

## RECONHECIMENTO DE PLACAS DE AUTOMÓVEIS ATRAVÉS DE CÂMERAS IP

Caio Augusto de Queiroz Souza  
caioaugusto@msn.com  
Éric Fleming Bonilha  
eric@digifort.com.br  
Gilson Torres Dias  
gilson@maempec.com.br  
Luciano Gerizani  
walge@uol.com.br  
Oswaldo Ortiz Fernandes Jr  
oswaldo@imes.edu.br  
Pedro Vilaça  
pvilaca@terra.com.br

Universidade IMES-São Caetano do Sul (SP)

### Abstract

The goal of this work is the optimization of a algorithm for real time License Car Plate recognition using as base the existing tonal variation algorithm

### Resumo

O objetivo deste trabalho é a otimização de um algoritmo para a localização de placas automotivas em tempo real baseando-se no algoritmo já existente de análise de variação tonal.

**Palavras Chaves:** Placas Automotivas, Localização de Placas, Análise de Variação Tonal

### MOTIVAÇÃO

A motivação para este trabalho é otimizar um algoritmo de localização de placas de automóveis visando à melhoria do desempenho do algoritmo já existente de localização de placas chamado Análise de Variação Tonal. O algoritmo existente de Análise de Variação Tonal não apresenta um desempenho satisfatório para aplicações em tempo real, assim, o novo algoritmo deve ser capaz de analisar as imagens e localizar a placa do automóvel em velocidade apropriada para a sua utilização em sistemas de tempo real.

### 1 ALGORITMO DE ANÁLISE DE VARIAÇÃO TONAL EXISTENTE

Para este trabalho, foram pesquisados diversos métodos de localização de placas através de Análise de Variação Tonal e foi utilizado como base, os algoritmos descritos nos trabalhos [TESEFER-2000], [FERMARTIN-99], [FATHY-2000] e [SOHY-94]

O Algoritmo de análise por variação tonal consiste em realizar uma varredura seqüencial na imagem para localizar a “impressão digital” da placa. Dizemos impressão digital, pois a placa de um automóvel é formada de dígitos em cima de um fundo (Geralmente Dígitos Pretos em cima de Fundo Cinza) gerando uma grande diferença tonal nas bordas dos dígitos [FATHY-2000], como mostra a figura 1, veja a diferença da linha A e da linha B. Na linha B onde o algoritmo analisou a área da placa é possível perceber o padrão que se forma no meio da imagem, o algoritmo de Análise por Variação Tonal tem o objetivo de identificar estas áreas e classificá-las como uma possível área de localização da placa.

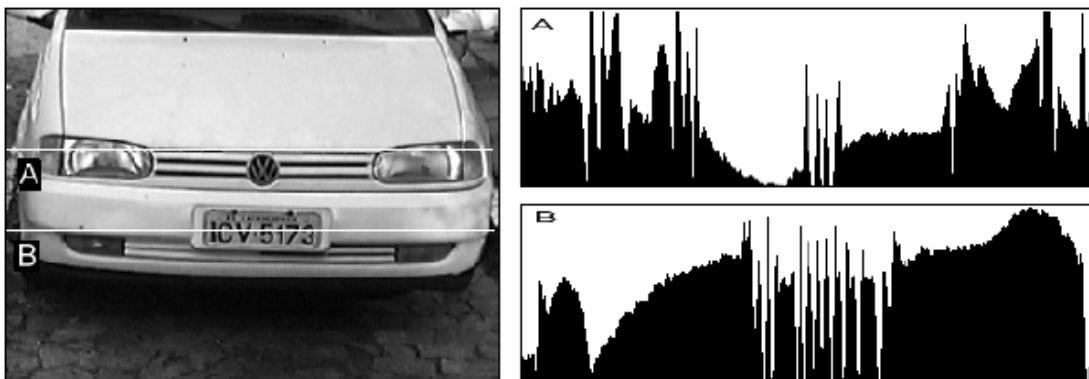


Figura 1 – Região da “Impressão Digital” da placa [TESEFER-2000]

Nos trabalhos pesquisados para se realizar esta análise os seguintes passos eram necessários [TESEFER-2000]:

1º O primeiro passo é a utilização do filtro de localização de bordas Sobel sobre cada linha amostrada da imagem, este filtro tem por objetivo realçar as bordas dos objetos na imagem, facilitando a sua localização. A figura 2 ilustra a saída do filtro de bordas Sobel:



Figura 2 – Saída do Filtro Sobel [TESEFER-2000]

O Filtro Sobel é um filtro de Convolução que utiliza uma janela de 5x5 pixels para analisar a linha amostrada, sendo assim, para cada pixel analisado, uma vizinhança de tamanho 5x5 também é analisada, o que demanda processamento (Diminuindo a performance geral do algoritmo).

