UNIVERSIDADE ESTADUAL DE CAMPINAS

Instituto de Computação.

Análise de Imagem Orientada a um Problema do Mundo Real

Título:

Detecção da placa do veículo.

Alunos:

Darwin Danilo Saire Pilco
Jhosimar George Arias Figueroa
Juan Felipe Hernández Albarracín
Marcos Piaia
Ricardo Nishihara

Novembro 2015

Detecção da placa do veículo

Com o grande crescimento da frota de veículos nos últimos anos, surgem problemas em que a engenharia de tráfego tem que lidar de forma a conseguir informações rápidas e precisas destes veículos, não apenas para controle e monitoração, mas também para solucionar questões de segurança e planejamento.

É por isso que se precisa de um sistema automático de reconhecimento da licença de um veículo (ALPR) o qual tem quatro módulos principais: (1) Pré-processamento de imagem, (2) Detecção / localização da placa, (3) Segmentação de caracteres e (4) Reconhecimento de caracteres.

Entre eles, a tarefa da "Detecção da placa" é considerada como o estágio mais crucial no sistema ALPR, já que apresenta mais problemas para a localização por o efeito da luminância, resolução, ruído, e outros. Além disso, se esta tarefa não é realizada corretamente, limitará as outras tarefas, afetando diretamente os resultados.

1. Seleção de candidatos

1.1. Morfologia Matemática (MM)

O método baseado em MM apresenta a vantagem de depender pouco das condições de iluminação, dessa forma se adequando as mais variadas aplicações. Tendo como característica principal, reduzir significativamente o número de candidatos extraídos aumentando assim a velocidade subsequente do reconhecimento da placa.

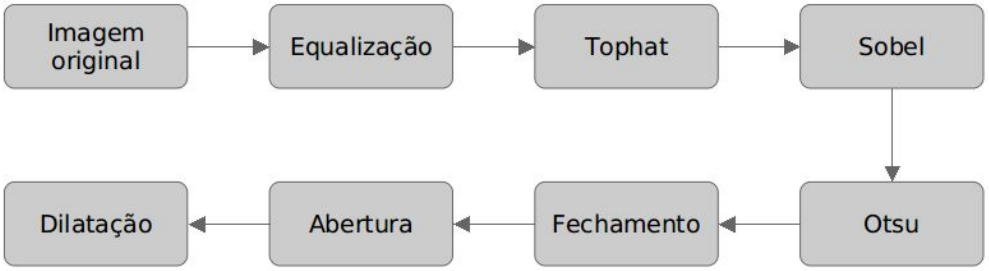


Figura 1: Etapas de morfologia matemática, para a seleção dos candidatos.

Equalização: Utilizamos a equalização de histogramas para melhorar o contraste da imagem.

Top-hat: Top-hat por Fechamento é composto de um fechamento com um elemento estruturante retangular com largura da espessura dos caracteres, assim consegue-se borrar as letras. Em seguida subtraem-se as imagens, então os caracteres serão destacados.

