B. Formato das Funções Adaptativas (FAD)

Quanto ao formato das FAD, pela Definição 4, as ações

adaptativas são indicadas pelo par ordenado (F;P), correspondendo a chamadas paramétricas de FAD, abrangendo os seguintes dados:

- 1) *A identificação*: F é o nome da FAD;
- 2) *Possíveis argumentos*: P é uma sequência de argumentos (p_1, p_2, \dots) da FAD F, todos opcionais.

Esses argumentos (p_1, p_2, \dots) tem a finalidade de designar valores a serem substituídos aos correspondentes parâmetros formais de F.. Tais parâmetros comporão as ações adaptativas primitivas de inserção, consulta ou eliminação do autômato. Apesar de opcionais, se parâmetros forem especificados, terão de ser fornecidos para ativar a correspondente FAD F.

Em outras palavras, os parâmetros formais de FAD são variáveis especiais, utilizados para que valores reais (argumentos) sejam atribuídos, sempre que determinada FAD seja ativada. Assim, aos parâmetros da FAD são atribuídos os valores recebidos como argumentos antes de ativar a FAD, permanecendo inalterados durante toda a sua execução.

No caso geral, além dos parâmetros, uma FAD pode ser especificada por variáveis e geradores, também opcionais, com os significados seguintes:

Variáveis: 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 AFA tais como estados ou estímulos.

Geradores: identificadores que recebem um valor único (isto é, nunca antes utilizado) no início da ação adaptativa e permanecem com este valor até o final da ação. São uma espécie de variável que contém a identificação do próximo estado a ser criado.

Os geradores são indicados com um expoente * apenas na declaração de nomes, diferenciando das variáveis. A especificação de uma FAD F qualquer, com n parâmetros q consiste no seguinte:

- 3) *Um cabeçalho*: $F(q_1; q_2; \dots; q_n)$
- 4) *Um corpo, com o seguinte formato*
{declaração de nomes (opcional):
declaração de ações (opcional)}.

Sobre as variáveis, tanto as ações adaptativas elementares de consulta quanto as de exclusão efetuam uma busca no conjunto corrente de transições para encontrar qualquer transição que corresponda ao padrão dado. Quando tal transição é encontrada, às variáveis são atribuídos os valores reais correspondentes à transição encontrada, pois as mesmas são usadas no lugar de qualquer um dos componentes de ação adaptativa elementar.

Considerando $v_i; i = 1, 2, \dots, m$ como variáveis e $g_j; j = 1, 2, \dots, n$ como geradores, a declaração de nomes apresenta o aspecto seguinte: $v_1, v_2, \dots, v_m, g_1^*, g_2^*, \dots, g_n^*$.

A título de esclarecimento, veja abaixo o formato de uma FAD hipotética η , e a correspondente especificação de um gerador $ger1$, uma variável $var1$, dois parâmetros α e β e três ações adaptativas elementares, para um alfabeto $\Sigma = \{a, b, c, d\}$.

O par ordenado (F,P) é indicado por (η, α, β) que pode ser exemplificado por uma chamada de transição adaptativa qualquer tal como $[(1, a) : \eta (2, 6) \rightarrow 2]$. Nesse caso, a ação adaptativa que ativa a FAD η ocorre antes que o AFA mude do estado 1 para o estado 2, desde que um token a seja recebido.

$\eta(\alpha, \beta) \{ger1^*, var1: \\ ?[(r_i, b) \rightarrow \beta] \\ +[(ger1, a) \rightarrow \alpha] \\ -[(var1, \epsilon) : A r_{i+1}]\}$
--

Concluindo, da mesma forma como a FAD η é ativada nesse exemplo, pela maneira que as FAD são conectadas às transições do AFA define-se o comportamento do autômato resultando em aceitar ou rejeitar a *string* de entrada.

Gráficamente, uma determinada FAD R é representada por R . (R seguido de ponto), caso seja do tipo antes. Similarmente, uma FAD S é representada por $.S$ (ponto seguido de S), caso seja do tipo depois.

III. O ESTADO-DA-ARTE

Este tópico apresenta uma breve revisão do estado-da-arte sobre SLRD.

Tendo em vista que a Prop3 não é muito clara, esta foi posteriormente melhor formalizada pela propriedade da corda, descrita a seguir.

A. Linha Reta

A noção de linha reta está associada à propriedade da corda.

Para a vizinhança-8, [9] demonstrou que a condição necessária e suficiente para que um arco digital qualquer seja um SLRD ideal é atender à propriedade da corda.

Definição 5: A Propriedade da Corda: 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 [9].

A Definição 5 implicou na demonstração da existência de uma estrutura hierárquica composta de números consecutivos correspondentes às corridas dos símbolos especificados por Prop1 e Prop2. Essa estrutura de números consecutivos é expressa por uma propriedade adicional Prop4: 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 $P+1$).

Pela propriedade da corda, define-se um SLRD ideal como a seguir.

Definição 6: Linha Reta Digitalizada Ideal (SLRD ideal): Uma linha reta digitalizada ideal é aquela que atende à propriedade da corda, na vizinhança respectiva. Note que a propriedade da corda foi originariamente definida para a vizinhança-8, existindo especializações para a vizinhança-4 [18].

Também o conceito de pré-imagem é importante para o entendimento da perda de informação no processo de representar linhas e segmentos.

