

PROCESSAMENTOS PARA IDENTIFICAÇÃO AUTOMÁTICA DE PLACAS DE VEÍCULOS USANDO UMA TÉCNICA SINTÁTICA

Airton Marco Polidório* e Díbio Leandro Borges†

RESUMO. Este artigo propõe um conjunto de métodos usados no reconhecimento automático de caracteres, aplicado na identificação de números de placas de veículos, usando uma técnica sintática. Todos os métodos de pré-processamentos e pós-processamentos, necessários à solução do problema, são descritos aqui. Os resultados obtidos apresentam alta qualidade e a performance do processamento é adequada.

Palavras-chave: processamento de imagem, reconhecimento de padrões, visão computacional.

PROCESSING METHODS FOR AUTOMATIC IDENTIFICATION OF VEHICLE LICENSE PLATES USING A SYNTACTIC TECHNIQUE

ABSTRACT. This study deals with a set of methods for recognizing automatically characters used to identify license plates by means of a syntactic technique. All preprocessing and postprocessing methods required to solve the problem are discussed. The results present high quality and an efficient processing performance.

Key words: image processing, pattern recognition, computer vision.

* Departamento de Informática, Universidade Estadual de Maringá, Av. Colombo, 5790, Câmpus Universitário, 87020-900, Maringá-Paraná, Brasil. E-mail: ampolido@din.uem.br

† Escola de Engenharia Elétrica (EEE), Universidade Federal de Goiás, Praça Universitária S/N, 74605 - 220, Goiânia-Goiás, Brasil. E-mail: dibio@eee.ufg.br

Correspondência para Airton Marco Polidório

Data de recebimento: 04/09/97

Data de aceite: 28/11/97

INTRODUÇÃO

A identificação automática de veículos através de suas respectivas placas de licença é importante em um grande número de aplicações: levantamento de dados de tráfego, gerenciamento de estacionamentos, controle de entrada e saída de *containers* de portos, controle de veículos pesados (fiscalização de massa transportada) em rodovias, controle de limite de velocidade em vias públicas (ruas, rodovias etc.), controle de avanço de sinal vermelho em semáforos, etc. Todas essas aplicações exigem que uma imagem completa do veículo seja visível. Esta exigência faz com que a placa do veículo ocupe uma pequena parte da imagem, resultando em caracteres com baixa resolução. A resolução típica dos caracteres é de 15x20 (aproximadamente 0.5% da área da imagem) *pixels*. Este fato traz tantas dificuldades que alguns pesquisadores como (Lotufo *et al.*, 1990), preferiram trabalhar com imagens, onde a placa ocupa 25% da imagem, limitando assim, o uso desse sistema a alguns tipos de aplicações.

Essa ferramenta automática deve resolver três problemas para justificar o seu uso: 1) localizar a placa do veículo; 2) segmentar os caracteres da placa em relação ao seu fundo; 3) reconhecer esses caracteres segmentados da placa, os quais identificam a placa do veículo infrator.

Este trabalho descreve as estratégias e os métodos que foram desenvolvidos para possibilitar o reconhecimento de caracteres alfanuméricos, que compõem a placa de um veículo, usando uma técnica sintática (Polidório e Borges, 1996) e faz-se também uma análise quantitativa e qualitativa dos resultados alcançados.

MÉTODOS E ESTRATÉGIAS USADAS

Tendo em vista os problemas que devem ser resolvidos/contornados objetivando atingir os resultados finais, são necessárias algumas simplificações e determinação de comportamentos padrões que tornem possível a elaboração/uso de métodos, junto com a adoção de certas estratégias.

Os métodos/estratégias aplicados neste trabalho, podem ser divididos em dois grupos:

1. aqueles usados na fase de pré-processamento das imagens, e
2. aqueles usados na fase de pós-processamento das imagens.

Nas seções a seguir, são mostrados os métodos e as estratégias que permitem resolver/contornar os problemas inerentes ao pré-processamento e ao pós-processamento das imagens.

Pré-processamentos e estratégias usadas

No grupo do pré-processamento, enquadram-se os seguintes métodos/estratégias: 1) marcar a região da placa (estratégia); 2) segmentar os caracteres em relação ao fundo da placa; 3) afinar os caracteres segmentados; 4) localizar os caracteres na placa (estratégia); 5) extrair os atributos pertencentes a cada caractere.

Marcando a região da placa

Os tipos de imagens, que estão envolvidas neste tipo de problema, são muito complexas, tanto em seus respectivos fundos que não são constantes, como no objeto de interesse secundário, o veículo que possui formas, partes salientes e reentrâncias, cor, textura, brilho, as quais variam muito de uma imagem para outra. Além de poder estar usando uma placa de identificação, objeto de interesse principal, entre os seis padrões possíveis, sujeitas às interferências do ambiente.

Após extenso treinamento/testes, com aproximadamente 10.000 imagens, de um método baseado em lógica fuzzy, (Nijhuis *et al.*, 1995) conseguiram, de forma automática, localizar a placa do veículo na imagem. Entretanto, as imagens usadas no treinamento tinham um único padrão de cor: caracteres em preto e fundo em amarelo. Mesmo assim, em 25% dos casos, o método proposto não consegue localizar a placa do veículo. Nestas circunstâncias, vê-se que este problema precisa de melhores avaliações para se obter uma solução que produza resultados mais quantitativos.

