

# Reconhecimento de Caracteres Alfanuméricos para Identificação de Placas de Veículos

AIRTON MARCO POLIDÓRIO<sup>1</sup>

DÍBIO LEANDRO BORGES<sup>2</sup>

CEFET - PR - Centro Federal de Educação Tecnológica do Paraná

Departamento de Informática e CPGEI

Av. 7 de Setembro, 3165 - 80230-901 Curitiba, PR, Brasil

{airton, dibio}@dainf.cefetpr.br

<sup>1</sup>Professor da Universidade Estadual de Maringá

<sup>2</sup>Professor Visitante

**Abstract.** This paper proposes a syntactic technique for recognizing alphanumeric characters from intensity images. The solution is applied to identification of vehicle plates, and the requirements of a working system in this domain are set to be achieved. The system is under development and an analysis is made on the steps of the solution.

## 1 Introdução

Este trabalho descreve um sistema de reconhecimento de caracteres alfanuméricos aplicado a identificação de placas de veículos. Atualmente, o processo de reconhecimento da placa do veículo é feita de forma manual por pessoas que analisam as imagens. Recentemente, em um ponto da cidade de São Paulo, um desses *guardas eletrônicos* aplicou 2.200 multas em quinze dias de funcionamento [Pinho (1996)]. Nessas circunstâncias, pesquisar uma solução automática para o problema de reconhecimento torna-se importante, também do ponto de vista prático. Nas próximas seções deste trabalho, serão abordados elementos de reconhecimento de padrões (*RP*) por técnicas sintáticas (*TS*), analisando a solução proposta em [Cowell (1995)] e apresentando uma nova proposta de solução.

## 2 Reconhecimento de Padrões: Técnicas Sintáticas

Existem várias técnicas que podem ser aplicadas em problemas de *RP*, como: estatísticas (*TE*), sintáticas (*TS*) e baseadas em redes neurais artificiais (*RNA*). As *TE* e *RNA* necessitam de um grande número de amostras para realizarem o treinamento de suas aproximações, já as *TS* necessitam apenas de um conjunto de atributos que são constantes para cada padrão. As *TS* expressam os padrões em termos de primitivas e operadores unidos por regras que descrevem o relacionamento espacial entre eles. Uma de suas vantagens é a necessidade de um pequeno conjunto de primitivas para especificar padrões complexos. A união do conjunto de primitivas  $V_T$ , do conjunto dos símbolos não-terminais, com o conjunto das regras de produção  $P$  e do símbolo inicial  $S$  formam a gramática  $G$  representada pelo grafo  $G = (V_T, V_N, P, S)$ . Para definir o grafo de uma gramática são necessários dois estágios: o primeiro é determinar quais elementos de um padrão constituem um nodo, e o segundo é desenvolver operadores que sejam

capazes de expressar os relacionamentos espaciais entre os nodos. Os relacionamentos entre os nodos podem ser expressos formalmente através de uma gramática *Web* [Schalkoff (1992)], a qual representa através de um grafo, o padrão a ser identificado, onde os nodos são as primitivas extraídas do padrão. As regras de formação nessa gramática são recursivas, e podem ser assim sintetizadas: (1) Os nodos (exceto um) correspondem aos elementos de  $V_N$  (conjunto dos elementos não-terminais), incluindo  $S$  (o elemento inicial). Usa-se um nodo terminal  $T$ ; (2) Um arco com origem em  $A_i$  e destino para o nodo  $A_j$ , rotulado com o valor  $a$  existe para cada produção da forma  $A_i \rightarrow aA_j$ ; (3) Um arco com origem no nodo  $A_i$  para o nodo  $T$ , rotulado com o valor  $a$ , existe para cada produção da forma  $A_i \rightarrow a$ .

Um dos principais trabalhos nessa área é apresentado em [Cowell (1995)]. Na gramática por ele descrita, existem quatro tipos de nodos: o tipo  $a$  representa um final de linha; o tipo  $b$  representa o encontro de duas linhas; o tipo  $c$  representa uma mudança de direção de linhas em ângulo de  $90^\circ$ ; e o tipo  $d$  representa o cruzamento de duas linhas. Existem também três operadores: o operador '-', usado para simples concatenação; o operador '&', usado quando uma linha se divide em duas e não se encontram novamente; e o operador '\*', usado para indicar a divisão de uma linha que se reencontrará posteriormente. O processo de geração de uma sentença da gramática é iniciado em um nodo do tipo  $a$ . Se não existe um nodo  $a$ , então um do tipo  $b$ , caso contrário um  $c$  é escolhido. Considere, por exemplo, o caractere 'E', Figura 1. 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'. A solução de [Cowell (1995)] é independente de translação, rotação e do tamanho dos caracteres a serem reconhecidos. Observa-se porém uma deficiência: os caracteres {O(letra), 0(zero), D}, {W, M}, {Z, 7,

J} e {6, 9} geram sentenças homônimas. Tentando resolver esse problema, derivou-se do conjunto de atributos base, dois outros subconjuntos de atributos. Com isto, minimizaram-se os problemas de conflitos mas, sem resolvê-los totalmente. Para garantir a solução completa, é necessário a aplicação de métodos heurísticos.