Definição 7: Pré-imagem: Dado um segmento digital em linha reta (SLRD), pré-imagem é o conjunto de linhas retas Euclidianas, cuja codificação para o formato de *string* resulta no mesmo SLRD dado. Demonstra-se que a pré-imagem corresponde a um polígono convexo em um espaço de parâmetros [6].

Além do comentado pela Definição 7, o processo de digitalização de uma linha reta Euclidiana específica introduz distorções e ruído, causando que o correspondente SLRD não atenda à propriedade da corda, resultando em muitos segmentos curtos na vizinhança de um pixel, de acordo com a mencionada propriedade. Portanto, é necessário algum tipo de medida, denominada métrica, apropriada para avaliar se dois SLRD pertencem a uma única estrutura linear [19]. Essa métrica pode definir uma função de vizinhança escolhida de modo a expressar como agrupar SLRD aproximados.

Contudo, “aproximado” é entendido no tocante a linhas retas visualmente corretas dentro de uma tolerância, não necessariamente no tocante à propriedade da corda. Portanto, de agora em diante, a menos que de outra maneira seja especificado, a nomenclatura deste texto não faz nenhuma distinção entre um SLRD ideal e as linhas “quase” retas nas proximidades da primeira, considerando que esse conjunto (de SLRD ideais e SLRD aproximados) é reconhecido pelo dispositivo adaptativo proposto.

Por outro lado, trabalhos como [20] mostraram exemplos de SLRD que violam a regularidade implícita na propriedade da corda, comentando que, na prática, Prop4 e Prop3 e são inviáveis na análise de arcos digitais. É mais razoável esperar uma ligeira variação nas corridas, dentro de um nível de tolerância, mas sempre mantendo a inclinação geral, delineando assim SLRD aproximados. O critério usado por [20] se concentrou em *strings* que satisfaçam as duas primeiras propriedades da conjectura, denominados códigos monotônicos, pois representam arcos digitais que são ascendente ou descendente com referência aos eixos de coordenadas x e y .

Adicionalmente, três trabalhos principais norteiam esta proposta. No primeiro, [21] segue um procedimento algorítmico similar ao de Freeman o qual define linhas discretas como digitalização de linhas Euclidianas. O segundo, [22] utiliza a noção de “segmentos borrados” fundamentada na “geometria discreta aritmética” introduzida por [23]. O terceiro, [24] fundamenta-se na noção de “linha discreta aritmética” fundamentada na “geometria discreta aritmética” introduzida por [23]. O trabalho [24] se propôs a ser uma solução para a rigidez no modelo de Freeman, mesmo após certa flexibilidade ter sido introduzida por [21]; bem como para a perda de conexão com a aritmética de [22].

Enfatize-se que o relacionamento 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 [25].

Esta pesquisa leva em conta que a representação adaptativa possibilita alteração das escalas de segmentos, de tal maneira que arcos irregulares numa determinada escala possam ser

reconhecidos como SLRD alterando-se as escalas relativas. Portanto, arcos irregulares podem se revelar, na verdade, como retas ao serem analisados numa escala compatível, utilizando métricas. A adaptatividade surge, portanto, como uma alternativa para incorporar os princípios da geometria discreta aritmética ao modelo de Freeman.

IV. ANÁLISE ESTRUTURAL

Outro método de representação, aplicado neste tópico, é a representação das linhas digitais com base em frações contínuas, estudado por [10], que resultou num algoritmo válido apenas para retas com inclinações de números racionais (a inclinação é o ângulo do segmento com referência ao eixo x , expressado, por exemplo, pela tangente desse ângulo). O estudo da modelagem de retas por frações contínuas tem sido continuamente pesquisado, com vários desdobramentos após o trabalho de [10], descritos em [4].

Similarmente à aproximação de pontos por polinômios de determinada ordem, o enfoque matemático da representação por frações contínuas propicia modelos, facilitando avaliação de erros e aproximações no processo de digitalização de retas partindo das correspondentes inclinações.

Sobre as correspondências entre inclinações e periodicidade, o chain code de retas com inclinações representadas por números racionais são periódicas, enquanto para inclinações representadas por irracionais não ocorre periodicidade, comentado na página 312 de [26].

Entretanto, mesmo sem considerar distorções e ruído, os modelos digitais têm que atender às condições da grade de amostragem corresponder a números inteiros num reticulado, o que resulta em modelos aproximados, independentemente de inclinações racionais ou irracionais.

A seguir, apresenta-se uma breve análise estrutural dos SLRD por meio dos formatos das respectivas *strings*, apresentadas como exemplos. Sem perda de generalidade, considerase o chain code representando uma linha contínua de ângulo de orientação φ com o eixo positivo x tal que $\varphi \in \left[0, \frac{\pi}{4}\right]$ (indica que φ pertence ao intervalo fechado entre 0 até $\frac{\pi}{4}$ radianos) [27]. Nesse intervalo de φ , as *strings* correspondentes aos SLRD apresentam o símbolo a ocorrendo isolado enquanto o símbolo b ocorre agrupado.

A. Ordem dos Modelos

Conforme [28], dependendo do grau de precisão requerido, derivam-se fórmulas para o ângulo de orientação dos SLRD bem como modelos matemáticos gerais ao se aproximar um SLRD 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 digital em linha reta (USLR) é importante:

Definição 8: Unidade de Segmento Digital em Linha Reta (abreviado USLR) é o menor segmento possível em que um SLRD pode ser subdividido a fim de manter o correspondente ângulo de orientação.