Em consideração a estes fatos, espera-se justificar a opção de segmentação manual da placa de um veículo em uma imagem usada neste trabalho.

De posse de uma imagem de veículo, (Figura 1-a), o primeiro passo, é marcar a região da placa do mesmo (Figura 1-b).

Esta marcação não é feita de forma automática, e a qualidade dos resultados dos processamentos posteriores depende da subjetividade da região marcada como sendo a área da placa, uma vez que todos os processamentos posteriores ocorrerão somente dentro dessa região. Em muitos casos, pode haver certa folga na escolha dos limites dessa região.

Entretanto, convém salientar que os processamentos posteriores produzem melhor resultado, se a área considerada representar a placa, e somente ela, em sua totalidade.



Figura 1. (a) Imagem original de um veículo. (b) Área da placa do veículo marcada.

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.

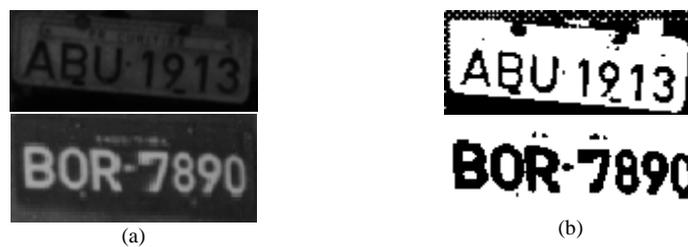


Figura 2. Exemplos da segmentação de caracteres de placas de veículos. (a) Imagens originais. (b) Caracteres segmentados.

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 2.

Afinando os caracteres segmentados

A segmentação dos caracteres em relação ao fundo da placa, permite a pesquisa dos seus respectivos atributos tipo traço e posicional. Para fazer a pesquisa dos atributos é necessário seguir o traçado das linhas que compõem o caractere. Para facilitar esta ação, 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; 4) produzir um esqueleto com formato parecido com o do objeto original; 5) minimizar o efeito de geração de ramos em linhas com reentrâncias e saliências; 6) eliminar o maior número de pixels redundantes possível.

Considerando essas características, o algoritmo escolhido foi o proposto por (Zhang e Suen, 1984). (Bushofa e Spann, 1997) preferiram usar uma variação deste. (Cowell, 1995), usou o algoritmo de Yokoi descrito em (Tamura, 1978).

O algoritmo de (Zhang e Suen, 1984) é relativamente simples de ser implementado, com tempo de processamento baixo, quando aplicado a caracteres com as dimensões tratadas neste trabalho (15 x 20 *pixels* em média), adiciona-se ao seu desempenho a qualidade do resultado que o mesmo produz. Os caracteres afinados têm boa qualidade devido ao fato desse algoritmo não ser muito sensível às saliências e reentrâncias muito comuns nos caracteres binarizados.

O algoritmo remove os *pixels*, que contornam o objeto, exceto aqueles que compõem o esqueleto do objeto, de forma iterativa. Para preservar a conectividade, a remoção dos *pixels* é dividida em duas subiterações.

Quando aplicado ao problema em questão, o algoritmo de Zhang-Suen, produz bons resultados. A Figura 3 mostra um desses resultados.



Figura 3. Os caracteres segmentados, afinados pelo algoritmo de Zhang-Suen.

Localizando os caracteres na placa

A busca pelo primeiro caractere da placa é feita através de uma varredura seqüencial na direção horizontal no sentido da esquerda para a direita pelo meio da altura da região marcada (a qual se espera ser a placa). Com este procedimento, evita-se o processamento desnecessário das informações irrelevantes (nome da cidade, estado) contidas na parte superior da placa.

Os caracteres, em média, têm dimensões de 15×20 pixels, exceto a letra I e o número 1. Observou-se na prática, que as interferências ruidosas e informações irrelevantes, contidas nas placas, formam blocos homogêneos com no máximo 6×6 pixels, assim, todos os esqueletos que tiverem essas dimensões, ou menores, são descartadas durante o processamento.

Encontrado o esqueleto do primeiro caractere (letra) da placa, o mesmo é processado, *pixel a pixel*, em busca dos atributos desejados e de suas respectivas localizações, bem como os valores da altura e da largura do caractere. Esses atributos e valores de dimensão são usados para montar a sentença sintática, a qual permite o reconhecimento do caractere. Para localizar o próximo caractere, continua-se a operação de varredura descrita anteriormente, com a diferença de que, agora, é feita pelo meio vertical do caractere que foi processado anteriormente. Com isso, evitam-se problemas em encontrar os próximos caracteres no processamento de placas com certo grau de inclinação (Figura 4). Este procedimento é repetido para os próximos caracteres.

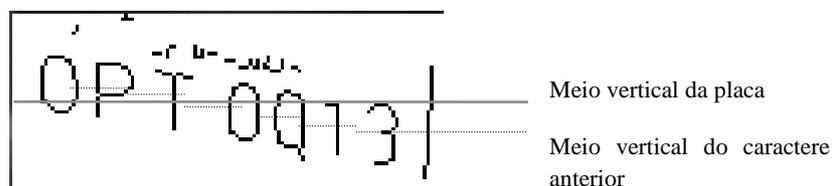


Figura 4. Ilustração da estratégia usada para localização dos caracteres em uma placa.