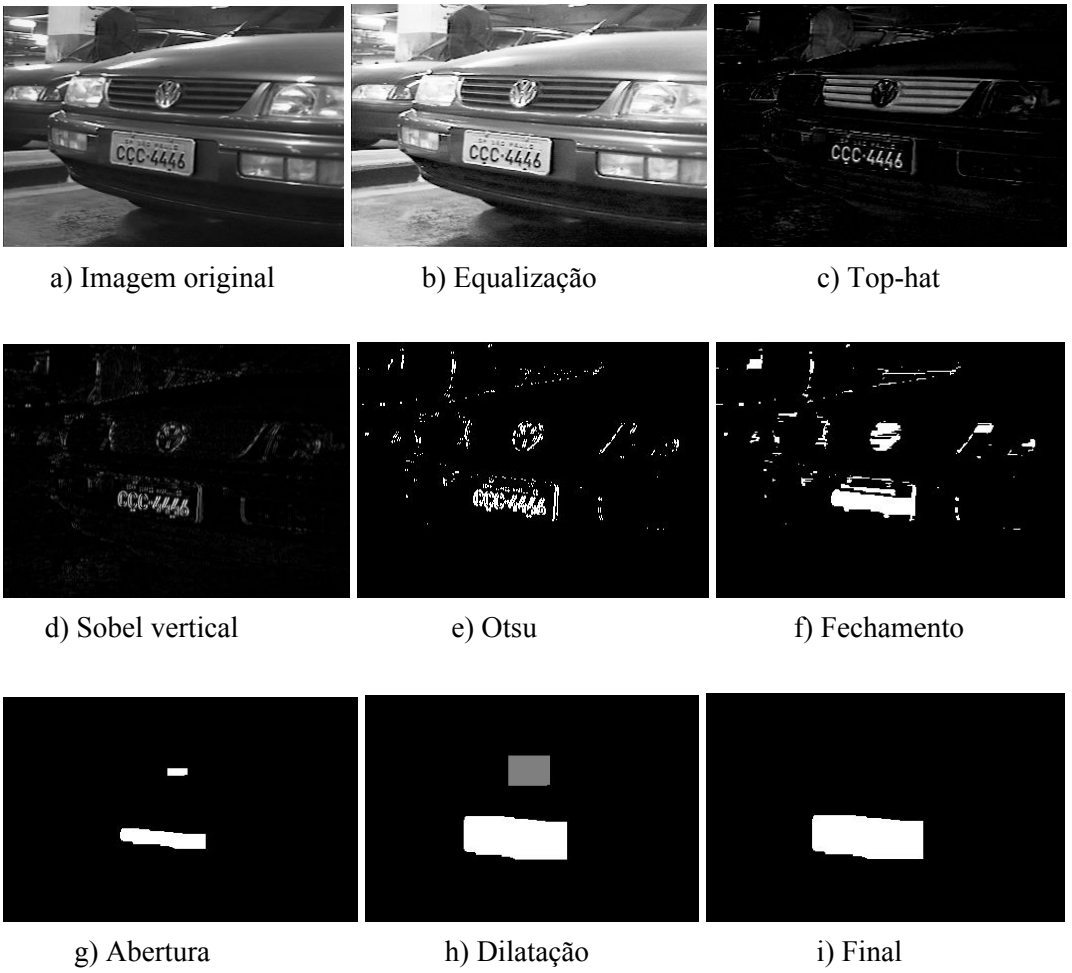
Sobel: É utilizado para trabalhar com o gradiente na vertical e assim, poder obter o gradiente entre os caracteres alfanuméricos e a placa, como os dois componentes, geralmente apresentam diferentes tons, o gradiente nesta zona vá ser forte, ressaltando assim, os caracteres do fundo da placa.

Binarização: Para esta binarização utilizaremos o abordagem de otsu para selecionar o melhor limiar.

Fechamento horizontal: Realiza-se o fechamento com um elemento estrutural horizontal linear, com a largura igual à maior separação entre os caracteres possíveis. Isto converte o conjunto dos caracteres em um retângulo branco.

Abertura vertical: Agora buscaremos eliminar os ruídos existentes que não satisfazem as características da placa. Fazendo uma abertura com um elemento estrutural vertical linear eliminamos os objetos de altura inferior à altura mínima dos caracteres

Dilatação: agora dilatamos os objetos resultantes para pegar os bordes da placa, não só os caracteres, alem disso, eliminamos os componentes com uma areá minima.



a) Imagem original b) Equalização c) Top-hat

d) Sobel vertical e) Otsu f) Fechamento

g) Abertura h) Dilatação i) Final

Figura 2: Pré-processamento para a seleção dos candidatos.

Podemos notar na Figura 3, os componentes antes de ser dilatados são aproximadamente da altura e largura dos caracteres da placa e podem ser utilizado como sementes do objeto para a seguinte tarefa, além disso, o as bordas do componente dilatado pode também ser utilizado como sementes do fundo e assim poder fazer a tarefa de delimitação.

a)Imagem resultante b) Possíveis sementes.

Figura 3: Preparação dos sementes para a etapa de delimitação.

A medição dos resultados para a seleção dos candidatos e feita com métrica de sensibilidade, a qual diz de todos os candidatos quantos deles estão certos em relação ao conjunto de treinamento.
Geramos 1403 candidatos no total, dos 990 placas no label, se obteve 983 dando uma sensibilidade de 99.2%
Os falsos positivos foi elevado, geramos 418 falsos positivos, isto é 29.7 %, mas este porcentagem vai ser diminuído com o uso dos classificadores.