As premissas de [28] para especificação dos modelos

foram: *i)* O SLRD deve atender à propriedade da corda; *ii)* O SLRD não pode ser subdividido indefinidamente, de acordo com a Definição 8; *iii)* A estrutura do arranjo de pontos digitais do SLRD depende exclusivamente da correspondente inclinação.

A inclinação, definida pela tangente do segmento Euclidiano no primeiro octante, pode ser expressa pela fração contínua da Expressão 3 aproximada a um modelo nos moldes da Definição 9, a seguir.

$$SL = \frac{B}{A} = \frac{1}{P \pm \frac{1}{M \pm \frac{1}{K \pm \dots}}} \quad (3)$$

em que os sinais positivos e negativos são para acelerar a convergência da fração e reduzir o número de termos da mesma, A , B e P são inteiros ($B < A$), enquanto M , K são inteiros não negativos.

Definição 9: 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.

Assim, a Expressão 3 pode se apresentar de várias formas; tais como $SL = \frac{B}{A} = \frac{1}{P}$, $SL = \frac{B}{A} = \frac{1}{P \pm \frac{1}{M}}$, $SL = \frac{B}{A} = \frac{1}{P \pm \frac{1}{M \pm \frac{1}{K}}}$,

denominadas frações contínuas de primeira ordem, de segunda ordem e terceira ordem, respectivamente. A essas três, seguem-se ordens superiores, dependendo dos valores de A , B , ou dos requisitos de precisão.

Os modelos de primeira e segunda ordem são considerados principais para descrever o arranjo no ajuste dos padrões de pontos de SLRD na grade. Note que os ângulos das USLR relativamente ao eixo x , denominados θ_U podem variar no SLRD. A Definição 10, abaixo, comenta sobre o ângulo de orientação principal do SLRD, que é o ângulo que se destaca dentre a distribuição de θ_U ao longo do segmento.

Definição 10: Ângulo de Orientação Principal: Entende-se como ângulo de orientação principal do SLRD, denominado θ_S , como aquele ângulo que se destaca, por algum critério, dentre a distribuição de ângulos das USLR individuais do segmento. Por exemplo, [20] denomina a direção (o ângulo de inclinação) correspondente ao símbolo que ocorre isoladamente de direção de transição, e a direção correspondente ao outro símbolo de direção principal.

B. 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 SLRD da Fig. 4 atravessa $B = 1$ linhas e P colunas, resultando para a inclinação SL o valor de $SL = \frac{B}{A} = \frac{1}{P}$, onde P é um número inteiro denominado fator de inclinação de primeira ordem. P como número inteiro significa que os pontos digitais do segmento de reta são uniformemente distribuídos em linhas e colunas da grade, tal que cada linha é um USLR que contém P pontos consecutivos, com o arranjo dos padrões do SLRD do tipo PPP...PP...P.

Para o modelo de primeira ordem, as *strings* S_U de cada USLR seguem o formato da Expressão 4 em que m é o mesmopara toda USLR do SLRD.

$$S_U: b^m a; m \geq 1 \quad (4)$$

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

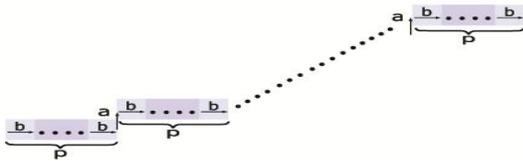


Fig. 4. Modelo de primeira ordem, caracterizado por P constante

C. Modelo de Segunda Ordem

No modelo de segunda ordem, P é o número inteiro mais próximo de A/B , ou seja, $P = \lfloor A/B \rfloor$. Nesse modelo, o SLRD tende a ajustar os pontos digitais em B linhas, tão uniformemente quanto possível, a fim de manter a inclinação global. No modelo de segunda ordem da Fig. 2, o arranjo dos padrões do SLRD é do tipo PPP....QPP....PQ em que o ajuste na grade 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. As strings S_U das USLR serão de dois tipos, em que o primeiro tipo é $b^m a$ e o segundo vai depender do padrão de números consecutivos $(m+1)$ ou $(m-1)$: $S_U: \begin{cases} b^m a, \\ b^{m+1} a; m \geq 1 \text{ ou } b^{m-1} a; m \geq 2. \end{cases}$

Nesse modelo, o ângulo de orientação principal, aquele que se destaca, é $\theta_S = \arctan(1/P)$ com inclinação $1/P$, entretanto ocorrem USLR 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 $\varphi \in [\arctan(\frac{1}{P+1}), \arctan(\frac{1}{P})]$ (supondo $Q = P+1$) [27]. Esse resultado de φ confirma o comentado na seção III sobre o trabalho de [9] quanto ao padrão de números inteiros consecutivos das corridas.

V. ESTRUTURAS ADAPTATIVAS PROPOSTAS

[29] 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 strings, a string observada de entrada e a string correspondente ao modelo ideal.

Analogamente, com o propósito de apresentar os fundamentos da solução proposta por este trabalho, seja um AFA capaz de avaliar a similaridade (dentro de uma certa tolerância) entre duas strings, o SLRD observado e a string reta, ou seja a string definida pelo modelo de SLRD conforme a propriedade da corda. A configuração inicial do AFA é um AF ND_0 que aceita a string reta. A contraparte adaptativa AM é formada de “ações adaptativas”, que levam em conta todas as possíveis instâncias da string observada devido aos possíveis erros. Tais ações adaptativas têm a finalidade de

alterar a configuração inicial para uma nova configuração (representada por um AF diferente de ND_0), conseqüentemente levando o AFA 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.

