

# Reconhecimento de Placas de Veículos

Arthur Ribeiro<sup>1</sup>, João Batista Matos<sup>1</sup>, Vinícius Coelho<sup>1</sup>

<sup>1</sup>Instituto de Informática – Universidade Federal de Goiás (UFG)  
Goiânia – GO – Brasil

arthurfranklin@hotmail.com, {joaobatistamatos, viniciuscoelho}@inf.ufg.br

**Abstract.** *Over the years, the society needed resources that would aid in traffic. With this work, we try to create a system that optimizes the traffic control that is so necessary in today's urban centers.*

**Resumo.** *Com o passar dos anos, a sociedade necessitou de recursos que auxiliassem no trânsito. Com esse trabalho, tentamos criar um sistema que otimize esse controle de tráfego que é tão necessário nos centros urbanos de hoje.*

## 1. Introdução

Com o constante crescimento da frota mundial de veículos e o aumento da intensidade dessa frota nos centros urbanos, tornou-se cada vez mais necessário o controle do trânsito dos automóveis, afim de se manter a ordem e até mesmo a segurança de determinados locais caso algum imprevisto aconteça. O reconhecimento de placas hoje é útil para várias áreas distintas, como a de controle e monitoramento de tráfego, localização de carros roubados, aplicação de multas aos condutores infratores, controlar o acesso a estacionamentos, coletar estatísticas de fluxo de entrada e saída de certos locais ou mesmo gerar subsídios para a área de turismo. Mesmo nas grandes metrópoles brasileiras, a identificação do

veículo, a partir das fotografias dos fiscalizadores eletrônicos, ainda é feita manualmente e não em tempo real. E otimizar esse processo já é uma necessidade para o Brasil e muitos outros países, é um problema que vem recebendo crescente atenção na comunidade de Reconhecimento de Padrões devido a sua praticidade e visibilidade de resultados diretos, logo a criação de um sistema de reconhecimento automático de placas se tornaria um aliado a mais da sociedade no trânsito.

## 2. Reconhecimento e Localização da “Placa”

As placas dos automóveis possuem padrões impostos a todos os veículos circulantes. Por conta dessa padronização o sistema deve procurar reconhecer a placa em uma imagem através de análises como a forma da placa, a concentração ou os agrupamentos de cores, a textura, as sequências de caracteres pré-determinadas ou ainda as assinaturas e as variações em escala cinza em determinado ponto da imagem. Além da procura por padrões, os algoritmos de segmentação aplicados às imagens devem ser capazes de contornar problemas de visualização das placas, como neblina na imagem ou deterioração da placa pelo tempo. Claro na medida do possível, tal qual como os seres humanos só identificam algo até um certo limite. A “Figure 1” mostra a padronização vigente para as placas de automóveis brasileiros.

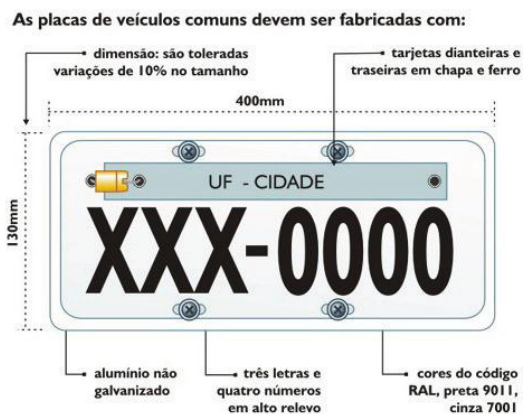


Figure 1. Padrão de placas nos automóveis brasileiros

Visto que as placas possuem características geométricas exclusivas, como forma e proporções de comprimento e altura, o uso de técnicas de segmentação focadas em operadores geométricos mostra-se bastante interessante. Assim, a busca pela placa do automóvel em uma imagem resume-se na procura por uma figura geométrica retangular a qual respeite as características de uma placa de automóvel. Afim de resolver problemas de perspectiva a bibliografia especializada cita especificamente três abordagens as quais atingem tais objetivos. São elas: a translação, a rotação e o “scaling” da imagem, que nada mais são do que formas de promover o reposicionamento da placa na imagem, pelo reposicionamento e interpolação de pixels, garantindo as proporções afim de validar o encontro ou não da placa. A técnica de aproximação Douglas-Peucker (DP), capaz de selecionar cada um dos objetos presentes na imagem. Utilizando de vértices, um polígono é criado contornando o objeto real a ser considerado. No caso a placa do carro o polígono gerado deve ser analisado e tem de respeitar as seguintes características:

- O polígono deve possuir quatro vértices;
- A área do polígono deve ser compatível com a de uma placa de au-

tomóvel, a chamada proporção.

- O polígono deve ser convexo, ou seja, não deve possuir reentrâncias;
- O ângulo entre duas arestas consecutivas deve ser de 90 graus.

O polígono que respeitar as condições impostas acima possui grande chance de ser um retângulo e, consequentemente, a placa de um veículo. Todos os polígonos encontrados com tais características são marcados na imagem para que seja realizada posteriormente uma análise mais minuciosa de cada um deles.

