

# Um Método de Reconhecimento Sintático de Caracteres para Identificação de Placas de Veículos

AIRTON MARCO POLIDÓRIO<sup>1</sup>  
DÍBIO LEANDRO BORGES<sup>2</sup>

<sup>1</sup> CEFET-PR e UEM - Universidade Estadual de Maringá  
DIN - Departamento de Informática  
Av. Colombo, 3690 - Campus Universitário  
87020 - 900 Maringá, PR, Brasil  
ampolido@din.uem.br

<sup>2</sup> UFG - Universidade Federal de Goiás  
Escola de Engenharia Elétrica  
Praça Universitária S/N  
74605 - 220 Goiânia, GO, Brasil  
dibio@eee.ufg.br

**Abstract.** This paper proposes an automatic method system for recognizing alphanumeric characters, applied for identification of vehicle number plates. For solving the problem, it was developed a syntactic technique that joins two sets of the attributes through syntactic rules. The sets of attributes, and the syntactic rules are both detailed in this paper. All the preprocessing methods, necessary for solving the problem, are given here. The results present high quality, and processing performance is efficient.

## 1 Introdução

Pesquisas em reconhecimento automático de caracteres alfanuméricos que compõem a placa de um veículo [Polidório e Borges (1996)], [Cowell (1995)] e [Nijhuis *et al.* (1995)], através de processamento de imagens, remontam ao início da década de 70, quando foram desenvolvidos os primeiros protótipos. Entretanto, sistemas comerciais ficaram disponíveis somente a partir do início da década de 80.

O problema de reconhecer os caracteres de placas de veículos pode ser dividido em três outros problemas: (1) localizar a placa do veículo; (2) Segmentar os caracteres em relação ao fundo da placa, e pré-processá-los e; (3) Identificar os caracteres segmentados. Neste trabalho, é apresentada uma solução para o problema (3).

Para o reconhecimento dos caracteres aplicam-se técnicas de reconhecimento de padrões [Shalkoff (1992)]. As técnicas mais comuns são: estatísticas, sintáticas e baseadas em redes neuronais. Alguns pesquisadores, como [Nijhuis *et al.* (1995)], preferem combinar diferentes técnicas para alcançar melhores resultados. Entretanto, como em [Cowell (1995)], aqui são considerados que os caracteres usados em placas de veículos possuem fortes informações estruturais, as quais formam a base das técnicas sintáticas. Justificando assim, a aplicação desta técnica para solucionar o problema.

## 2 Conceitos básicos de reconhecimento de padrões e algumas de suas técnicas

As duas aproximações mais usadas em reconhecimento de padrões são as aproximações estatísticas (ou teoria de decisão) e as sintáticas (ou estruturais). Recentemente, redes neuronais artificiais estão provendo uma terceira forma de aproximação.

O objetivo dessas técnicas é distinguir tipos de padrões que são constituídos de diferentes características. Essas características podem ter um conceito geométrico, ou um valor numérico associado. As técnicas mais empregadas são baseadas no conceito de similaridade de padrões, onde um padrão é identificado como sendo  $x$ , se as características extraídas dele coincidirem o mais próximo possível das características de  $x$ .

Um sistema básico de reconhecimento de padrões é formado por um sensor de aquisição (por exemplo, uma câmera), um mecanismo de extração de características, e um algoritmo de descrição e/ou classificação. Dependendo do tipo de aproximação usada, também é necessário um conjunto de padrões altamente representativo, chamado de conjunto de treinamento, usado, inicialmente, para treinar o sistema.

A dificuldade maior em sistemas de reconhecimento de padrões está em determinar o conjunto de

características, ou atributos, passíveis de extração, ou seja, quais primitivas devemos buscar em um padrão que permitirá a sua descrição ou classificação, se possível, de forma única. Nota-se que um significativo esforço computacional é necessário para a extração dos atributos desejados de um padrão os quais, podem ser obtidos com erros e/ou ruídos.

Muitas vezes, procura-se classificar, reconhecer, ou descrever um padrão que é invariante quanto a algumas mudanças ou desvios em analogia com um *caso ideal*. Entretanto, na prática, os desvios são relevantes e devem ser considerados. Esses desvios podem ser provocados por perturbações geométricas causando alterações de escala e/ou de translação e/ou rotação no padrão, ou ainda distorções na forma geométrica do mesmo.