A. Preliminares

1. Recuperação de Erros

Vários autores têm relatado os métodos de recuperação de erros, em particular [30] na área de compiladores. Supondo uma string submetida a um processo de análise sintática por um reconhecedor, denomina-se recuperação de erros ao método de tratamento de erros que consiste em termos gerais, nas seguintes etapas: *i*) O analisador se mantém no processo de consumo de símbolos (ou caracteres) de entrada, até que um símbolo errado seja identificado; *ii*) Isolado o erro, ativa-se algum mecanismo de recuperação de erros; *iii*) 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. Restrições existem em alguns métodos, exigindo que o usuário forneça informações adicionais sobre o processo de reparo.

É evidente a analogia entre a recuperação de erros na análise sintáticas de SLRD com a recuperação de erros em outros processos sintáticos gerais. Entretanto, é possível concluir que não ocorre uma igualdade total ou completa ao se retomar o trabalho de [29], referenciado no início desta seção.

[29] abstrai todos os erros no processo de digitalização, independentemente da sua origem, nos denominados erros de edição: substituição, eliminação e inserção de símbolos. Todavia, o mesmo procedimento não é viável para o caso de SLRD. 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.

O propósito desta pesquisa em recuperação de erros compreendeu estudos introdutórios sobre como modelar, por meio do AFA, 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 α, β, γ e δ são pequenos erros correspondentes às influências espúrias. Esses estudos preliminares foram cumpridos a fim de incorporar a incerteza nos modelos das linguagens.

B. Técnica Utilizada em Recuperação de Erros

A fim de exibir características comuns ao que está sendo aludido, lembre-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 pelo seu deslocamento com relação ao 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 [26].

As técnicas de recuperação de erros de SLRD desta abordagem empregam um dispositivo similar aos *spyrographs* na forma de *loops* adaptativos da Fig. 5, tal que, nesses *loops*, a circunferência mencionada é representada por *to* estados (de L_1 a L_{to}) do AFA, os quais são percorridos cíclicamente pelo autômato.

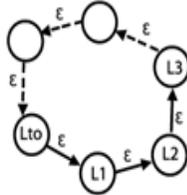


Fig. 5 Um grafo na forma de *loop* adaptativo genérico, interpretado como caso particular de sequência conexa da Definição 2 cujos estados inicial e final são idênticos.

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

C. Erros em Ângulo

Supondo inicialmente um SLRD S como uma *string* de comprimento n primitivas, $|S| = n$, tal *string* é composta da concatenação de λ *substrings*, onde $n > \lambda$ e $1 \leq i \leq \lambda$ sendo que cada *substring* é uma USLR U de S da seguinte forma

$$S: U_i ; i = 1, 2, \dots, \lambda.$$

(5)

Entretanto, na Expressão 5 a SLRD nula ($\lambda = 0$) não é definida, nem os casos em que o primeiro e o último USLR (U_1 e U_λ) estão truncadas, ou seja, fora da estrutura do ângulo θ_S de orientação do SLRD global. Essa situação é considerada no item C.1 desta seção.

Assim, cada USLR é também um SLRD elementar que apresenta um ângulo e um comprimento “locais”. Portanto, para cada USLR, definida pelo código isolado de Prop2, basta o SLRDA correspondente “entrar” em um loop adaptativo da Fig. 5, cuja quantidade de estados é variável adaptativamente em função das corridas do símbolo que ocorre em grupos.

D. Implementação de Segmento Digitalizado Adaptativo

1. Unidades de Segmento Inicial e Final

Os extremos de um SLRD hipotético podem estar truncados ou completamente fora do modelo estrutural. No primeiro caso, o SLRD deve ser incorporado ao modelo adaptativo, e rejeitado no segundo. A Fig. 6 mostra um AFA que modela a U_1 pertencendo ao conjunto $\{a^n b : n = 3, 4, 5\}$. Na figura, o parâmetro r_4 é o último estado de uma sequência conexa iniciando no estado r . A partir dessa sequência conexa, o AFA pode remover até quatro transições em vazio por intermédio da FAD RA apresentada na Tabela III, 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 AFA para o estado final; desde que não sejam recebidos mais do que quatro tokens a . Uma estrutura para a última USLR seria bastante semelhante.

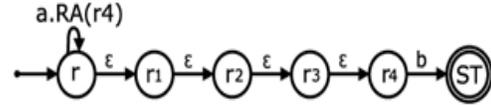


Fig. 6. SLRDA para modelar a primeira USLR de um SLRD.

TABELA III
FAD PARAMÉTRICA RA, DO AFA DA FIG 6

$RA(x_i)\{vr1, vr2, vr3, vr4:$ $-[(x_i-1, \epsilon) \rightarrow x_i]$ $-[(x_i, vr1) : vr2 \rightarrow vr3 : vr4]$ $+[(x_i-1, vr1) : vr2 \rightarrow vr3 : vr4]$ $-[(r, a) \rightarrow a : RA(x_i)]$ $-[(r, a) \rightarrow a : RA(x_i-1)]$
--

2. O Segmento Digitalizado Adaptativo

A representação das diferentes instâncias do modelo ideal afetado pelos erros em ângulo requer que o SLRDA atue numa faixa de ângulos. Este tópico ilustra a modelagem de SLRDA, exemplificando com USLR 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 correspondente a cada U_i de um SLRD S pode variar na faixa seguinte:

$$\arctan(3) \leq \theta_U \leq \arctan(5).$$

A Fig. 7 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 FAD do AFA estão descritas na Tabela IV.