### 3. Segmentação de Caracteres

A segmentação da imagem é um processo que possui o objetivo de separar em imagens distintas cada letra ou número de uma placa permitindo a identificação dessa placa. Para isso um algoritmo de reconhecimento de vizinhança foi feito, identificando o caractere através dos pixels pretos e brancos obtidos após a binarização. O algoritmo analisa a partir de um pixel de coordenadas (x,y) toda a sua vizinhança, ou seja analisa também os 8 pixels ao seu entorno. Desta forma o algoritmo vai verificando cada um dos pixels. Após o threshold os pixels pretos adquirem valor 0 e os brancos 1, com isso o algoritmo de contagem faz uma varredura a procura de um pixel preto. Após isso seu valor é trocado para 2 e seus

vizinhos se forem pretos trocam de valores de 3 em diante afim de marcar o espaço com pixel de valor zero. Após esse passo o algoritmo pula de 2 para o 3 e faz tudo de novo analisando seus pixels de vizinhança trocando mais valores além dos 9 possíveis anteriormente e isso é feito até acabarem os pixels negros da imagem e se recomeça uma varredura para encontrar o próximo pixel de valor zero, com isso as letras e números vão se formando como imagem e de formas separadas. Para otimizar o processo de segmentação um único algoritmo que faz a contagem e a segmentação das imagens será utilizado, e ele funciona da seguinte forma, o algoritmo original de contagem quando for terminada a identificação de um objeto para ser contabilizado, ele armazena em alguma matriz o valor que foi atribuído ao último pixel desse objeto. Com essa criação de matrizes há a separação de objetos em imagens distintas, já que cada matriz é gerada ao fim de cada contagem e armazenada de forma individual. Com isso as imagens poderão ser reconhecidas com mais uma etapa de processamento, a de reconhecimento de caracteres. Cada imagem nova criada de cada caractere separado vai ser submetida a última etapa do processamento, sendo composta apenas pelo objeto e pelo fundo que depois será transformado em dados, a imagem virará por exemplo o dado: X, derivado da placa XAB-0000.

#### **4. Filtragem de Caracteres**

Separadas imagens menores contendo as letras e números, passamos então para a parte de reconhecimento dos caracteres nessas figuras. Inicialmente havíamos pensado em selecionar os quatro pixels mais externos em cada imagem e tentar relacionar as distancias entres esses pixels e as características de cada letra e/ou número. Ao percebermos que isso não seria tão eficiente, pensamos em usar um outro algoritmo. Esse novo algoritmo agora iria percorrer a imagem comparando os pixels da

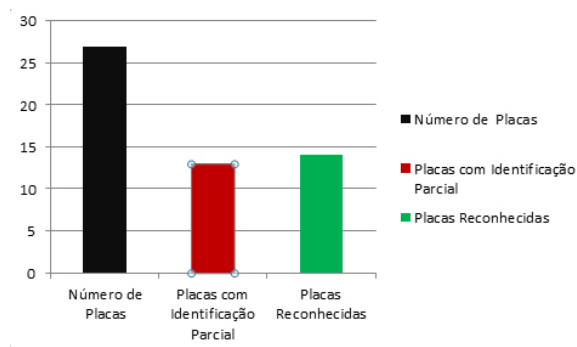
imagem a ser reconhecida com os pixels de imagens em um banco de dados de imagens, para cada pixel compatível incrementava-se um contador. No fim, dividíamos esse contador pelo total de pixels e analisávamos o percentual de compatibilidade entre essa imagem e a imagem do banco de dados, então selecionávamos o caracater com maior compatibilidade. Um problema com esse algoritmo era o tamanho diferente das imagens a serem comparadas. Para resolver esse problema percorríamos apenas a imagem a ser reconhecida pixel a pixel, os pixels do banco de dados eram percorridos a cada “largura da imagem do banco de dados/largura da imagem a ser reconhecida” e “altura da imagem do banco de dados/altura da imagem a ser reconhecida”. Esse ultimo algoritmo demonstrou certo sucesso, porém mostrou-se ineficiente por reconhecer também algumas imagens que não eram caracteres, apenas lixo captado pelo algoritmo. Enfim ficou claro que precisavamos reconhecer não somente o que era igual nas imagens, mas também o que diferia. O intuit do novo algoritmo passou a ser, reconhecer a imagem analisando as semelhanças e diferenças em cada borda da imagem. O novo algoritmo que gera quatro vetores, dois com o tamanho da largura da imagem a ser reconhecida e dois com o tamanho da altura da mesma. Em os dois vetores com o tamanho da altura são preenchidos um com a distância do primeiro pixel preto da borda esquerda da imagem e outro com a distância do último pixel preto da borda direita, para cada linha da matriz imagem. O dois vetores com o tamanho da largura são preenchidos seguindo a mesma logica, so que por sua vez com as distancia do primeiro pixel preto da borda superiores e o ultimo pixel preto da borda inferior respectivamente, para cada coluna da matriz imagem. Para as imagens do banco de dados, também são gerados os mesmos vetores e com as mesmas