Localizando e extraíndo os atributos

Cada caractere é formado por um conjunto de atributos gráficos os quais, guardam uma relação espacial específica dentro do contexto da grafia de cada caractere. Os atributos gráficos relevantes para este trabalho são aqueles descritos em (Polidório, 1997) chamados de atributos do tipo traço e atributos do tipo posicional.

Para localizar e extrair os atributos desejados pertencentes a um caractere, é necessário seguir as linhas de seu esqueleto. O ponto inicial de entrada para este processo é qualquer *pixel* do primeiro caractere da placa. Para cada *pixel* processado, verifica-se a existência de direções alternativas a serem tomadas. Se existirem, elas são empilhadas para processamento posterior.

Deve-se ter em conta a necessidade de medir dois atributos: os atributos do tipo traço e os atributos do tipo posicional. Como os atributos do tipo 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 em um caractere, é medida a sua posição (linha e coluna) em relação aos valores máximos e mínimos de cada dimensão (altura e largura) do caractere. Estes valores é que permitirão a obtenção dos eixos da grade para determinação dos atributos do tipo posicional.

Os atributos do tipo traço, exceto os atributos do tipo *c* e *e*, são determinados no momento do processamento de cada *pixel* do esqueleto. É desnecessário o uso de máscaras, ou avaliar caminhos alternativos a partir dele.

(Tamura, 1978) definiu 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 ponto (Tabela 1).

Tabela 1. Os números de transições preto/branco em torno de um *pixel* e as suas relações com os atributos do tipo traço.

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

Todos os atributos do tipo traço (*a*, *b* ou *d*) diagnosticados (Figura 5), são guardados em uma pilha junto com as suas respectivas coordenadas, (linha, coluna), em relação às dimensões da imagem. Isto é necessário para processamento posterior, quando serão encontradas as relações espaciais entre os atributos do tipo traço e os atributos do tipo posicional.

Os processamentos para extração dos atributos tipo traço *c* e *e* são mais complexos. Após o processamento dos outros atributos do tipo traço e posicional de um caractere, é verificada a real necessidade em se extrair os atributos do tipo traço *c* e *e* do mesmo. Se houver a necessidade, então, o caractere é reprocessado. Este novo processamento tem início na mesma posição de um atributo traço do tipo *a*, e para os caracteres que não possuem este atributo, (D, O e 0), o início do reprocessamento se dá no primeiro segmento do caractere que estiver no meio da altura do mesmo.



Figura 5. Exemplos de atributos do tipo traço localizados pelas relações propostas por (Tamura, 1978). (a) Atributos do tipo *a*. (b) Atributos do tipo *b*. (c) Atributos do tipo *d*.

Escolhido o ponto de partida para o reprocessamento, a partir deste é adotada uma direção, entre as possíveis, para seguir o esqueleto do caractere. Esta direção é adotada como sendo uma direção base.

A partir do *pixel* *P*, as possíveis direções são rotuladas com números entre 0 (zero) e 7 (sete), e segue o esquema da Figura 6.

Para determinar se existe uma mudança de direção em ângulo de 90°, basta calcular a diferença entre a direção base e a direção possível do

próximo *pixel*. Se a diferença entre essas duas direções, em valores absolutos, produzir o valor 2 ou 6, a mudança de direção em ângulo de 90° fica caracterizada, caso contrário, é tomada a direção do próximo *pixel* do esqueleto, ainda não processado, e mantendo-se a direção base original, repetem-se as ações descritas neste parágrafo.

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 6. O pixel *P* e seus 8-vizinhos.

Quando é localizada uma mudança de direção em ângulo de 90°, as coordenadas e o rótulo desse atributo do tipo traço são armazenados na pilha de atributos, onde já estão os outros atributos do tipo traço, para processamento posterior. Mas, se ainda não foi processado todo o esqueleto do caractere, então, esta última direção, que provocou a mudança de direção em 90°, é assumida como sendo a nova direção base, e a direção do próximo *pixel* do esqueleto, ainda não processado é tomada, e as ações descritas no parágrafo anterior são novamente executadas.

Confirmada a mudança de direção em ângulo de 90°, deve-se decidir se o atributo traço encontrado é do tipo *c* ou *e*.

A diferença entre esses dois atributos é sutil, e é medida em *pixels*. Toda mudança de direção em ângulo de 90° passa por uma transição que pode ser de um ou mais *pixels*. Se esta transição ocorrer em até no máximo dois *pixels*, diz-se ter um atributo do tipo *c*, caso contrário, tem-se um atributo do tipo *e*, Figura 7.



Figura 7. Exemplos de atributos traço com mudança de direção de 90°, detectados pelo método proposto. (a) Tipo *c*. (b) Tipo *e*.

Pós-processamentos e estratégias usadas

No grupo do pós-processamento, enquadram-se os seguintes métodos/estratégias: 1) relacionar espacialmente os atributos extraídos; 2) reconhecer os caracteres da placa; 3) tratar os caracteres não reconhecidos pelo sistema (estratégia); 4) verificar a ocorrência de sinônimos (estratégia); 5) identificar a placa.