A distorção do padrão torna-se um grande problema para o seu reconhecimento, e nem sempre é possível encontrar características que não sejam gravemente afetadas por esses elementos degenerativos. O processo de reconhecimento pode ser inviabilizado, caso o sistema de reconhecimento não esteja habilitado para tratar essas distorções. Mas, ter conhecimento *a priori* dos tipos de distorções que os padrões estão sujeitos nem sempre é possível, geralmente esses problemas são tratados após uma extensa experimentação envolvendo as situações possíveis de aquisição do padrão.

É comum, principalmente empregando técnicas estatísticas, o uso de vetores de características e de atributos espaciais para desenvolver uma visão geométrica dos atributos de um padrão. Neste caso, esses atributos são arranjados em um vetor  $d$ -dimensional, chamado de vetor das características, denotado por  $\underline{x}$ , produzindo um espaço de medidas multidimensionais ou *espaço das características*. Se cada atributo é um número real, ou pode ser medido como um, então este espaço é  $R^d$ . Em outras instâncias, como nas unidades de redes neuronais artificiais, é preferível restringir o espaço das características como um subespaço de  $R^d$ .

Os vetores das características são tipicamente usados em técnicas baseadas em modelos estatísticos, e técnicas baseadas em redes neuronais artificiais. Mas o mesmo se mostra ineficiente quando é necessário representar os relacionamentos espaciais entre os atributos de um padrão. Para obter esses relacionamentos espaciais, é preferível recorrer às técnicas estruturais, ou sintáticas.

A Figura 1 mostra como a extração de atributos é um problema comum, entretanto singular, para cada aplicação e aproximação em reconhecimento de padrões.



Figura 1 Extração de características e suas aplicações.

### 3 Trabalhos Correlatos

[Cowell (1995)], propõe uma técnica sintática para solucionar o problema. Na gramática por ele descrita, existem quatro tipos de nodos (atributos): o tipo *a* representa um final de linha (Figura 2.a); o tipo *b* representa o encontro de duas linhas (Figura 2.b); o tipo *c* representa uma mudança de direção de linhas em ângulo de  $90^\circ$  (Figura 2.c); e o tipo *d* representa o cruzamento de duas linhas (Figura 2.d).

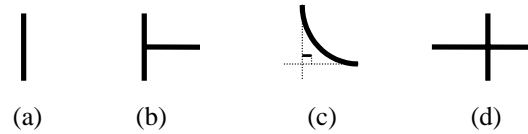


Figura 2 Tipos de nodos usados por Cowell.

Existem também três tipos de operadores, os quais são usados para representar os tipos de relacionamentos espaciais entre os atributos de um padrão: o operador ‘-’ é usado para simples concatenação; o operador ‘&’ é usado quando uma linha se divide em duas e não se encontram novamente; e o operador ‘\*’ usado para indicar a divisão de uma linha em duas, as quais se reencontram posteriormente.

O processo de geração de uma sentença da gramática é iniciado em um nodo do tipo *a*. Se não existe um, então preferencialmente, um nodo tipo *b* é escolhido, caso contrário um nodo tipo *c* é escolhido. Se existir mais de um ponto de entrada (nodo inicial), cada um gera uma sentença para identificação do padrão.

Considere, como exemplo, o caractere ‘E’, Figura 3. Se o ponto inicial é 1 ou 3 a sentença formada será *a-c-b-((a)&(c-a))*. Se for o ponto 2 a sentença será *a-b-((c-a)&(c-a))*. Qualquer uma das sentenças identifica o caractere ‘E’.

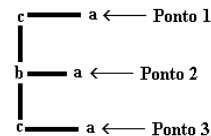


Figura 3 Os nodos do caractere ‘E’.

A solução de Cowell independe da translação, da rotação e do tamanho dos caracteres a serem reconhecidos. Mas este fato, gera um problema: os caracteres {O(letra), 0(zero), D}, os caracteres {W, M, S}, os caracteres {C, U, N, Z, 7, J}, os caracteres {Y, T}, os caracteres {I, 1} e os caracteres {6, 9}, geram sentenças homônimas entre si. Para tentar resolver este problema, além da informação contextual, foram derivados do conjunto de atributos base, dois outros subconjuntos de atributos. Com isto, minimizou-se o problema dos conflitos mas, sem resolvê-los totalmente. Para garantir a solução completa, foi necessária a aplicação de métodos heurísticos. A taxa de reconhecimento dos caracteres, com o emprego da técnica descrita, é de 95%, com 5% de erro.