2º O segundo passo é a análise das linhas amostradas onde todos os picos encontrados, independente de sua amplitude, são marcados em x1 e x2, indicando seu início e fim respectivamente. Aqueles picos cuja intensidade da imagem original nas marcas x1 e x2 não forem próximas, e ainda, cuja distância entre x1 e x2 for menor, ou maior, que a largura esperada de um segmento vertical de caractere são descartados. [TESEFER-2000]

3º O próximo passo é a criação de intervalos. Neste ponto as marcas x1 são descartadas e são criados grupos de intervalos entre as marcas x2 remanescentes. Uma análise dos grupos criados permite selecionar aquele que está dentro das dimensões horizontais 66 esperadas da placa, e que possui um número de intervalos, de tamanho pré-definido, entre um valor mínimo e máximo aceitáveis. [TESEFER-2000]

4º Uma vez encontrada uma linha que atenda aos quesitos da etapa anterior, uma nova análise é feita nas linhas adjacentes a ela. Quando um conjunto de linhas adjacentes maior que um tamanho pré-determinado, normalmente igual ao tamanho esperado do caractere, for encontrado, o algoritmo é terminado. Devido a algumas imperfeições na implementação deste algoritmo que causam uma imprecisão na localização exata do local da placa, e além disso, devido à grande variação dos tamanhos das placas contidas no banco de testes (entre 70x20 e 120x40 pixels), o local candidato a placa é inscrito em um retângulo de dimensões fixas, 120x50 pixels, aumentando a chance da placa estar contida nesta região. [TESEFER-2000]

Esta foi uma descrição sucinta do algoritmo existente estudado nos trabalhos em referência, assim após analisarmos profundamente este algoritmo encontramos pontos que poderiam ser melhorados. O principal ponto que encontramos foi a utilização do Filtro Sobel. Este, por ser um filtro de convolução que utiliza um kernel (Janela) de 5x5 para cada pixel analisado, demanda muito tempo de processamento, o que inviabiliza o seu uso em algoritmos de tempo real, sendo assim, o principal objetivo do novo algoritmo é não utilizar o Filtro Sobel, economizando processamento. Para o desenvolvimento do novo algoritmo impomos os seguintes objetivos:

- Não utilizar o Filtro Sobel
- Analisar toda a imagem em apenas 1 passada
- Não consumir mais de 5ms para analisar uma imagem de 352x240 em um computador de 2.4GHz de Clock.

Baseando-se nestes objetivos, desenvolvemos um novo algoritmo que supera em 96,77% o tempo de processamento do algoritmo existente.

## 2 NOVO ALGORITMO DE ANÁLISE DE VARIAÇÃO TONAL

O algoritmo de análise de variação tonal desenvolvido é dividido nas seguintes fases:

### Fase 1: Amostra das linhas

Esta fase consiste em amostrar as linhas da imagem pois não é necessário analisar todas as linhas de uma imagem, assim, apenas um subconjunto de linhas serão analisadas.

O espaçamento entre as linhas selecionadas dá-se pelo seguinte cálculo:

Altura da Placa / 5

Assim uma placa com altura estimada de 40 pixels irá ter um espaçamento na amostra de  $40/5 = 8$  Linhas.

### Fase 2: Algoritmo de seleção de variação da linha amostrada

Para cada linha amostrada na fase anterior esta fase deve fornecer como saída um vetor contendo apenas os pontos com maior variação tonal. A figura 4 ilustra a entrada desta fase.



Figura 4 - Exemplo de entrada (Linha selecionada (Vetor)) e a variação da linha. O objetivo deste algoritmo é analisar o gráfico gerado pelo vetor de Byte (Exemplificado na Figura 4) e limpar o vetor para que ele contenha apenas as áreas que contenham maior variação tonal, como exemplificado na figura 5:

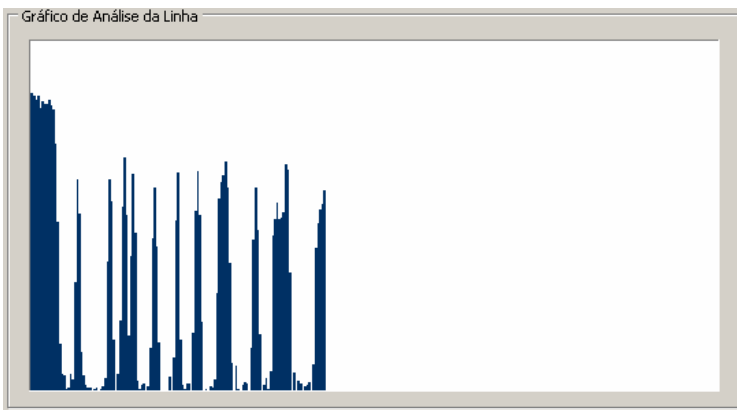


Figura 5 - Exemplo de saída do algoritmo

O algoritmo deve executar os seguintes passos para alcançar o seu objetivo:  
Primeiramente um valor deve ser usado como base para analisar se a variação é significativa ou não, de acordo com os testes realizados, o valor de 35 reconheceu a grande maioria das placas.