1.2. Transformada discreta de Fourier (DFT)

Essa abordagem foi utilizada em [1] e método em questão é baseado no uso da DFT considerando que a existência da placa em uma determinada região da imagem irá impactar um conjunto específico de componentes dessa transformação de forma que, ao se analisar o impacto nas linhas e nas colunas de forma independente será possível inferir a localização da placa.

Esse impacto mencionado acima é calculado com base na "energia" dessas componentes de forma que a informação mais relevante nesse contexto é o seu módulo.

Com isso, dado que este é uma informação real, decidiu-se alterar um pouco a experiência realizada em **[1]** e no lugar da DFT usar a DCT (Transformada Discreta do Cosseno), visto que esta já trabalha apenas com as componentes reais, de forma que o cálculo da "energia" total dos componentes impactados pela existência da placa torna-se apenas uma soma direta desses elementos.

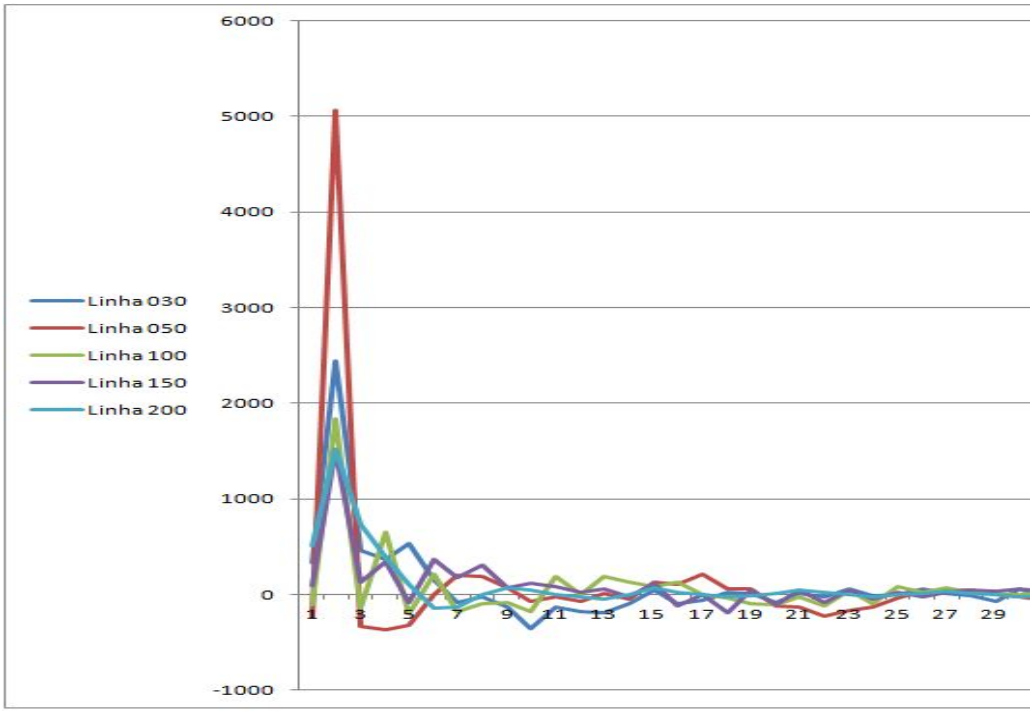
Para realizar a análise da imagem e o impacto da placa na DCT realizou-se o cálculo dessa primeiro linha a linha e depois coluna a coluna.

Segue abaixo o estudo de um dos exemplos usados na implementação em questão.

Imagem Original

Usando essa imagem, gerou-se os dados da DCT de linha para o estudo das componentes afetadas pela existência na placa, de forma a ser possível verificar se, efetivamente, tal correlação existia.

Segue abaixo um gráfico com os dados dessa DCT onde isolou-se, apara análise, linhas que onde havia intersecção com a placa (100, 150) e linhas onde isso não acontecia (30, 50 e 200):



Com base nos picos identificados utilizou-se os componentes 4,5 e 6 para a sua análise e calculando-se a “energia” dos mesmos foi possível identificar os seguintes pontos:

Imagem Original com a DCT de linha

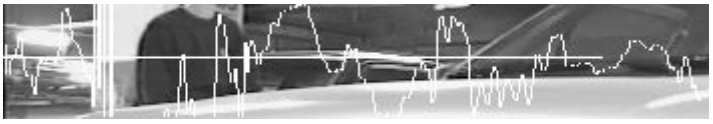
Com isso verificou-se que efetivamente havia um impacto decisivo nessas componentes quando a placa estava presente, restando apenas uma análise complementar para remover os picos indesejados, onde a placa não estava presente.