Em sua solução, [Cowell (1995)] acredita que, por possuir fortes informações estruturais, os caracteres que compõem as placas dos veículos podem ser identificados com a aplicação de técnicas sintáticas. Entretanto, [Nijhuis *et al.* (1995)] preferem combinar diferentes técnicas para alcançar melhores resultados.

As técnicas mescladas por [Nijhuis *et al.* (1995)] são baseadas em redes neuronais artificiais e a aplicação de lógica *fuzzy*.

O sistema, por eles, proposto possui quatro partes: (1) um pré-processador, o qual aplica algumas técnicas padrão de processamento de imagens, (correção de contraste, eliminação de ruídos, etc.), para melhorar a qualidade da imagem; (2) uma unidade de segmentação que determina a posição da placa na imagem, usando o algoritmo *fuzzy c-means clustering*; (3) o reconhecedor, e (4) um analisador sintático do resultado produzido, para verificar a consistência sintática do mesmo, usando informações contextuais; por exemplo, letras e números sempre aparecem aos pares (letra-letra e número-número), (ex. KD - 51 - BF).

O módulo reconhecedor, funciona da seguinte forma: Os componentes binários conectados são usados como entradas para este módulo do sistema. Dois diferentes tipos de redes neuronais são usados para realizar o reconhecimento.

Por meio de uma rede **DTCNN** (*discrete-time cellular neural networks*), gera-se quatro características: a projeção horizontal, a projeção vertical, os componentes conectados na direção horizontal, os componentes conectados na direção vertical, para cada caractere da placa.

Cada uma dessas características, são transformadas em outras cinco características, gerando então, um total de vinte e quatro características as quais, servem como

entrada para uma rede neuronal **MLP**-*neural network* (*multi-layer perceptron network*) com 24 neurônios na entrada, 15 neurônios na camada interna, e 36 neurônios na saída (uma saída para cada caractere possível). Um caractere é tido como reconhecido, se o valor armazenado no neurônio de saída, correspondente ao caractere, exceder a 0,85 e todos os valores armazenados nos outros neurônios não exceder a 0,25, quando somados.

Os resultados obtidos pelo módulo reconhecedor são fantásticos. A taxa de reconhecimento é de 98,51%, contra uma taxa de erro de 0,02% e uma taxa de rejeição de 1,47% do total de caracteres submetidos ao módulo reconhecedor. Entretanto, aproximadamente, 25% das imagens submetidas não passam pelo segmentador e portanto, não podem ser identificadas.

#### 4 Proposta de uma nova solução

Na solução aqui proposta, são utilizados dois grupos de atributos para construir a gramática. O primeiro grupo é baseado no apresentado em [Cowell (1995)] (Figura 2), com exceção do atributo do tipo *c*. Aqui, faz-se distinção entre uma mudança de direção de 90° brusca, por exemplo: encontrada no L, e a mudança de direção mais suave, por exemplo: encontrada no J. O primeiro tipo de curva em 90° gera um atributo do tipo *c* e o segundo, gera um atributo do tipo *c*<sub>1</sub>.

Esses atributos buscam propriedades no traçado dos caracteres e são denominados de atributos-traço. O segundo grupo de atributos se refere à posição do atributo-traço do caractere, e são denominados de atributos-posicional [Polidorio e Borges (1996)] (Figura 4), os quais buscam os relacionamentos espaciais entre os atributos-traço de um caractere.

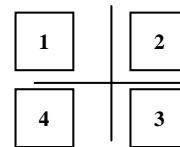


Figura 4 Atributos-posicional.

Os atributos-posicional são determinados por uma grade com quatro quadrantes, numerados de 1 até 4, começando pelo quadrante superior esquerdo no sentido horário (Figura 4). Para associar os atributos-posicional com os atributos-traço, a grade é encaixada sobre o caractere que está sendo processado, e assim, associa-se a cada atributo-traço uma posição quadrante onde este atributo se encontra.

A grade que delimita os quadrantes tem orientação fixa, mas o posicionamento dos seus eixos é variável, dependendo unicamente das dimensões do caractere a ser processado. Para encaixar a grade sobre o caractere, é necessário conhecer a altura ( $h$ ) e a largura ( $l$ ) do caractere, medidos em *pixels*.