Tendo o valor mínimo de variação, o algoritmo escaneia o vetor da esquerda para a direita, realizando os seguintes cálculos:

1º Marca-se o primeiro ponto do vetor (Ponto 0) como o ponto de referência

2º Escaneia-se o vetor da esquerda para a direita até encontrar uma mudança no sentido do gráfico.

Mudança neste caso é quando o gráfico está subindo e depois começa a descer, ou vice versa, como mostra a Figura 6.



Figura 6 - Mudança no sentido do gráfico

3º Quando o algoritmo encontra uma mudança no sentido do gráfico, ele verifica se a diferença do valor do ponto que gerou a mudança do sentido para o ponto de referência é maior do que o valor do limiar definido (35 neste caso) se for menor, ele ignora este ponto e continua analisando o vetor, se for maior ele verifica se a distância entre o ponto de referência e o ponto que gerou a mudança no sentido do gráfico é maior do que o tamanho esperado de um segmento da placa (O tamanho esperado é o tamanho horizontal da placa / 7), se for maior, ele descarta todos os pontos entre o ponto de referência e o ponto que gerou a mudança do sentido, caso contrário ele mantém o vetor como está e após isso, define o ponto de referência como o ponto que gerou a mudança de sentido.

O Algoritmo executa estes passos escaneando o vetor por completo, no final do seu processamento o que resta no vetor são apenas os locais onde o algoritmo encontra uma variação característica dos caracteres da placa.

O próximo passo é analisar o vetor resultante para eliminar possíveis ruídos, deixando apenas o local que mais se assemelha à variação tonal de uma placa.

Fase 3: Algoritmo de segmentação

Este algoritmo recebe como entrada o vetor resultante do algoritmo de análise de variação tonal e tem como objetivo filtrar os picos que não correspondem às características de uma placa automotiva, ou seja, picos isolados na amostra, pois apenas os picos que estejam próximos uns aos outros são características da variação tonal de uma placa.

Primeiramente ele irá dividir os valores do gráfico em segmentos, cada segmento irá conter os valores dos picos que estão entre valores nulos, alguns exemplos de segmentos são dados abaixo:

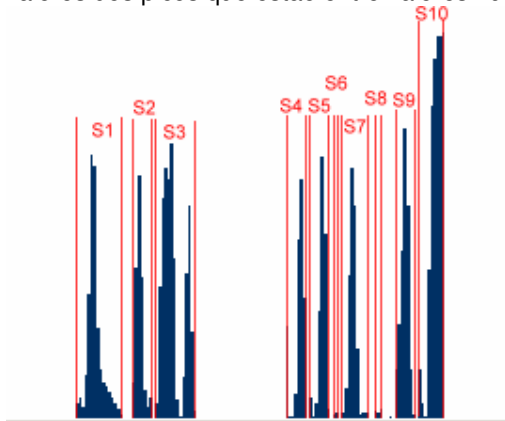


Figura 7 - Exemplo de segmentos de uma placa com alto contraste

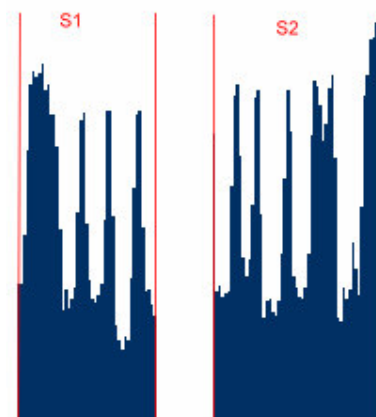


Figura 8 - Exemplo de segmentos de placa de baixo contraste

As Figuras 7 e 8 demonstram que uma placa pode conter de 1 a vários segmentos.

O tamanho máximo que um segmento pode ter é:

Tamanho Horizontal da placa \* 1.5, ou seja, o tamanho horizontal da placa acrescido de 50%.

Após a divisão do gráfico em segmentos, eles devem ser agrupados em grupos que serão analisados quanto ao tamanho da placa.

Para agrupar os segmentos, eles são analisados quanto à distância entre eles, sendo que a distância entre os segmentos não deve ser maior do que:  $(\text{Tamanho da placa} / 7) * 2$ .