Para isso, uma abordagem complementar a [1] foi usada e decidiu-se por gerar vários candidatos na imagem analisada e submeter cada um deles a uma análise da DCT de coluna.

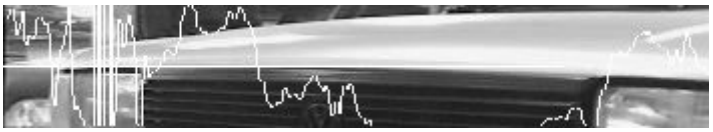
Esse procedimento gerou o seguinte conjunto de informações:

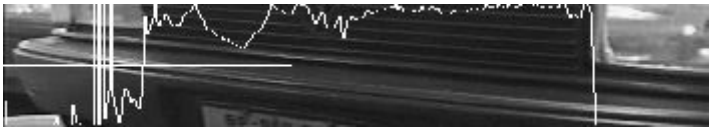
Realizou-se então o mesmo processo de análise de componentes realizada para a integral de linha e encontrou-se os seguintes coeficientes relevantes para a DCT de coluna:
2, 3, 25, 26, 27

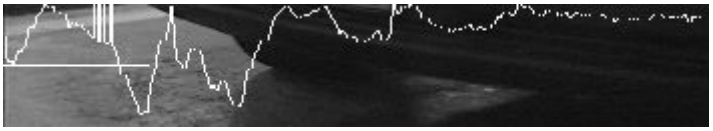
O resultado pode ser visto abaixo, onde também colocou-se uma projeção do componente da DCT de linha que gerou aquele candidato

(1) 

(2) 

(3) 

(4) 

(5) 

Nota-se então que, considerando a distância dos elementos da DCT de coluna em relação à essa projeção da DCT de linha como a "energia" da DCT de coluna, vemos que a imagem (2) apresenta a maior "energia", contendo então a placa.

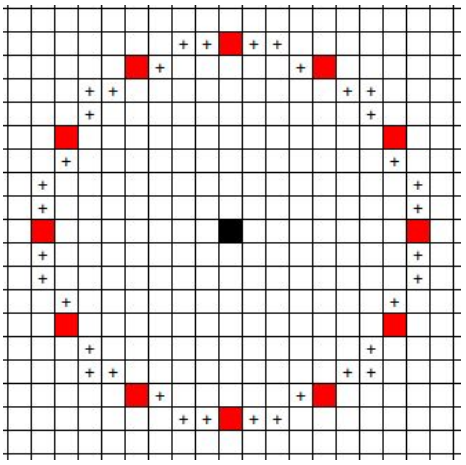
Ainda na imagem (2) podemos observar que a placa começa no primeiro elemento desse maior grupo positivo, de forma que podemos construir um retângulo ali, identificando a placa.

2. Extração de características

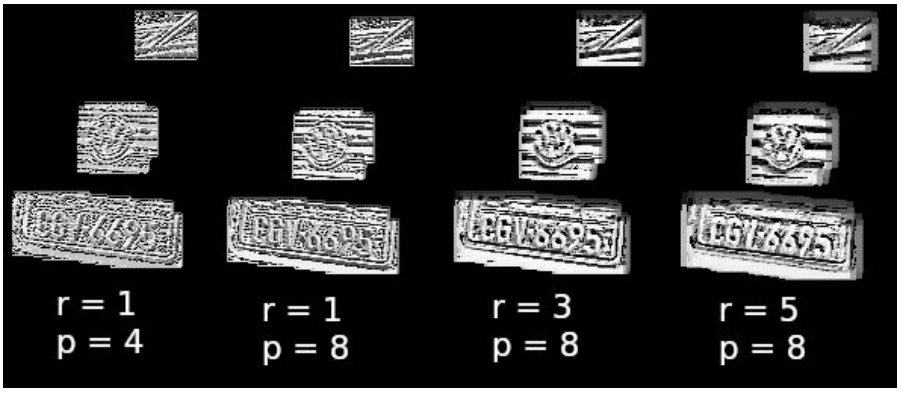
2.1. Local Binary Pattern (LBP)