dimensões. Para percorrer as linhas e colunas da imagens do banco de dados usamos a mesma lógica de incrementação usado no algoritmo anterior. Não são comparados par a par vetor base, vetor topo, vetor da esquerda e vetor da direita e novamente contamos o número de distâncias compatíveis e dividimos pelo tamanho de cada respectivo vetor. No algoritmo relacionamos os caracteres as maiores compatibilidades encontradas e depois de uma análise selecionamos um caracter como o caracter na imagem em reconhecimento. A complexidade desde ultimo algoritmo se encontra no fato de que alguns caracteres demonstram extrema semelhança, casos como “B” e “D”, “R” e “P”, “U” e “V”, “Q” e “O”, “C” e “G”, “N” e “H”, “E” e “F”, “6” e “8”, “9” e “8”, “9” e “0”, “0” e “8”, “0” e “6” entre outros. Para resolver isso procuramos identificar como o caracter em si, as imagens com maior compatibilidade possível em todas as bordas, e no maior número de bordas possível. Outro problema é a semelhança entre letras e números, em casos como “6” e “G”, “0” e “G”, “0” e

“B”, “8” e “B”, um e “I” e “1” e etc. A solução mais simples foi separar a detecção de letras da detecção de números, e começar a detecção dos números apenas depois de identificados as três letras das placas.

## 5. Conclusão

Foi possível perceber durante o desenvolvimento e testes do trabalho que o mais importante no reconhecimento é um rico e variado banco de imagens dos caracteres, pois quanto maior o banco maior é a eficiência na detecção. Notamos que para as imagens que tínhamos menos caracteres no banco de dados o sucesso na detecção era visivelmente menor. Letras como Y e M trouxeram dificuldades na detecção por conta do desgaste que a maioria das placas do nosso banco possuíam. Um descuido dos motoristas hoje em dia é com a sinalização de seu carro, facilmente vimos placas sem cor, placas descascando, placas entortadas entre outros. A “Figure 2” mostra o número de placas utilizadas e reconhecidas pelo software.



**Figure 2. Número de placas usadas, parcialmente e totalmente reconhecidas**

Uma grande dificuldade encontrada foi quanto a qualidade das imagens dos veículos e determinar o ponto de corte ideal do Threshold para selecionar os caracteres na placa, visto que em uma situação real o carro estará em movimento a imagem será feita de forma rápida e eficiente

pegando toda a área da placa. O sistema deve ser inteligente, capaz de detectar essa placa na imagem, de nada adianta reconhecer os caracteres se na fotografia também houver placas na rua e outros sinais que possam confundir o sistema. Por fim concluímos que é essencial para a sociedade

que o processo de reconhecimento de placas seja eficiente e rápido, diminuindo os erros humanos e acelerando todo o processo de fiscalização do tráfego e do trânsito nas nossas cidades. Além disso é importante manter um banco de imagens muito grande e com uma grande variedade de imagens. Outro ponto importante é garantir que serão tiradas imagens com um determinado padrão e qualidade para garantir a eficiência de todo o processo. Concluindo assim percebemos a importância de nosso trabalho e de nossa pesquisa no assunto.

## 6. References

[1] DAMBROS, A. L., “Sistema de Reconhecimento de Placas de Veículos Automotores”. Centro Universitário Feevale, Novo Hamburgo, 2008.

[2] BRADSKI, G.; KAEHLER, A., “Learning OpenCV: Computer Vision with OpenCV Library”. Sebastopol, CA, O’Reilly Media, Inc., September 2008. 577p.

[3] NUÑES, J. R., “Segmentação Automática e Classificação em Vídeos Esportivos”. Dissertação (Mestrado) - Pontifícia Universidade Católica do Paraná, Curitiba, 2006.

[4] “Principais Interesses de Pesquisa: Processamento de Imagens e Visão Computacional” - Unisinos

[5] MARANA, A. N.; BREGA, J. R. F., “Técnicas e Ferramentas de Processamento de Imagens Digitais e Aplicações em Realidade Virtual e Misturada”, Bauru, Canal6 Editora, 2008.

[6] Duda, R. O; Hart, P. E. “Pattern Classification and Scene Analysis”, John Wiley Sons, 1973.

[7] Kupac, G. V.; Rodrigues, R. J.; Thomé, A. C. G.; “Extração de características para Reconhecimento de Dígitos Cursivos”, VIth Brazilian Symposium on Neural Networks - SBRN2000, Rio de Janeiro, november 2000.

[8] Rodrigues, R. J.; Kupac, G. V.; Thomé, A. C. G.; “Character Feature Extraction Using Polygonal Projection Sweep (Contour Detection)”, accepted for presentation at IWANN2001. Granada - Spain, June 2001.

[9] Rodrigues, R. J.; Silva, E.; Thomé, A. C. G.; “Feature Extraction Using Contour Projection”, accepted for presentation at SCI2001. Orlando - USA, July 2001.