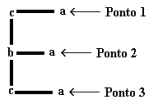


Figura 1 Nodos do caractere ‘E’.

### 3 Proposta de uma Nova Solução

A solução aqui apresentada visa reconhecer os caracteres da placa de um veículo segundo a atual legislação brasileira de trânsito, com sete caracteres (3 letras e 4 números). O sistema de reconhecimento é dividido em fases. A primeira, trata da aquisição da imagem do veículo, Figura 2.a. A segunda envolve o pré-processamento da imagem, Figura 2.b. A terceira, localiza a placa na imagem pré-processada, Figura 2.c, através de análise de contraste em uma determinada *região de busca* (região de maior probabilidade de ser encontrada a placa do veículo). Localizada a placa, a quarta fase é separar as imagens de cada caractere alfanumérico e antes do reconhecimento, realizar um outro pré-processamento sobre os mesmos para corrigir imperfeições e obter um esqueleto de cada um deles. A quinta, é a aplicação do reconhecedor propriamente dito. Na solução, serão utilizados dois grupos de atributos para construir a gramática: (1) o mesmo apresentado em [Cowell (1995)], os quais buscam propriedades no traçado dos caracteres; (2) referente à posição do atributo-traço do caractere (o caractere é colocado em uma grade com quatro quadrantes, Figura 3.a). Associa-se a cada atributo-traço uma posição quadrante. A grade tem orientação fixa mas, suas linhas têm posição variável, a qual depende das dimensões do caractere a ser processado. Conhecendo-se a altura (*h*) e a largura (*l*) do caractere, encaixa-se a grade sobre ele. A posição da linha vertical ( $P_V$ ) e horizontal ( $P_H$ ) da grade são dadas por:  $P_V=l/2+offset_V$  e  $P_H=h/2+offset_H$ . Os atributos que estiverem sobre a linha vertical pertencerão ao quadrante 2 ou 4, e se estiverem sobre a linha horizontal pertencerão ao quadrante 3 ou 4. Caso haja atributo sobre o cruzamento das duas linhas, ele pertencerá aos quadrantes 2, 3 e 4. As sentenças serão formadas conectando-se os atributos quadrantes com os do traço. As regras de concatenação são: (1) a ordem dos atributos quadrantes é 1, 2, 3 e 4, respectivamente; (2) como em um quadrante pode haver mais de um atributo-traço, o primeiro caractere será o atributo quadrante e após, todos os atributos traços pertencentes ao quadrante; (3) a ordem de concatenação dos atributos traços serão *a, b, c e d*, respectivamente.

A Figura 3.b mostra um exemplo, no qual a sentença formada será: *1abd2c3bcc4aab*. Como a grade é fixa em relação a rotação, para obter a independência da rotação do caractere é necessário haver mais de uma sentença na sua identificação, uma para cada posição de rotação do caractere. As posições de rotação são calculadas de 45° em 45°, iniciando em 0° até 315°. Como exemplo, montando as sentenças do caractere “E” considerando possíveis rotações de 45°, Figura 4, tem-se: (Rotação, Sentença) = {(0°, *1abc2a3a4c*), (45°, *1ac2ac4ac*), (90°, *1c2abc3a4a*), (135°, *1ac2ab3ac*), (180°, *1a2abc3c4a*), (225°, *2ac3ab4ac*), (270°, *1a2a3abc4c*), (315°, *1ac3ac4ab*)}



Figura 2 (a) Imagem original. (b) Imagem binária. (c) Placa isolada.



Figura 3 (a) Atributos posicionais. (b) Um exemplo de disposição de atributos-traços entre os quadrantes.

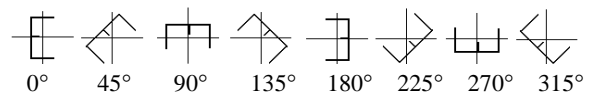


Figura 4 Exemplo de rotação com o caractere “E”.

Observa-se que, como qualquer uma dessas sentenças identifica o caractere “E”, localizada qual, encontra-se também o grau de rotação da placa. As possíveis ambigüidades entre outros caracteres são assim resolvidas.

### 4 Experimentos e Conclusões

O sistema se encontra em fase de desenvolvimento, e embora algumas imagens já estejam disponíveis, ainda não foi possível submetê-las ao sistema proposto. Os resultados obtidos serão posteriormente comparados com aqueles encontrados em [Cowell (1995)]. Como objetivo final, espera-se reconhecimento de cerca de 95% das placas, e tempo de resposta em frações de segundo.

### Referências Bibliográficas

J. R. Cowell, Syntactic Pattern Recognizer for Vehicle Identification Numbers, *Image and Vision Computing*, **13** (1995) 13-19.  
 R. Shalkoff, *Pattern Recognition*, John Wiley, 1992.  
 C. Pinho, Punição Eletrônica. *ISTOÉ*, **1374**(70) 31/01/96.