Construindo as sentenças

Após o término do processamento do esqueleto de cada caractere, no qual são extraídos os atributos do tipo traço e posicional, é montada a sentença sintática que permitirá a sua identificação nas respectivas bases de dados.

Através do uso das equações sugeridas em (Polidório e Borges, 1996) e (Polidório, 1997), as quais recebem como parâmetros os valores da altura (h) e da largura (l) do esqueleto do caractere, são calculadas as posições dos eixos da grade, que dividem o esqueleto do caractere em quatro quadrantes.

De posse das equações de reta que representam esses eixos, a pilha de atributos do tipo traço que foram extraídos do esqueleto do caractere, juntamente com as suas respectivas localizações na imagem, é processada, e usando as regras sintáticas definidas em (Polidório, 1997), são definidos os quadrantes onde cada atributo do tipo traço está localizado. Os atributos do tipo traço pertencentes a cada quadrante são ordenados em ordem alfabética crescente formando uma cadeia de caracteres. Concatena-se então, o atributo posicional (número do quadrante) com sua respectiva cadeia de caracteres de atributos traços ordenados, formando a relação dos atributos do tipo posicional com os atributos do tipo traço em cada quadrante. Por fim, concatenam-se estas relações em uma única cadeia de caracteres uma a uma na ordem preestabelecida dos respectivos quadrantes (1 ao 4). Obtendo-se portanto, a sentença sintática do caractere que foi processado.

Identificando o caractere

De posse da sentença sintática do caractere que foi processado, faz-se a pesquisa em uma base de dados previamente definida na fase de treinamento do sistema, a qual contém as sentenças sintáticas dos caracteres junto com o respectivo caractere ASCII que ela representa. A

identificação do caractere é feita quando for encontrada uma sentença na base de dados idêntica àquela gerada pelo caractere imagem.

Para reconhecer os caracteres, cabe a aplicação de uma estratégia: a forma de disposição espacial dos caracteres que compõem as placas dos veículos é sempre a mesma. Os três primeiros caracteres são letras, combinadas de A a Z, e os quatro últimos são números, combinados de 0 a 9. Este fato, proporciona uma informação contextual muito forte: os padrões são separados em duas classes pela própria forma de disposição espacial. Assim, a base de dados é dividida em duas partes: uma que armazena os dados dos números e outra os dados das letras. Como, sempre se sabe qual a natureza do caractere, letra ou número, que está sendo processado, evitam-se os problemas de geração de sinônimos entre as letras e os números, como por exemplo: B e 8, I e 1, e O (letra) e 0 (número).

Caso, sejam geradas sentenças que não pertençam às bases de dados, o sistema gera o caractere "?", indicando que aqueles caracteres da placa não puderam ser identificados, e que as sentenças geradas por eles devem ser incorporadas em suas respectivas bases de dados, juntamente com os caracteres ASCII que elas representam, para que sejam contempladas em uma próxima ocorrência. Estas operações são resumidas nos seguintes passos do algoritmo original:

para cada caractere-imagem (i) da placa (i = 1..7)

faça:

relacionar os atributos extraídos do caractere-imagem(i);

montar a sentença sintática(i) do caractere-imagem(i);

se $i \leq 3$

então procure a sentença sintática(i) na base de dados das letras;

senão procure a sentença sintática(i) na base de dados dos números;

se houve sucesso na busca da sentença sintática(i)

então recuperar o caractere-ASCII(i) que está associado com a sentença sintática(i);

senão caractere-ASCII(i) = "?";

concatenar o caractere-ASCII(i) à *string* placa;

Estratégia para tratamento dos caracteres não reconhecidos

As sentenças geradas por caracteres que não são contempladas nas bases de dados, levam ao não reconhecimento dos mesmos. A resposta

do sistema para estes casos, é a geração do caractere “?” em substituição aos caracteres que não foram reconhecidos.

Quando uma placa está sendo processada, e ocorrem nela caracteres não reconhecidos, o caractere “?” serve de aviso para o sistema de que sentenças não contempladas nas bases de dados ocorreram. Isto significa que caracteres com diferenças estruturais àqueles das bases de dados estão presentes nesta placa. Essas diferenças estruturais podem ser provocadas por dois motivos básicos: 1) os caracteres, apesar do mesmo significado gráfico, possuem reais diferenças estruturais; 2) as diferenças estruturais podem ter sido provocadas por agentes de interferências.

Quando ocorre o caso 1), a medida a ser tomada para contornar o problema é a inserção da sentença sintática, juntamente com o caractere ASCII que ela representa na sua respectiva base de dados.

O caso 2) é mais complexo de ser tratado. Veja, como exemplo, o caso de caracteres com defeitos na pintura de seu interior.

É comum haverem falhas na pintura dos caracteres da placa, principalmente naquela fixada no pára-choques dianteiro do veículo, causada pela ação do tempo, do vento e, por colisões (poeira, pedregulhos etc.), deixando-os vazados em alguns pontos (Figura 8-a). Se a área afetada for considerável, essas falhas são intensificadas no processo de segmentação dos caracteres (Figura 8-b).

As falhas observadas no interior dos caracteres da Figura 8-b, são propagadas para os processos de afinamento e extração de atributos, fazendo com que atributos não esperados para aquele caractere específico apareçam, Figura 8-c, tornando-o irreconhecível para o sistema.