A seguir, para cada USLR, o AFA percorre um ciclo de estados que inicia e termina no estado u ($u, u_1, u_2, u_3, u_4, u_5, u$) formando uma variante de *loop* adaptativo com quantidade de estados contante, porém as transições consumidas variam pela ação da FAD RA, seguindo o descrito no item D1. A FAD RB garante a existência das transições entre os estados u_4 e u_5 após cada USLR U_i processada, tendo em vista que a FAD RA remove as transições para esses estados em cada volta no ciclo. Este processo é repetido até que o fluxo de entrada se esgote aceitando USLR do tipo $\{a^n b : n = 3, 4, 5\}$.

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 elementar $- [(x_i, vr1) : vr2 \rightarrow vr3 : vr4]$ de RA remove a transição correspondente ao token c , rejeitando o segmento.

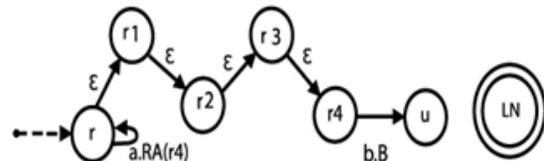


Fig. 7. Configuração inicial de SLRDA, considerando erros em ângulo

vizinhança $\max\{|x - h|, |y - h|\} < n$; onde n é a ordem do modelo, dependente da situação momentânea e dos estímulos.

Relembrando que o modelo de primeira ordem da Fig. 4 se caracteriza por P constante, um esboço dos modelos de ordens superiores pode ser observado na mesma Fig. 4, desde que habilitando P a ser um número inteiro variável teoricamente até infinito: $P(a, b^m): 0 \leq m < \infty$.

A. Implementação e Teste da Vizinhança Adaptativa

A vizinhança adaptativa associa um conjunto de arcos a um SLRDA correspondente, que reconhece esse conjunto. Este tópico apresenta a implementação e teste de modelos de SLRDA atendendo à proposta de vizinhança adaptativa, as quais estão associadas a regiões delimitadas por arcos responsáveis pelos seus contornos. Há dois tipos principais desses arcos delimitadores: arcos côncavos e arcos convexos. O item a seguir mostra apenas o caso de vizinhança delimitada por arco côncavo, pois o caso convexo apresenta estruturas aproximadas ao côncavo.

1. Vizinhança delimitada por arco côncavo

Na seção V. o loop adaptativo da Fig. 5 foi a estrutura usada para obter os comprimentos corrigidos l_E . Neste tópico a mesma estrutura é aplicada, tendo o valor t_0 o significado de comprimento de referência para o autômato alterar a vizinhança adaptativamente.

O SLRDA da Fig. 9 reconhece *strings* do tipo $W = \Psi S$ em que S é o SLRDA a ter sua vizinhança alterada adaptativamente em função de seu próprio comprimento dinâmico, relativamente à quantidade de estados t_0 do loop. Ψ é uma *string* empregada para infomar o valor de t_0 ao SLRDA. Um símbolo adicional y foi utilizado nas implementações, apenas para informar os novos valores de t_0 ao autômato, ou seja, Ψ é a *string* $\Psi = y^{t_0}$.

O SLRDA mencionado constrói o loop adaptativo de quantidade de estados t_0 de acordo com o número de símbolos y , pela FAD RO. A transição $[(r_1, a) \rightarrow r_1]$ é uma simplificação para modelar os possíveis símbolos a da USLR inicial.

Com o primeiro símbolo b , o AFA passa a consumir as USLR subsequentes. Cada USLR de S ativa as FAD IB e RB,

Para uma *string* de entrada $S: U_i; i = 1, 2, \dots, \lambda$, a configuração de SLRDA mostrada à esquerda da Fig. 9 é mantida enquanto $1 \leq i < t_0$ aceitando USLR do tipo $\{a^l b: l = 2\}$, sendo que quando $i = t_0$ a FAD IB insere uma nova transição em vazio na sequência conexas de s . Para $t_0 \leq i < 2t_0$, as USLR aceitas passam a ser na faixa $\{a^l b: l = 2, 3\}$. Quando $i = 2t_0$, a FAD IB insere nova transição em vazio na sequência conexas de s alterando a configuração para a Fig. 9, à direita. A USLR aceitas passam a ser na faixa $\{a^l b: l = 2, 3, 4\}$. E assim por diante, a cada i múltiplo de t_0 a FAD IB insere uma transição na mencionada sequência conexas alterando a vizinhança.

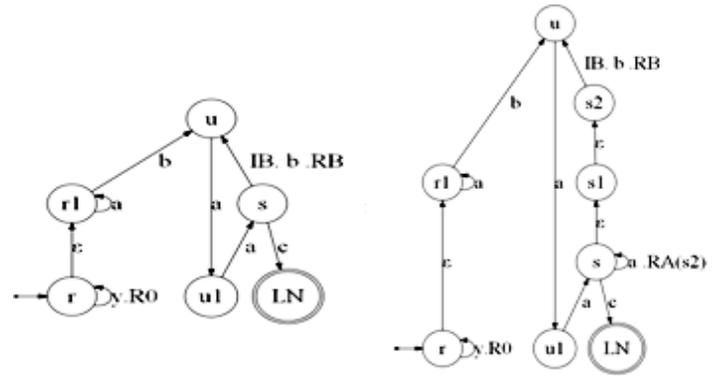


Fig. 9 À esquerda está a configuração inicial de SLRDA que ajusta a correspondente vizinhança em função de t_0 , do loop adaptativo, pela quantidade de estados da sequência conexas de s . À direita está a configuração no início da terceira volta, pela inclusão do estado s_2 .