A posição do eixo vertical da grade é dada pela Equação 1, e do eixo horizontal pela Equação 2.

$$P_v = \frac{l}{2} + offset_v \quad (1)$$

$$P_h = \frac{h}{2} + offset_h \quad (2)$$

onde, os valores *offsets* significam o deslocamento dos eixos horizontal e vertical da grade em relação ao centro do caractere. Através de experimentos, seus valores foram determinados como:  $0.1l$  e,  $0.1h$  para  $offset_v$  e,  $offset_h$ , respectivamente.

Para aplicar essas equações, o caractere não pode ser muito estreito, como o I e o l. Para estes casos, foi criada uma exceção: a posição do eixo vertical é dado por  $l - 1$ . Isto garante que as extremidades desses caracteres estejam sempre nos quadrantes 2 e 3, reduzindo o número de sentenças produzidas, causadas por pequenas rotações desses caracteres.

Para determinar a posição-quadrante dos atributos-traço dos outros caracteres, faz-se uso da seguinte seqüência de passos:

1. Os atributos-traço posicionados no quadrante  $i$  pertencem ao quadrante  $i$ .
2. O atributo-traço posicionado sobre o cruzamento dos eixos da grade pertence a todos os quadrantes (1, 2, 3 e 4).
3. Os atributos-traço que estiverem sobre o eixo vertical da grade são considerados como pertencentes ao quadrante 2, se estiverem numa posição superior ao eixo horizontal, ou ao quadrante 3, se estiverem numa posição inferior ao eixo horizontal.
4. Os atributos que estiverem sobre o eixo horizontal da grade são considerados como pertencentes ao quadrante 3, se estiverem à direita do eixo vertical, ou ao quadrante 4, se estiverem à esquerda do eixo vertical.

As sentenças usadas para identificar o caractere, serão formadas concatenando-se os atributos-posicional

com os atributos-traço. As regras para a concatenação são as seguintes:

1. A ordem dos atributos-posicional é 1, 2, 3 e 4, respectivamente.
2. Como em um quadrante pode haver mais de um atributo-traço, o primeiro caractere será o respectivo atributo-posicional, e após, todos os atributos-traço pertencentes ao quadrante.
3. A ordem de concatenação dos atributos-traço será  $a, b, c, d$  e  $c_1$ , respectivamente.

Uma restrição geral, que facilita o processamento, é adicionada aos conjuntos de regras citados anteriormente. Com base em observações práticas, verificou-se que caracteres que tinham atributos-traço do tipo  $b$  e/ou  $d$ , {A, B, E, F, H, K, M, N, P, Q, R, T, X, Y, W, 4, 6, 8, 9}, não geravam sentenças homônimas entre eles, se fossem excluídos os atributos  $c$  e  $c_1$  de suas sentenças. Como as verificações de mudanças de direção de 90 graus são feitas de forma separada da verificação dos outros atributos, esta restrição permite um ganho de tempo no processamento de placas que tenham esses caracteres.

Como a grade é fixa em relação a rotação, para obter a independência da rotação do caractere, é necessário haver mais de uma sentença para poder identificá-lo, uma para cada posição de rotação do caractere (Figura 5).

Como exemplo, montando as sentenças do caractere “E” (Tabela 1), considerando as possíveis rotações de 45 graus (Figura 5).

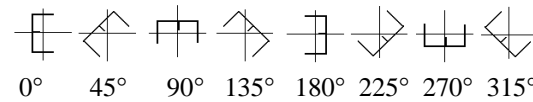


Figura 5 Exemplo de rotação com o caractere “E”.

Rotação	Sentença
0°	1ab2a3a4
45°	1ab2a34a
90°	12ab3a4a
135°	1 <sup>a</sup> 2ab3a4
180°	1 <sup>a</sup> 2ab34a
225°	12 <sup>a</sup> 3ab4a
270°	1 <sup>a</sup> 2a3ab4
315°	1 <sup>a</sup> 23a4ab

Tabela 1 Exemplo das sentenças geradas pela rotação do caractere E, aplicando o método proposto e sem os atributos  $c$  e  $c_1$ .

## 5 Pré-processamentos

Antes de submeter os caracteres de uma placa ao reconhecedor, é necessário submetê-los a alguns processamentos.