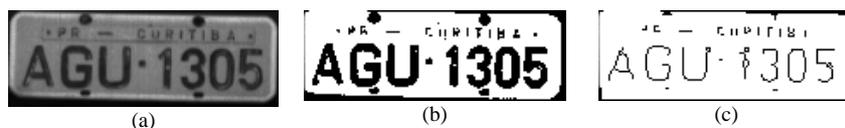


Figura 8. (a) Exemplo de caracteres com falhas de pintura no interior. (b) Imagem binária realçando as falhas no interior dos caracteres. (c) Caracteres afinados.

Entretanto, é possível corrigir essas falhas no interior dos caracteres. Uma das maneiras de se fazer isto é através da aplicação do operador de fechamento morfológico, definido na morfologia matemática (Serra, 1982) e (Serra, 1988).

O problema é determinar, de forma automática, quando é necessária tal operação. Uma solução, paliativa, para este problema, é a análise do sucesso no reconhecimento de todos os caracteres da placa. Se pelo menos um caractere da placa não for reconhecido então, executa-se a operação de fechamento. Se após a aplicação da operação de fechamento, não se conseguir o reconhecimento do(s) caractere(s) então, eles são tidos como não reconhecidos de forma automática, exigindo a identificação manual.

A estratégia para a aplicação da operação de fechamento é descrita através dos seguintes passos: 1) os caracteres afinados, Figura 8-c, são fechados pelo elemento estruturante em cruz (Figura 9-a). A operação de fechamento deve ser realizada sobre caracteres já afinados. Este é um detalhe importante pois, evita o estabelecimento de conexões entre os caracteres e dos mesmos com os parafusos/buracos da placa, bem como perdas de atributos, principalmente os cruzamentos de linhas; 2) o resultado produzido pelo passo 1), Figura 9-b, é novamente afinado para eliminação dos *pixels* excedentes, inseridos na operação de fechamento; 3) os caracteres afinados, Figura 9-c, são submetidos ao módulo extrator de atributos, os quais serão processados no módulo do reconhecedor.

Se após a aplicação destes procedimentos, o sistema insistir em não reconhecer os caracteres então, as sentenças sintáticas geradas pelos mesmos devem ser incorporadas em suas respectivas bases de dados para que sejam contempladas em uma próxima ocorrência.

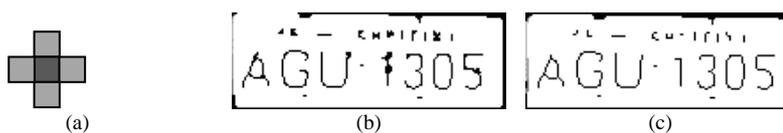


Figura 9. (a) Elemento estruturante em cruz; (b) caracteres fechados com o elemento estruturante em cruz; (c) caracteres afinados.

Tratamento dos Sinônimos

Alguns caracteres, apesar de terem formas gráficas bem diferentes, possuem os mesmos atributos do tipo traço, os quais podem se relacionar espacialmente da mesma forma. Outros, como as letras D e O, são muito parecidos, em sua forma gráfica, e as diferenças entre seus atributos do tipo traço são mínimas. Estes fatos podem levar o sistema a gerar

sentenças homônimas para caracteres com significado diferente e assim, reconhecê-los de forma errônea.

No caso de caracteres muito parecidos, como as letras D e O, optou-se pela remoção das sentenças que provocam sinônimos da base de dados. Assim, toda vez que o sistema gera uma dessas sentenças, a sua resposta será sempre o caractere “?” o qual estará substituindo uma letra D, ou O, obrigando o reconhecimento manual da placa.

Entretanto, para os caracteres que têm formas gráficas bem diferenciadas, mas que possuem os mesmos atributos do tipo traço, os quais se relacionam espacialmente da mesma forma, existe um tratamento automático para contornar o problema da geração de sentenças homônimas.

Durante a etapa de treinamento do sistema, observou-se que o problema de geração de sentenças homônimas podia ser contornado se houvesse conhecimento *a priori* de que aquela sentença específica gerava homônimo(s).

Para adquirir esse conhecimento, a base de dados dos números e das letras foram divididas, cada uma, em duas partes. Uma parte que contém as sentenças que identificam os respectivos caracteres e, uma outra, que contém informações sobre as sentenças homônimas.

Assim, quando um caractere da placa é processado e sua sentença sintática, a qual permite o seu reconhecimento, é gerada, a mesma, primeiro, é pesquisada na respectiva base de dados, números ou letras, que contém as sentenças homônimas. Se esta não for encontrada nessas bases então, inicia-se a pesquisa na respectiva base de dados que identifica o caractere, significando que a sentença gerada não é homônima de nenhuma outra, pelo menos até este instante.

Quando uma sentença homônima ocorre em uma base de dados de sentenças sinônimas, o sistema é avisado de tal fato, exigindo dele as providências para contornar o problema.

As providências a serem tomadas são simples, e são baseadas na modificação do relacionamento espacial existente entre os atributos do tipo traço.