Padrão binário local (LBP pelo seu acrônimo em inglês) é um descritor de textura simples e rápido, bastante popular para diversas aplicações. Em termos gerais, este descritor consiste na frequência de cada um dos padrões binários em um conjunto já definido para cada pixel da imagem. Dado um pixel, uma vizinhança circular é definida; no caso padrão, essa vizinhança são os 8 pixeis adjacentes. O padrão binário desse pixel nesse caso teria 8 bits (um por cada pixel adjacente) onde 0 significa que a intensidade do pixel central é maior que a intensidade do pixel adjacente correspondente, e 1 é o caso contrário. O descritor final é o histograma dos 2^8 = 256 padrões binários.

Uma extensão do LBP para vizinhanças circulares de diferentes rádios foi implementada. O algoritmo recebe o rádio (r) e o tamanho da amostra (p), que indica o número de pixeis que serão considerados no padrão binário. O tamanho final do descritor será 2^p. A seguinte figura mostra um exemplo de vizinhança com r = 7,8 e p = 12.



As características foram extraídas a partir da imagem equalizada, e os vectores resultantes foram normalizados com a norma L2.



A figura mostra a imagem gerada pelos padrões binários de três candidatos, dos quais um deles é placa. Dado que o vetor de características é um histograma, as dimensões dos candidatos não representaram problema nenhum.

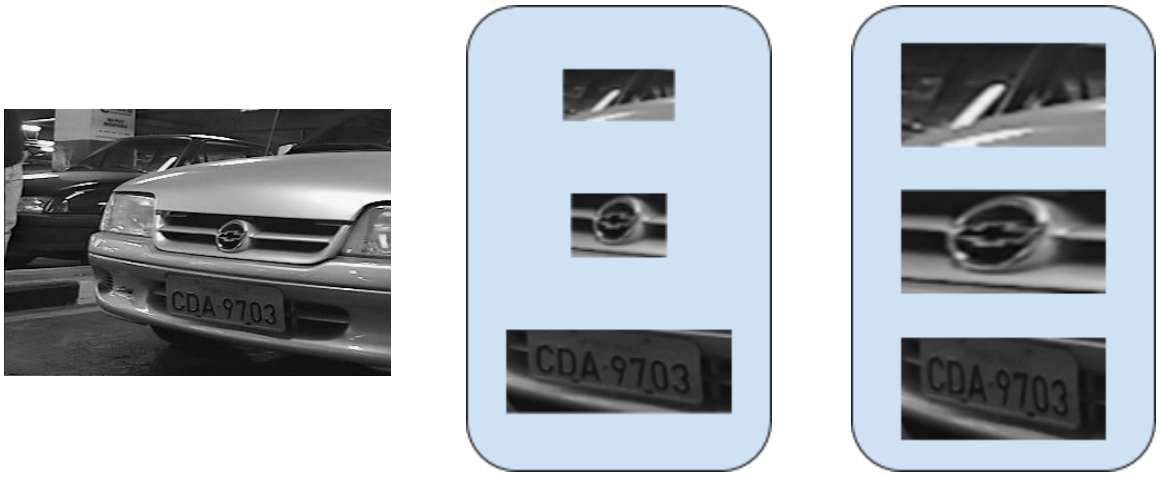
Os experimentos foram feitos variando os valores de *r* e *p*. O melhor desempenho foi conseguido com um rádio de 1,0 e uma amostragem de 8. Maiores valores de *p* deram uma maior dimensionalidade e não representaram nenhuma melhora.

2.2. Histogram of Oriented Gradients (HOG)

Histograma de gradientes orientados (HOG) é uma excelente escolha para a detecção da placas devido a sua capacidade de detecção de objetos. O HOG foi usado em cada candidato detetado.

Um problema no momento da aplicação do descritor foi que o descritor não é invariante a escala, por tanto não é possível aplicar o descritor diretamente sobre os possíveis candidatos devido a que o bounding box de cada candidato pode ter diferentes dimensões. A solução considerada para esse problema foi realizar um resize do candidato com uma janela de tamanho fixo de 128x64 fazendo uso de uma interpolação bilinear 2D. Desse jeito todos os

candidatos teriam as mesmas dimensões para poder aplicar o descritor. Possíveis problemas com esta abordagem seriam a introdução de ruido no candidato, caso o candidato aumente de tamanho, ou perda de informação, caso o candidato diminuía de tamanho. Porém, na maior parte dos casos, os candidatos incrementam de tamanho por tanto não há perda de informação. A seguinte figura mostra a imagem original, os candidatos obtidos e o resultado do resize aplicado sobre cada candidato.