Resumindo o processo, cada USLR de S ativa as FAD IB e RB, tal que: i) As transições da sequência conexas $s_1, s_2, s_3, \dots, s_v$ que existirem no autômato são removidas pela FAD RB (isso porque a USLR anterior pode não ter a quantidade máxima de tokens a previstos na vizinhança afeta à configuração do AFA); ii) A FAD IB insere novamente no autômato as mesmas transições $s_1, s_2, s_3, \dots, s_v$ removidas pela FAD RB, a fim de possibilitar o consumo da USLR seguinte; iii) A FAD IB movimentada um ponteiro¹³ uma posição adiante no loop adaptativo; iv) A cada volta completa no loop adaptativo efetuada pelo ponteiro, a FAD IB executa também o seguinte: a) Altera a vizinhança do SLRDA pela inclusão de uma transição $[(s_v, \varepsilon) \rightarrow s_{v+1}]$ incrementando a sequência conexas $s_1, s_2, s_3, \dots, s_v$ do estado s_{v+1} ; b) Altera a FAD RB para que remova, no passo i) acima, também as transições relacionadas ao estado s_{v+1} , incluído no passo anterior; (c) Remove a transição $[(s, a) \rightarrow s.RA(s_v)]$ e insere a transição $[(s, a) \rightarrow s.RA(s_{v+1})]$.

A vizinhança é delimitada por arco côncavo devido ao ângulo máximo de cada USLR que aumenta gradativamente seguindo os ângulos $\arctan(2), \arctan(3), \arctan(4), \dots, \arctan(\lfloor \lambda/t_0 \rfloor)$ relacionados à sequência conexas do estado s .

B. Comparação com outros modelos

Este tópico posiciona esta pesquisa dentro do contexto do estado-da-arte apontando as semelhanças e diferenças entre os diversos modelos, indicando vantagens e desvantagens.

Confrontando esta proposta com os modelos tradicionais, sem adaptatividade, um primeiro aspecto é quanto à questão da flexibilidade dos modelos.

1. Flexibilidade

O modelo tradicional, sem adaptatividade, tem que atender à propriedade da corda, rejeitando os arcos da Fig. 10 caracterizados por USLR U na faixa $U(b, a^m): 2 \leq m < 5$, pois, de acordo com a propriedade da corda, deve existir um

¹³ Entende-se como ponteiro uma transição elementar, podendo ser em vazio, denominado pelo seu estado que se mantém fixo, enquanto o outro estado varia.

padrão de números consecutivos nas corridas, sem apresentar irregularidades. Na Fig. 10, à esquerda, o arco é codificado por $(a^3 b)^6(a^4 b)^5(a^5 b)^5(a^6 b)^2 a^4 b (a^3 b)^6$. O arco à direita é codificado por $(a^3 b)^6(a^4 b)^5(a^5 b)^5(a^6 b)^5$.



Fig. 10. Exemplos de SLRD reconhecidos pelo SLRDA da Fig. 9 com $to = 5$; porém não reconhecidos pelo método tradicional, sem adaptatividade.

Dentre os modelos tradicionais, [21] obteve 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 foi $d = \lfloor (p + 1)/2 \rfloor$ tal que os arcos aceites tenham USLR U apenas na faixa $U(b, a^m)$: $p \leq m \leq d$. Quanto aos resultados obtidos por [21] destacam-se a redução em espaço de memória afeto às estruturas de dados e a complexidade em tempo dos algoritmos torna-se linear.

De modo geral, ao apresentarem comportamento invariável, os métodos tradicionais tendem a segmentar os arcos digitais em grande quantidade de segmentos curtos. Em especial, os métodos com valores pré-definidos numa faixa, exemplificados por [21], reduzem o número de segmentos curtos, porém perdem a característica global dos arcos, rejeitando os SLRD da Fig. 10, que são seccionados em diversos SLRD.

As alterações dinâmicas de vizinhanças desta proposta, que alteram a funcionalidade dos algoritmos em função dos estímulos, permitem uma melhor caracterização global dos arcos, reconhecendo, por exemplo, os SLRD da Fig. 10.

2. Complexidade computacional

Devido às limitações de comportamento invariável ou predefinido numa faixa, os métodos tradicionais tendem a seccionar os arcos digitais em um conjunto composto por muitos segmentos curtos. Ao reduzir sensivelmente os elementos componentes desse conjunto, a proposta adaptativa minimiza as estruturas de dados requeridas para representação e armazenamento de dados.

Além disso, a introdução de tolerâncias adaptativas simplifica os algoritmos comparativamente aos métodos que apresentam comportamento invariável, reduzindo a complexidade em tempo dos algoritmos adaptativos, além de torná-la linear.

3. A vantagem da escala adaptável

Por estarem limitados a linguagens regulares em 1979, [7] foi obrigado a introduzir um factor de normalização de escalas, que poderia ser o comprimento total de um contorno

padrão. Entretanto, a finalidade principal da transformação de escala desta proposta passa a ser conforme os comprimentos relativos de partes de um arco com relação ao arco como um todo, e não eliminar o problema enfrentado por [7] dos comprimentos variáveis de arcos.

A escala adaptável reutiliza, num novo contexto, uma técnica conhecida desde a década de 70 do século passado. Nesse novo contexto, arcos identificados como linhas retas apenas pela geometria discreta aritmética de [23], podem ser avaliados corretamente pelo modelo proposto, numa escala compatível.