Este procedimento é possível, graças à flexibilidade dos atributos do tipo posicional. Conhecidas as dimensões do caractere que está sendo processado, são calculadas as posições dos eixos vertical e horizontal da grade que delimita os quatro quadrantes, os quais formam o conjunto dos atributos do tipo posicional, usando as equações propostas em (Polidório

e Borges, 1996). Para modificar o posicionamento dos eixos, basta recalculá-los usando novos valores para os parâmetros *offsets* aplicados nas equações.

Os valores de *offsets* padrões usados até este ponto eram $0.1l$ e $0.1h$ para os parâmetros $offset_v$ e $offset_h$, respectivamente. Então, para estes casos, buscaram-se novos valores de *offsets* que, obrigatoriamente provoquem mudanças nos relacionamentos espaciais entre os atributos do tipo traço do caractere. Observe que quando uma sentença homônima ocorre, tem-se o conhecimento do grupo de caracteres que podem gerar aquela sentença. Isso, facilita a tarefa de determinação dos novos valores de *offsets*.

Os registros que formam as bases de dados das sentenças sinônimas, possuem as seguintes informações: a sentença sintática (sinônima) e os dois valores de *offsets*, que devem ser usados para recalculá-los o novo posicionamento dos eixos da grade.

Com o novo posicionamento dos eixos da grade, uma nova sentença sintática é montada da forma já descrita neste trabalho, e uma nova pesquisa deve ser feita, como descrito nesta seção, começando novamente pela base de dados das sentenças sinônimas, pois, podem ocorrer novos sinônimos.

Para melhor esclarecer, observe os casos reais de homônimos existentes entre algumas formas gráficas dos caracteres {H, M e K} e {T, Y e V} depois de afinados. Quando os eixos da grade, calculados com os valores $0.1l$ e $0.1h$ para os parâmetros $offset_v$ e $offset_h$, respectivamente, são colocados sobre os caracteres {H, M e K}, Figura 10, as sentenças geradas em cada caso é a mesma $1ab2ab3a4a$, resultando na sinonímia entre estes caracteres.

Sabe-se, *a priori*, quando esta sentença ocorre, que a mesma está se referenciando somente um, entre os três caracteres possíveis. Assim, para haver o desempate, há a necessidade em recalculá-los a posição dos eixos da grade.

No caso específico para as letras H, M e K, a nova posição dos eixos é recalculada usando os valores $0.2l$ e $-0.3h$ para os parâmetros $offset_v$ e $offset_h$, respectivamente, o que resulta nas expressões: $EIXO_v = 0.7l$ e $EIXO_h = 0.2h$, quando aplicadas às equações da grade. Com este novo posicionamento dos eixos da grade, a sentença sintática obtida para cada caractere em questão, Figura 11, fica da seguinte forma: (H, $1a2a3ab4ab$), (M, $1ab2ab3a4a$) e (K, $1a2a3a4abb$).

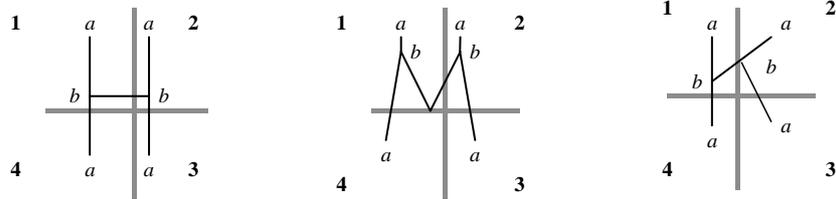


Figura 10. Ilustração do caso de geração de sentenças homônimas entre as letras H, M e K.

Resolvendo, portanto, o problema de sentenças sinônimas entre esses três caracteres. Um caso interessante de geração de sentenças sinônimas envolve os caracteres T, Y e V. Como se pode observar na Figura 12, quando se usam os valores $0.1l$ e $0.1h$ para os parâmetros $offset_v$ e $offset_h$, respectivamente, a sentença gerada para cada caractere é: (T, $1ab2a34a$), (Y, $1ab2a34a$) e (V, $1a2a34ab$).

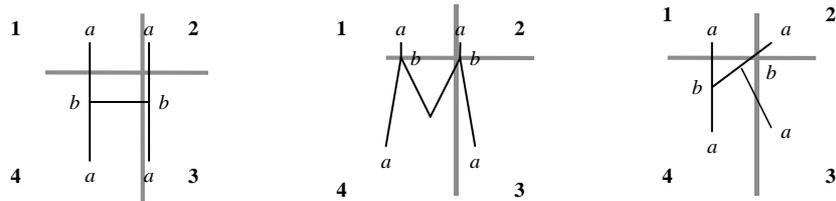


Figura 11. Ilustração do caso de geração de sentenças homônimas entre as letras H, M e K, com posicionamento dos eixos da grade recalculado.

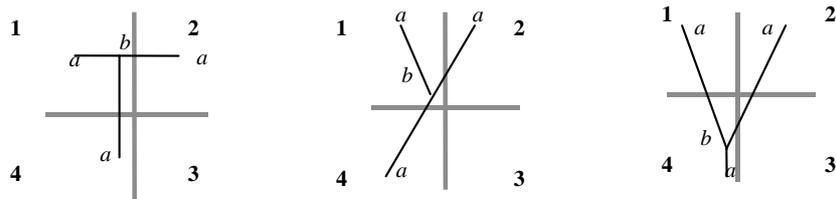


Figura 12 Ilustração do caso de geração de sentenças homônimas entre as letras T, Y e V.