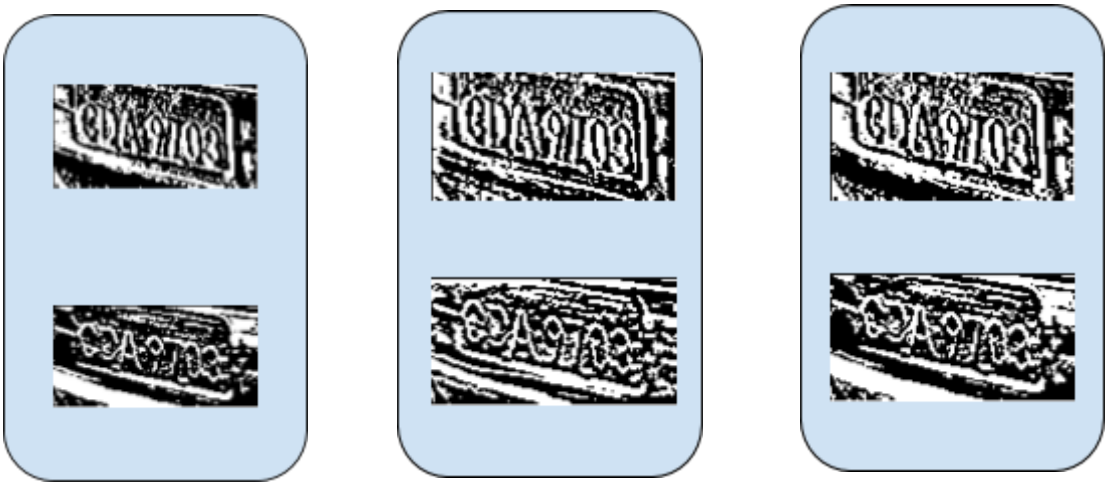
(a) Imagem original (b) Candidatos (c) Resize 128x64

Normalização da imagem: O primeiro passo para o cálculo do descritor foi normalizar a imagem de cada candidato para reduzir efeitos da iluminação. Foram testadas três abordagens de normalização: A normalização padrão usada pelos autores originais considerando L=255, a normalização proposta com adjacência circular de radio igual a 5 e a equalização de histograma sobre a imagem. A seguinte figura mostra os resultados apos cada normalização.

(a) padrão **(b) proposta** **(c) equalização**

Figura: Normalização dos candidatos

Vetor Gradiente e sua orientação: O segundo passo foi o cálculo do vetor gradiente, usou-se o cálculo do gradiente padrão sobre cada uma das normalizações.



(a) padrão **(b) proposta** **(c) equalização**

Figura: Gradientes dos candidatos após normalização

Histograma de cada célula: A imagem foi dividida em 8x8 células e foi considerado 9 bins no histograma de cada célula. A distribuição dos pesos foi feita por meio de interpolação trilinear com as células vizinhas de cada célula. O resultado é uma matriz de células, onde cada entrada da matriz contém um histograma de 9 bins previamente calculados.

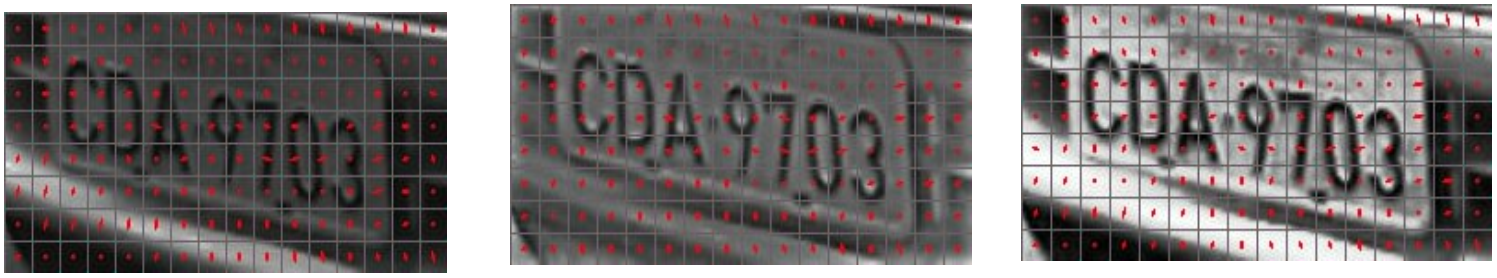
Descritor de bloco e sua normalização: Finalmente a matriz de células foi dividida em blocos de 2x2. Isto é 16x16 pixels da imagem original e foi considerado sobreposição de 1x1 células entre blocos no eixo horizontal e vertical. A normalização usada na concatenação dos histogramas foi a normalização L2. O tamanho final do vetor de características é dado pelo seguinte:

#Características = #Blocos * #CélulasPorBloco * #BinsPorCélula

Considerando os parâmetros usados neste trabalho, o número de características foi:

$$\#Características = (128/8 - 1) * (64/8 - 1) * (2 * 2) * 9 = 3780$$

A seguinte figura mostra a visualização dos gradientes sobre os candidatos:



(a) padrão **(b) proposta** **(c) equalização**

Figura: Visualização de HOG

A implementação do descritor é flexível, isto quer dizer que a implementação não esta baseada só em parâmetros fixos, é possível especificar os parâmetros desejados como o número de células, número de blocos, número de bins, rango dos ângulos dos gradientes(180 ou 360) e o número de células de sobreposição.

2.3. Scale-Invariant Feature Transform (SIFT)

SIFT é um algoritmo relativamente complexo proposto por David Lowe para detecção de características robustas a variação de escala, sendo composto por um detector de pontos de interesse e de um descritor calculado em uma vizinhança dos pontos de interesse selecionados. Pode-se dividi-lo em 5 etapas principais:

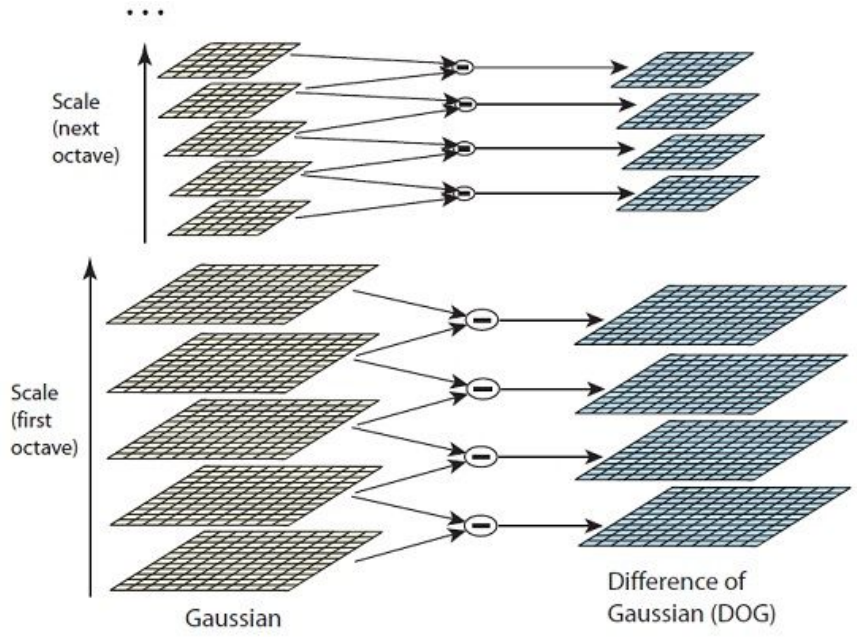
1. Obtenção do espaço de escala, em que objetivo é obter representações da imagem original que assegurarem invariância a variação de escala.
2. Detecção de pontos de extremos no espaço de escala
3. Eliminação dos pontos candidatos de baixo contraste e de pontos que estão sobre bordas
4. Associação de uma orientação aos pontos de interesse.
5. Geração do descriptor SIFT.