Esses processamentos envolvem:

1. segmentação dos caracteres em relação ao fundo da placa;
2. Afinamento dos caracteres;
3. Extração dos atributos-traço e dos atributos-posicional de cada caractere da placa.

Nesta seção, são abordados esses aspectos.

### 5.1 Segmentação dos caracteres

As imagens de placas de veículos são obtidas em ambiente natural, e sofrem toda a influência de agentes que deterioram a qualidade da imagem (iluminação, poeira, lama, etc.). Além disso, no Brasil, existem diversos padrões de placas pintadas com cores diferentes (cinza e preto, vermelho e branco, branco e vermelho, verde e branco, branco e preto, e amarelo e preto). Desta forma, não é pela aplicação de qualquer método segmentador que se consegue resultados positivos na segmentação dos caracteres em relação ao fundo da placa. Este segmentador deve ser do tipo adaptativo às variações degenerativas causadas pela ação dos agentes naturais e ainda, considerar os diversos padrões de placas citados acima.

Assim, foi desenvolvido um novo método de segmentação de caracteres em relação ao fundo da placa [Polidório (1997)], onde os resultados são obtidos rapidamente e com alta qualidade, conforme mostra a Figura 6.

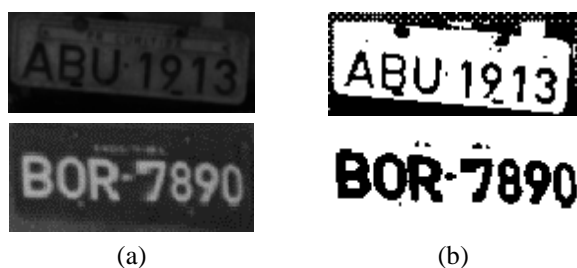


Figura 6. Exemplos da segmentação de caracteres de placas de veículos. (a) Imagens originais. (b) Caracteres segmentados.

### 5.2 Afinamento dos caracteres

Para identificar e localizar os atributos-traço e posicional envolvidos na gramática, é necessário seguir o traçado das linhas que compõem o caractere. Com o objetivo de facilitar esta tarefa, realiza-se o afinamento dessas linhas.

A escolha do algoritmo de afinamento para ser aplicado ao problema em questão deve ter algumas características especiais, como:

1. Executar rapidamente a ação de afinamento;
2. Não causar desconexões entre os *pixels* que formam as linhas dos caracteres;
3. Não remover pontos finais (atributo-traço  $a$ );
4. Minimizar a geração de ramos em linhas com reentrâncias e saliências;
5. Eliminar o maior número de *pixels* redundantes possível.

Considerando essas características essenciais, o algoritmo escolhido foi o proposto por [Zhang e Suen (1984)]. Esse algoritmo é relativamente simples de ser implementado, com tempo de processamento baixo, para caracteres com as dimensões tratadas neste trabalho (15 x 20 *pixels* em média), adiciona-se ao seu desempenho a qualidade dos resultados que ele produz. Os caracteres afinados têm boa qualidade devido ao fato desse algoritmo não ser muito sensível às saliências e reentrâncias muito comuns em caracteres binarizados.

O algoritmo remove os *pixels* que contornam o objeto, exceto aqueles *pixels* que compõem o esqueleto do mesmo, de forma iterativa. Para preservar a conectividade, a remoção dos *pixels* é dividida em duas sub-iterações. Em cada uma dessas sub-iterações, o *pixel*  $P(i, j)$  para ser removido deve ter alguns relacionamentos especiais em relação aos 8-vizinhos, ( $P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7$ ), (Figura 7).

$P_3$ ( $i-1, j-1$ )	$P_2$ ( $i-1, j$ )	$P_1$ ( $i-1, j+1$ )
$P_4$ ( $i, j-1$ )	$P$ ( $i, j$ )	$P_0$ ( $i, j+1$ )
$P_5$ ( $i+1, j-1$ )	$P_6$ ( $i+1, j$ )	$P_7$ ( $i+1, j+1$ )

Figura 7 O pixel  $P$  e seus 8-vizinhos.