Nota-se, que a sinonímia existe, nesta situação, somente entre as letras T e Y. Entretanto, para reposicionar os eixos da grade para resolver este sinônimo, entre o T e o Y, é usada a expressão $EIXO_h = 0.2h$ e isto, faz com que as novas sentenças sintáticas para esses caracteres sejam $1ab2a34a$ e $1a2a34ab$ para a letra T e Y, respectivamente. Note que

agora existe o sinônimo entre a letra Y e a letra V. Se o caractere que estiver sendo processado for a letra Y, haverá a necessidade de um novo posicionamento dos eixos da grade para efetuar o desempate entre os dois caracteres.

ANÁLISE DE RESULTADOS E CONCLUSÕES

Tendo em vista os conteúdos anteriormente apresentados neste trabalho, procura-se, agora, analisar os resultados obtidos, avaliando os aspectos qualitativos e quantitativos destes. Para prover subsídios e dados que possibilitassem essas avaliações, foram submetidas 135 placas de veículos, obtidas em condições ambientais variadas e abrangendo os diversos padrões, tamanhos e graus de inclinação, aos processamentos do sistema, em sua fase de testes.

O primeiro processamento importante é a segmentação dos caracteres em relação ao fundo da placa pois, para se obter resultados efetivos nos processamentos posteriores, é necessário que este processo produza resultados de boa qualidade. Entretanto, ressalta-se, que a aplicação deste método produziu resultados para todas as placas processadas, mas nem sempre esses resultados tinham a qualidade mínima para prosseguirem nas etapas posteriores de processamento exigidos pelo sistema de identificação de placas de veículos. Das 135 placas submetidas à segmentação, somente em 4 casos não se produziram os resultados com a qualidade desejada. Isto implica que a Taxa de Segmentação efetiva do método é de aproximadamente 97%.

O afinamento dos caracteres, algoritmo de (Zhang e Suen, 1984), provê facilidades algorítmicas para localizar e rotular cada um dos atributos do tipo traço. Entretanto, este processo é o gargalo do sistema, pois, é ele que demanda maior tempo de processamento, mais que todos os outros processamentos juntos. Contudo, dentro da metodologia aplicada para solucionar o problema de reconhecimento sintático de caracteres de placas de veículos, este tornou-se uma peça fundamental.

O método usado para extrair os atributos do tipo traço dos esqueletos dos caracteres, descrito em (Tamura, 1978), mostrou-se dotado de grande eficiência, produzindo resultados altamente confiáveis com baixo custo computacional.

Considerando que das 135 placas submetidas ao segmentador de caracteres 131 tinham qualidade para prosseguir no processo, então, o

sistema foi testado com, exatamente, 917 caracteres (393 letras e 524 números). Destes, 816 (88,99%) foram reconhecidos corretamente, 25 foram reconhecidos de forma incorreta (2,73%) e 76 foram rejeitados (8,28%).

A taxa de rejeição é um valor considerado alto, o ideal seria em torno de 1%. Entretanto, a rejeição de um caractere submetido ao reconhecimento significa que o mesmo não foi reconhecido mas, também, não foi confundido com nenhum outro. Isto se deve ao fato destes caracteres possuírem uma forma gráfica diferente ou, terem sofrido algum tipo de interferência que adicionaram ou removeram características dos mesmos ou, ainda, podem ter sofrido uma rotação suficiente a ponto de mudar a forma do relacionamento espacial entre seus atributos. Esses fatos podem gerar sentenças sintáticas não esperadas para aqueles caracteres e, desta forma, estas não são contempladas nas respectivas bases de dados de reconhecimento.

A Tabela 2 mostra o resumo das avaliações feitas em vários sistemas. Os valores assinalados com o caractere “?” não puderam ser avaliados por falta de informações complementares nos respectivos trabalhos.

Tabela 2. Uma comparação entre os sistemas de reconhecimento de placas.

Referência	Taxa de Reconhecimento (%)	Taxa de Erro (%)	Taxa de Rejeição (%)
(Davies <i>et al.</i> , 1990)	61,7	11,2	27,1
(Williams <i>et al.</i> , 1989)	63,0	37,0	?
(Lotufo <i>et al.</i> , 1990)	76,0	?	?
(Lisa <i>et al.</i> , 1993)	95,5	0,9	3,6
(Föhrand e Raus, 1994)	98,2	1,8	0,0
(Yoo <i>et al.</i> , 1994)	95,0	5,0	0,0
(Tanabe <i>et al.</i> , 1994)	98,2	?	?
(Abe e Lan, 1993)	96,8	3,2	0,0
(Kanayama <i>et al.</i> , 1991)	90,0	?	?
(Hwang <i>et al.</i> , 1992)	97,0	1,0	2,0
(Kertész <i>et al.</i> , 1994)	94,0	?	?
(Cowell, 1995)	95,0	?	?
(Nijhuis <i>et al.</i> (1995))	98,51	0,02	1,47
Este trabalho	88,99	2,73	8,28

O desempenho das partes do sistema é medido em unidade de tempo de processamento (utp) ou seja, o tempo gasto no processamento de determinados processos foram medidos em unidades de *clock* do processador e podem ser observados na Tabela 3. A coluna de Processos indica quais os processos tiveram seus respectivos desempenhos avaliados, os quais são: a segmentação dos caracteres em relação ao

fundo da placa, o afinamento dos caracteres segmentados e o processamento de atributos, as construções das sentenças sintáticas e o reconhecimento de cada caractere de uma placa e a identificação da placa. A coluna T_{utp} relaciona o tempo médio gasto no processamento, em utp, de cada um dos processos. A coluna T_r refere-se ao tempo relativo gasto em cada processo. A coluna T_{min} refere-se ao caso em que aquele processo levou menos tempo para ser processado, e a coluna T_{max} ao caso em que o processo levou mais tempo para ser processado.