4. Posicionamento quanto ao estado-da-arte

O trabalho de [31] relaciona as 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 [21] e [24]; e guiado por regras (com adaptatividade hierárquica multi-nível) cujo conjunto de regras é variável apresentando funções adaptativas modificáveis, na qual se insere esta proposta.

Resulta, portanto, em todo um potencial evolutivo desta proposta partindo da reutilização e generalizações de formalismos que têm sido tradicionalmente utilizados.

VII. CONSIDERAÇÕES FINAIS

Esta pesquisa introduz a abordagem adaptativa no assunto da representação de linhas e arcos digitais, considerando pequenos erros. Comparando com outras representações, duas das principais vantagens verificadas na proposta 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.

Existem aplicações de SLRD, aproximando curvas por conjuntos de segmentos. Conforme [27], 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 SLRD tendem a apresentar bom desempenho com curvas gerais, prevendo que um dispositivo adaptativo represente arcos digitais em diferentes escalas. Em outra aplicação, [33] propõe um *polyautomaton* que reconhece linhas retas visando computação paralela. A idéia era que cada célula da grade fosse representada por um AF, que pode ser substituído por um dispositivo adaptativo, conforme esta proposta.

Os modelos e as técnicas de tratamento de erros e de tolerâncias da proposta, certamente auxiliarão em problemas sintáticos recorrentes, independentemente da área aplicada, e vice-versa. Por exemplo, ao se interpretar um contorno digital complexo, composto por arcos orientados em diversas direções, como um percurso a ser percorrido pelo autômato, foi relatada por esta pesquisa uma arquitetura para reconhecimento do contorno, que se baseia no mapeamento de

ambientes desconhecidos por robôs móveis, nos moldes de [34], tendo por base técnicas de inferência descritas em [35]. Concluindo, destaque-se a versatilidade da proposta, no sentido de que as técnicas adaptativas apresentadas podem ser aplicadas sempre que ocorrerem abstrações envolvendo modelos de retas e arcos, permitindo reutilização, generalizações ou revitalização de formalismos que têm sido tradicionalmente utilizados [32].

REFERENCES

- [1] João José Neto, Contribuições à metodologia de construção de compiladores. Tese de livre docência. Escola Politécnica da Universidade de São Paulo, 1993.
- [2] João José Neto, “Um levantamento da evolução da adaptatividade e da tecnologia adaptativa.” Revista IEEE América Latina. Volume 5. Número 7, pag. 496–505, Novembro. 2007.
- [3] Hemerson Pistori, Tecnologia Adaptativa em Engenharia da Computação: Estado da Arte e Aplicações, Ph. D. thesis, Escola Politécnica da Universidade de São Paulo. 2003.
- [4] R. A. Klette and A. B. Rosenfeld, “Digital straightness - a review,” Discrete Applied Mathematics, vol. 139, no. 1-3, pp. 197–230, 2004.
- [5] João José Neto, “Adaptive rule-driven devices - general formulation and case study,” in Proceedings of the..., Springer-Verlag, Ed., Pretoria, South Africa, July 2001, IMPLEMENTATION AND APPLICATION OF AUTOMATA 6TH INTERNATIONAL CONFERENCE, vol. 2494, pp. 234–250, CIAA 2001.
- [6] L. Dorst and A. W. M. Smeulder, “Length estimators for digitized contours,” Computer Vision, Graphics, & Image Processing, vol. 40, no. 3, pp. 311–333, 1987.
- [7] Kai Ching You and King-Sun Fu, “A syntactic approach to shape recognition using attributed grammars,” IEEE transactions on systems, man, and cybernetics, vol. 9, no. 6, pp. 334–345, June 1979.
- [8] H. Freeman, “Boundary encoding and processing,” Picture Proceedings and Psychopictorics, pp. 241–266, 1970.
- [9] Azriel Rosenfeld, “Digital straight line segments,” IEEE Transactions on Computers, vol. C-23, no. 12, pp. 1264–1269, 1974.
- [10] R. Brons, “Linguistic methods for the description of a straight line on a grid,” Computer Graphics and Image Processing, vol. 3, no. 1, pp. 48 – 62, 1974.
- [11] S. Shlien, “Segmentation of digital curves using linguistic techniques” Computer Vision, Graphics and Image Processing, vol. 22, no. 2, pag. 277– 286, 1983.
- [12] H.R. Lewis and C.H. Papadimitriou, Elements of the Theory of Computation, Prentice-Hall, 1981.
- [13] J. Feder, “Languages of encoded line patterns,” Information and Control, vol. 13, no. 3, pp. 230–244, 1968.
- [14] R. L. A. Rocha and J. J. Neto, “Autômato adaptativo, limites e complexidade em comparação com máquina de Turing.” in Proceedings of the..., Faculdade SENAC, Ed. PROCEEDINGS OF THE SECOND CONGRESS OF LOGIC APPLIED TO TECHNOLOGY, 2001, p. 33 a 48.
- [15] Forchhammer S. Aghito, S.M., “Context-based coding of bilevel images enhanced by digital straight line analysis,” IEEE Transactions on Image Processing, vol. 15, no. 8, pp. 2120–2130, 2006.
- [16] D. Proffitt and D. Rosen, “Metrication errors and coding efficiency of chain-encoding schemes for the representation of lines and edges,” Computer Graphics and Image Processing, vol. 10, no. 4, pp. 318–332, 1979.
- [17] K. S. Fu, Syntactic Methods in Pattern Recognition, Academic Press, 1974.
- [18] Christian Ronse, “A strong chord property for 4-connected convex digital sets,” Computer Vision, Graphics, and Image Processing, vol. 35, no. 2, pag. 259 – 269, 1986.
- [19] Peter F.M. Nacken, “Metric for line segments,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 12, pp. 1312–1318, 1993, cited By (since 1996) 18.
- [20] H. C. Lee and K. S. Fu, “Using the fft to determine digital straight line chain codes,” Computer Graphics and Image Processing, vol. 18, no. 4, pp. 359–368, 1982.
- [21] Partha Bhowmick and Bhargab B. Bhattacharya, “Fast polygonal approximation of digital curves using relaxed straightness properties,”

- IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, pp. 1590–1602, 2007.
- [22] I. Debled-Rennesson, F. Feschet, and J. Rouyer-Degli, “Optimal blurred segments decomposition of noisy shapes in linear time,” Computers & Graphics, vol. 30, no. 1, pp. 30 – 36, 2006.
- [23] J. P. Reveillès, Géométrie discrète, calcul en nombres entiers et algorithmique, Ph.D. thesis, Université Louis Pasteur, Strasbourg, 1991.
- [24] F. Feschet, “The lattice width and quasi-straightness in digital spaces,” in Proceedings of the..., Tampa, FL, 2008, INT. CONF. PATTERN RECOGN.
- [25] C. Fiorio, D. Jamet, and J. L. Toutant, “Discrete circles: an arithmetical approach with non constant thickness,” in Proceedings of the..., Electronic Imaging, Ed., San Jose (CA), 2006, SPIE VISION GEOMETRY XIV, vol. Volume 6066.
- [26] Reinhard Klette and Azriel Rosenfeld, Digital geometry: geometric methods for digital picture analysis, Morgan Kaufmann, 2004.
- [27] Nahum Kiryati and Olaf Kübler, “Chain code probabilities and optimal length estimators for digitized three-dimensional curves,” Pattern Recognition, vol. 28, no. 3, pp. 361–372, 1995.
- [28] Shu-Xiang Li and Murray H. Loew, “Analysis and modeling of digitized straight-line segments,” in Proceedings of the..., Rome, Italy, 1988, PROCEEDINGS OF INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION, pp. 294–296, Publ by IEEE, Piscataway, NJ, United States.
- [29] J. R. Garitagoitia, J. R. G. de Mendivil, J. Echanobe, J. J. Astrain, and F. Fariña, “Deformed fuzzy automata for correcting imperfect strings of fuzzy symbols,” IEEE Transactions on Fuzzy Systems, vol. 11, no. 3, pp. 299–310, 2003.
- [30] Roland C. Backhouse, Syntax of programming languages: theory and practice, Prentice-Hall International, 1979.
- [31] João José Neto, “Adaptatividade: Generalização conceitual,” in 3º Workshop de Tecnologia Adaptativa. Escola Politécnica da Universidade de São Paulo, 2009.
- [32] Gonzalo Bailador and Gracián Triviño, “Pattern recognition using temporal fuzzy automata,” Fuzzy Sets and Systems, vol. 161, no. 1, pag. 37 – 55, 2010, Special section: New Trends on Pattern Recognition with Fuzzy Models.
- [33] Jerome Rothstein and Carl Weiman, “Parallel and sequential specification of a context sensitive language for straight lines on grids,” Computer Graphics Image Processing, vol. 5, no. 1, pp. 106–124, Mar. 1976.
- [34] Miguel Angelo de Abreu de Sousa, “Mapeamento de ambientes desconhecidos por robôs móveis utilizando autômatos adaptativos,” M.S. thesis, Escola Politécnica da Universidade de São Paulo, 2006.
- [35] P. M. Ivone, “Um estudo dos processos de inferência de gramáticas regulares e livres de contexto baseados em modelos adaptativos.” Dissertação, Escola Politécnica da Universidade de São Paulo, 2006.

Leoncio C. Barros Neto. Formado em Engenharia Elétrica pela Escola de Engenharia Mauá, São Caetano do Sul, SP, em 1979. Concluiu o mestrado pela USP em 1994. Em 1981 foi aprovado em concurso público de âmbito nacional, integrando o Corpo de Engenheiros Navais da Marinha do Brasil (MB). Na MB exerceu atividades técnicas em Setores de Manutenção e Pesquisas, iniciando programa de doutorado na USP em 2006, após transferência para a reserva. Suas áreas de interesse e pesquisa abrangem automação, sensores inteligentes, eletrônica de potência, sistemas embarcados, guerra eletrônica e suas aplicações.

André R. Hirakawa. Recebeu os títulos de Engenheiro Eletricista e correspondente mestrado pela USP em 1990 e 1992 respectivamente. Em 1992, como pesquisador, foi integrado no programa de doutorado da Yokohama National Universidade, em Yokiohoma, Japan, tendo recebido o título de PhD em 1997. Em 1998, iniciou atividades no Departamento de Engenharia de Computação e Sistemas Digitais da USP, sendo atualmente Professor Associado. Suas áreas de interesse e pesquisa abrangem automação e robótica, sensores inteligentes, eletrônica de potência, planejamento de rotas, AVG’s, e sistemas sem fio, tanto aplicados à automação agrícola quanto em outros ambientes.

Antonio M. A. Massola. Professor Titular da USP.