Considerando que o objeto de interesse tem seus *pixels* com valor 1 e o fundo tem valor 0, as condições para eliminação do *pixel*  $P$  para a primeira sub-iteração são:

- 1)  $2 \leq B(P) \leq 6$
- 2)  $A(P) = 1$
- 3)  $P_0 * P_2 * P_6 = 0$
- 4)  $P_0 * P_4 * P_6 = 0$

onde:

$A(P)$  é o número de transições 0 e 1 em torno das 8-vizinhanças do *pixel*  $P$ , e

$B(P)$  é o número de vizinhos 8-conectados diferentes de zero do *pixel*  $P$ .

Para remover um *pixel* na segunda sub-iteração, as condições são:

- 1)  $2 \leq B(P) \leq 6$
- 2)  $A(P) = 1$
- 3)  $P_0 * P_2 * P_4 = 0$
- 4)  $P_2 * P_4 * P_6 = 0$

As subiterações são repetidas enquanto existirem pixels a serem removidos.

Um exemplo dos resultados produzidos pela ação do algoritmo de Zhang e Suen, podem ser vistos na Figura 8.



Figura 8. Exemplos da ação do algoritmo de afinamento.

## 5.2 Localizando e extraindo os atributos

Para localizar os atributos-traço e posicional pertencentes a um caractere, é necessário seguir as linhas de seu esqueleto. O ponto inicial de entrada para este processo é encontrado fazendo uma varredura seqüencial na direção horizontal e sentido esquerda-direita, pelo meio, vertical, da região da placa, até encontrar um *pixel* pertencente ao esqueleto do primeiro caractere (os *pixels* do fundo são brancos e os do esqueleto são pretos). A partir desse ponto, somente *pixels* pertencentes ao esqueleto são processados.

Para cada *pixel* processado, verifica-se a existência de direções alternativas a serem tomadas. Se existirem, elas são empilhadas para processamento posterior. Como os atributos-posicional dependem das dimensões do caractere, somente depois do término do processamento

completo do esqueleto do caractere é que eles serão definidos.

Para cada *pixel* processado, é obtida a sua posição (linha e coluna) em relação aos valores máximos e mínimos de cada dimensão (altura e largura). Estes valores é que permitirão a obtenção dos eixos da grade para determinação dos atributos-posicional.

Os atributos-traço, exceto o atributo dos tipos  $c$  e  $c_1$ , são determinados no momento do processamento de cada *pixel* do esqueleto. É desnecessário o uso de máscaras, ou de avaliação de caminhos alternativos a partir dele.

Em [Tamura (1978)] é definida uma forma fácil e rápida de determinar as propriedades de um *pixel*, simplesmente contando o número de transições preto/branco, CN, existentes nas 8-vizinhanças do *pixel* que está sendo processado. O valor obtido (CN) indica a propriedade do *pixel* (Tabela 2). Na Figura 9 podem ser vistos exemplos de detecção de atributos-traço pela análise proposta por Tamura.

CN	Significado	Tipo do atributo-traço
0	elemento isolado	nenhum
1	elemento ponto final	$a$
2	elemento conectado	nenhum
3	elemento junção em T	$b$
4	elemento com cruzamento	$d$

Tabela 2 Relação entre o número de transições preto/branco em torno de um *pixel* e o tipo do atributo-traço.

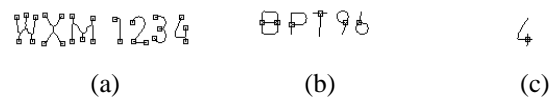


Figura 9 Exemplos de atributos-traço localizados pelas relações propostas por Tamura. (a) Atributos do tipo  $a$ . (b) Atributos do tipo  $b$ . (c) Atributos do tipo  $d$ .

Os atributos do tipo  $c$  e  $c_1$  são avaliados somente para caracteres que não possuam atributos do tipo  $b$ , e/ou,  $d$ , {C, D, G, I, J, L, O, S, U, V, Z, 1, 2, 3, 5, 7, 0}.



Figura 10 Exemplos de atributos-traço com mudança de direção de  $90^\circ$ . (a) Tipo  $c$ . (b) Tipo  $c_1$ .

A diferença entre esses dois atributos é muito sutil, e é medida em *pixels*. Se numa mudança de direção de 90° entre dois segmentos de retas existirem mais de dois *pixels*, considera-se como sendo determinado um atributo do tipo  $c_1$ , senão um atributo do tipo  $c$  é determinado numa posição entre os dois segmentos de reta considerados (Figura 10).