Tabela 3. Relação de processos e seus respectivos desempenhos.

Processos	T_{utp}	$T_r = T_{\text{utp}} / 100$	T_{min}	T_{max}
Segmentação dos caracteres em relação ao fundo da placa	100	1,0	50	120
Afinamento dos caracteres segmentados	2248	22,5	1700	3200
(Processamento dos atributos + construção da sentença + reconhecimento) de cada caractere + identificação da placa	400	4,0	160	550

REFERÊNCIAS BIBLIOGRÁFICAS

- BUSHOFA, B.M.F. & SPANN M. Segmentation and recognition of arabic characters by structural classification. *Image and Vision Computing*, 15:167-179, 1997.
- COWELL, J.R. Syntactic pattern recognizer for vehicle identification numbers. *Image and Vision Computing*, 13:13-19, 1995.
- DAVIES, P.P., EMMOTT, N. & AYLAND, N. License plate recognition technology for tool violation enforcement, IEE COLLOQUIUM ON IMAGE ANALYSIS FOR TRANSPORT APPLICATIONS, 35:7/1-7/5, 1990.
- FÖHRAND, R. & RAUS, M. Automatisches lesen amtlicher Kfz-Kennzeichen. *Elektronik*, 1:60-64, 1994
- HWANG, C., SHU, S., CHEN, W. & WEN, K. A PC-Based license plate reader. Proceedings Vision Applications, Architectures and Systems Integration, Boston, SPIE 1823:272-283, 1992,
- JANG, B.K. & CHIN, R.T. Analysis of thinning algorithms using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):541-551, 1990.
- KANAYAMA, K, FUJIKAWA, Y., FUJIMOTO, K. & HORINO, M. Development of vehicle-license plate recognition system using real-time image processings its application to travel-time measurement. IEEE VEHICULAR TECHNOLOGY CONFERENCE, 41, St. Louis. p.798-804, 1991.
- KERTÉSZ, A., KERTÉSZ, V. & MÜLLER, T. An on-line image processing system for registration number identification. IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS, 6, Orlando. p.4145-4147, 1994.

- LISA, F., CARRABINA, J., PÉREZ-VINCENTE, C., AVELLANA, N. & VALDERRAMA, E. Two-bit weights are enough to solve vehicle license recognition problem. INTERNATIONAL CONFERENCE ON NEURAL NETWORKS, 3, San Francisco. p.1242-1246, 1993.
- LOTUFO, R.A., MORGAN, A.D. & JOHNSON, A.S. Automatic number-plate recognition. IEE COLOQUIUM ON IMAGE ANALYSIS FOR TRANSPORT APPLICATIONS, 35, p.6/1-6/6, 1990.
- NIJHUIS, J.A.G., BRUGGE, M.H., HELMHOLT, K.A., PLUIM, J.P.W. SPAANENBURG, L. VENEMA, R.S. & WESTENBERG, M.A. Car license plate recognition with neural networks and fuzzy logic. IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS, p.2232-2236, 1995.
- POLIDÓRIO A.M. Segmentação e reconhecimento de caracteres para identificação de placas de veículos. Curitiba, 1997. Dissertação Mestrado em Engenharia Elétrica e Informática Industrial) - Cefet-Pr.
- POLIDÓRIO A.M. & BORGES, D.L. Reconhecimento de caracteres para identificação de placas de veículos. SIBIGRAPI, 9, Caxambu. p.363-364, 1996.
- SERRA, J. *Image analysis and mathematical morphology*. London: Academic Press, 1982.
- SERRA, J. *Image analysis and mathematical morphology: theoretical advances*. London: Academic Press, 1988.
- TAMURA, H.A Comparison of line thinning algorithms from digital geometry viewpoint, proc. INT. CONF. PATTERN RECOGNITION, 4, p.715-719, 1978.
- TANABE, K., MARUBAYASHI, E., KAWASHIMA, H, NAKANISHI, T. & SHIO, A. PC-Basead Car License Plate Reading. *Image and Video Processing II, SPIE*, 2182:220-231, 1994.
- WILLIAMS, P., KIRBY, H., MONTGOMERY, F. & BOYLE, R. Evaluation of video-recognition equipment for number-plate mathing. INTERNATIONAL CONFERENCE ON ROAD TRAFFIC MONITORING, 2, p.89-93, 1989.
- YOO, J., CHUN, B. & SHIN, D. A Neural network for recognizing characters extracted from moving vehicles. WORLD CONGRESS ON NEURAL NETWORKS, p.162-166, 1994.
- ZHANG, T.Y. & SUEN, C.Y. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3):236-239, 1984.