Um exemplo de grupos é dado abaixo:

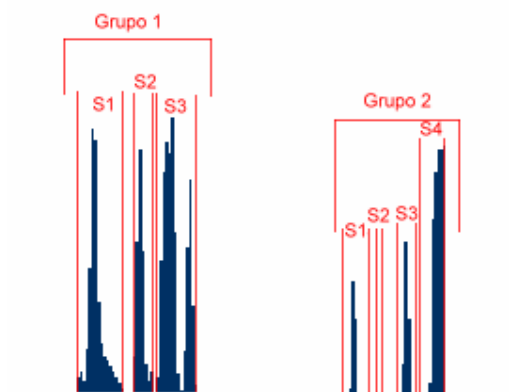


Figura 9 - Exemplo 1 de classificação dos segmentos em grupos

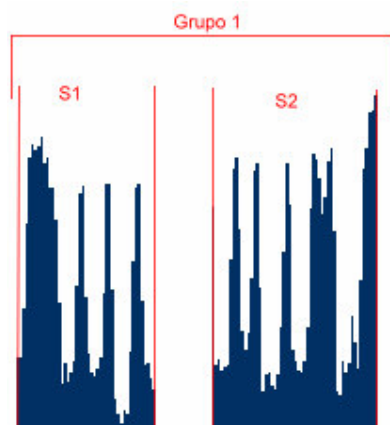


Figura 10 - Exemplo 2 de classificação dos segmentos em grupos

Após a separação em grupos, cada grupo é analisado para verificar se ele está dentro dos limites horizontais de uma placa com 50% de tolerância. A fórmula abaixo exemplifica este cálculo:

$$\text{TAMANHOPLACA} - 50\% \leq \text{TAMANHOGRUPO} \leq \text{TAMANHOPLACA} + 50\%$$

Aplicando este cálculo sobre os dois exemplos acima, no exemplo 1 ambos os grupos serão descartados e no exemplo 2, o grupo será mantido.

Fase 4: Análise das linhas adjacentes

Quando uma linha amostrada da imagem é passada para o algoritmo de análise de variação tonal e filtro de grupos, o retorno são os grupos encontrados na linha, se o total de grupos for igual ou maior que 1, então uma análise nas linhas adjacentes se faz necessária uma vez que estas linhas não foram analisadas devido à amostragem de linhas na Fase 1.

As fases 2 e 3 são repetidas para as linhas adjacentes à linha analisada.

Após a análise de todas as linhas, as linhas que contiverem a placa do automóvel irão retornar grupos, que irão iniciar na primeira letra da placa e finalizar no último número da mesma, retornando assim sua localização na imagem.



Figura 11 - Exemplo da análise das linhas. Resultado do algoritmo de localização de placas

As saídas do algoritmo de localização da placa são uma ou mais regiões prováveis de localização. Na figura 11 apenas uma região é encontrada, mas o algoritmo poderá eventualmente indicar outras regiões como candidatas, estas são regiões ruidosas e uma análise de cada região se faz necessária na fase de segmentação dos caracteres.



Figura 12 - Exemplo de localização com Ruídos

No exemplo da Figura 12, o algoritmo processou uma imagem de 640x480 em 5,4ms retornando 3 áreas candidatas à localização da placa. Estas áreas devem ser processadas posteriormente na fase de segmentação dos caracteres.

### 3 CONCLUSÕES

Comparações feitas com o trabalho de [TESEFER-2000] mostram a eficácia do algoritmo de localização principalmente no que se refere a desempenho. Foram testadas 300 imagens, onde o algoritmo obteve um resultado positivo em 290 delas, obtendo um valor de 96% eficiência.

Nos testes realizados o desempenho do algoritmo (Analisando tempo de execução) é dado na tabela abaixo:

	320x240	640x240	640x480
Algoritmo Desenvolvido	2 ms	3 ms	6 ms
SIAV 1.0 [TESEFER-2000]	62 ms	X	X

Para este teste não foi possível testar as imagens de resolução 640x240 e 640x480 com a versão 1.0 do SIAV [TESEFER-2000], pois o mesmo não localizou as placas, mas se extrapolarmos o resultado obtido com a imagem de 320x240 do SIAV é possível ver que seu desempenho não está nem próximo ao desempenho do algoritmo desenvolvido neste trabalho.

Esta diferença de tempo de processamento ocorre devido ao SIAV utilizar o filtro Sobel de realce de bordas, o qual demanda um tempo significativo de processamento, enquanto que o algoritmo desenvolvido neste trabalho não utiliza este filtro.

#### **4 REFERÊNCIAS BIBLIOGRÁFICAS**

- [FERMARTIN-99] Martín, Fernando – Automatic Reading of Vehicle License Numbers  
[FATHY-2000] M.Fathy - Application of pattern recognition for Farsi license plate recognition  
[SOHY-94] SOH Y. S., CHUN B. T., YOON H. S., Design of Real Time Vehicle Identification System,  
[TESEFER-2000] Peixoto, Fernando – Localização e Leitura Automática de Caracteres Alfanuméricos – Uma aplicação na identificação de Veículos.