Detectados os atributos-traço e posicional pertencentes ao caractere que está sendo processado, é montada a sentença sintática de acordo com as regras da gramática descritas na Seção 4 deste trabalho. A sentença em questão permite pesquisar uma base de dados que relaciona a sentença com o caractere que ela representa, e assim são identificados todos os caracteres que compõem a placa de um veículo.

## 6 Experimentos e análise dos resultados

Este sistema foi implementado em um micro computador DX4 100MH com 8MB de RAM.

O tempo médio necessário para realizar os pré-processamentos e o reconhecimento dos caracteres é de 2,5 a 3 segundos. O algoritmo de afinamento é o mais demorado, o tempo gasto por ele é superior ao tempo de todos os outros processamentos realizados somados.

Cerca de 40 imagens foram utilizadas para gerar a base de dados, treinamento, do sistema (280 caracteres). Os testes do sistema foram feitos, usando outras 60 imagens (420 caracteres).

A taxa de reconhecimento é de 91%, a taxa de erro é de 0,8% e a taxa de rejeição é de 8,2%. A rejeição significa que o caractere não foi reconhecido, mas não foi confundido com nenhum outro. Isto se deve ao fato de haver pequenas diferenças estruturais entre caracteres de mesma grafia, mas o motivo principal da alta taxa de rejeição é devido às rotações dos caracteres. A maior falha do sistema está, em algumas situações, em confundir as letras D e O.

Na fase de testes, alguns caracteres não foram reconhecidos, mas não foram confundidos com nenhum outro (rejeição). Este fato ocorreu com maior frequência nos caracteres que tinham os atributos do tipo  $c$  e  $c_1$  avaliados, indicando que caracteres de mesma grafia de alguma forma eram estruturalmente diferentes, motivados pela rotação e/ou agentes degenerativos.

Um outro fator que contribuiu para o não reconhecimento dos caracteres está ligado com a forma de aquisição da imagem. A distância focal ideal para aquisição da imagem deve estar entre 2,5 e 3,0 metros

(distância suficiente para enquadrar três veículos de pequeno porte). Para distâncias maiores, ocorrem degenerações estruturais nos caracteres.

## 7 Conclusões

Apesar dos esforços empregados, não se conseguiu evitar totalmente as sentenças sinônimas entre as letras D e O. Isto é justificado pelas pequenas diferenças estruturais entre os dois caracteres, que facilmente se perdem pela forma de aquisição e nos pré-processamentos necessários.

Os outros caracteres que foram rejeitados pelo sistema, ou eram estruturalmente diferentes daqueles usados no treinamento, ou algum tipo de deterioração modificou as suas respectivas estruturas. Considerando-se que a rejeição é originada pela falta de sentenças que contemple alguns casos particulares, uma complementação do treinamento do sistema com mais imagens pode elevar a atual taxa de reconhecimento de 91% para 98%.

Em analogia ao trabalho proposto por Cowell, a nossa proposta gerou menos sinônimos, e por ter processamentos mais simples, acredita-se que o custo de processamento desta proposta seja menor.

## Referências

- J. R. Cowell, Syntactic Pattern Recognizer for Vehicle Identification Numbers, *Image and Vision Computing*, **13**, pp. 13-19, 1995.
- J.A.G. Nijhuis, et. al., *Car License Plate Recognition with Neural Networks and Fuzzi Logic*; EPIA'95 - Seventh Conference on Artificial Intelligence; pp. 229 - 235, Portugal, 1995.
- A. M. Polidório, D. L. Borges, *Reconhecimento de Caracteres para Identificação de Placas de Veículos*, Anais do IX SIBGRAPI, pp. 363-364, 1996.
- A. M. Polidório, *Segmentação e Reconhecimento de Caracteres para Identificação de Placas de Veículos*, Dissertação de Mestrado, CEFET-PR, setembro, 1997
- R. Shalkoff, *Pattern Recognition: Statistical, Structural and Neural Approaches*, John Wiley & Sons Inc., 1992.
- H. Tamura, *A Comparison of Line Thinning Algorithms from Digital Geometry Viewpoint*, Proc. 4th Int. Conf. Pattern Recognition, pp. 715-719, 1978.
- T. Y. Zhang, C. Y. Suen, *A Fast Parallel Algorithm for Thinning Digital Patterns*, Communications of the ACM, Vol. 27, No. 3, pp. 236-239, 1984.