2.3.1 Espaço de escala

Conforme ilustrado no esquema da figura e nas equações a seguir, o espaço de escala das features SIFT é gerado através de filtragens gaussianas sucessivas aplicadas sobre a imagem original *I(x,y)*. Cada uma das imagens filtradas resultantes, *L(x,y,k*sigma)*, define uma escala (nível de borramento). Após uma sequência de filtragens, em que dobramos o nível de borramento (escala), completando uma oitava de escalas, fazemos o resize diminuindo a imagem para metade do tamanho, e realizamos uma nova sequência de filtragens, repetindo novamente o mesmo processo, até completarmos o número desejado de oitavas.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

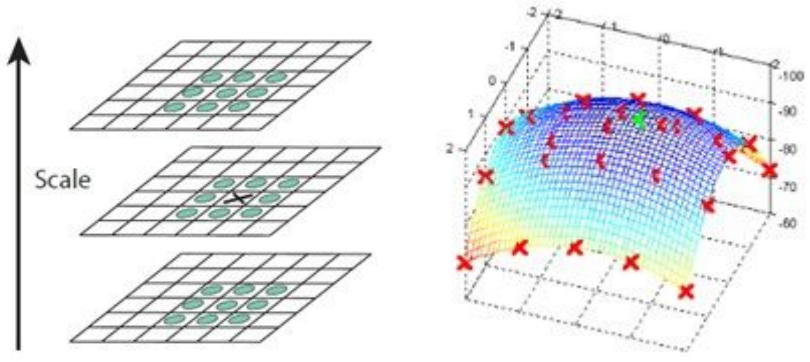
$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$



Obtido o espaço de escalas, o passo seguinte é calcular as imagens relativas aos laplacianos (derivadas de segunda ordem) das escalas gausssianas (LoG´s) obtidas, sobre as quais será feita a busca dos pontos de interesse (pontos SIFT). Conforme ilustrado no esquema da figura anterior, as chamadas LoG´s são computadas de maneira aproximada através da diferença entre duas escalas sucessivas , conhecidas como DoG´s.

2.3.2 Detecção de pontos extremos (candidatos a pontos de interesse)

O passo seguinte é procurar por pontos extremos nas imagens DoG´s computadas anteriormente. Estes pontos extremos são máximos ou mínimos locais em uma vizinhança tridimensional 3x3x3 (x,y,escala), conforme mostra a figura a seguir:



A localização dos pontos extremos é refinada para se obter precisão de sub-pixel, usando um ajuste quadrático baseado na expansão de Taylor.

2.3.3 Eliminação de candidatos com baixo contraste e pontos em bordas

Dentre os pontos obtidos no passo anterior são descartados os pontos com baixo contraste e os pontos que forem bordas, restando somente os pontos de maior contraste considerados

"cantos". O critério é bastante similar ao critério usado no detetor de Harris, conforme mostram as equações a seguir:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

$$\mathrm{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta,$$

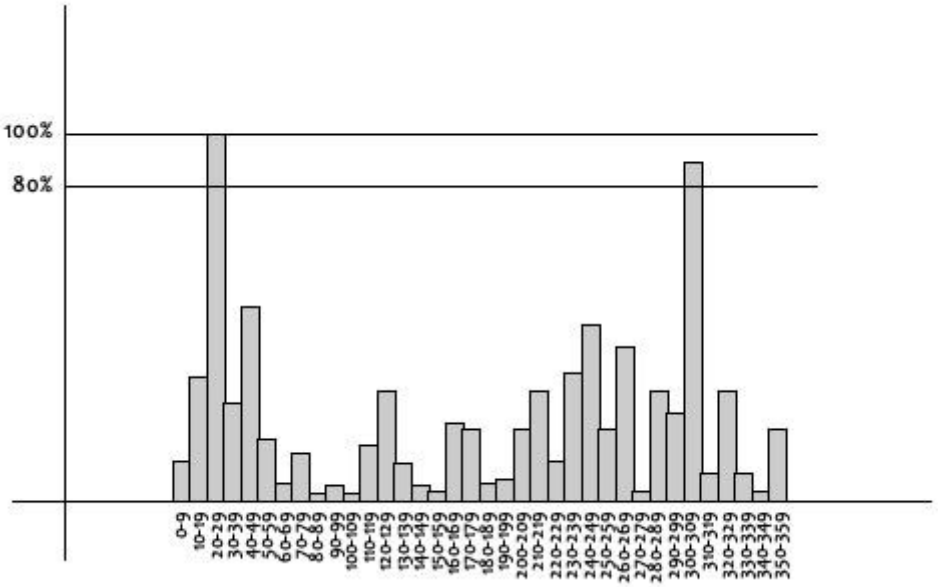
$$\mathrm{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta.$$

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r},$$

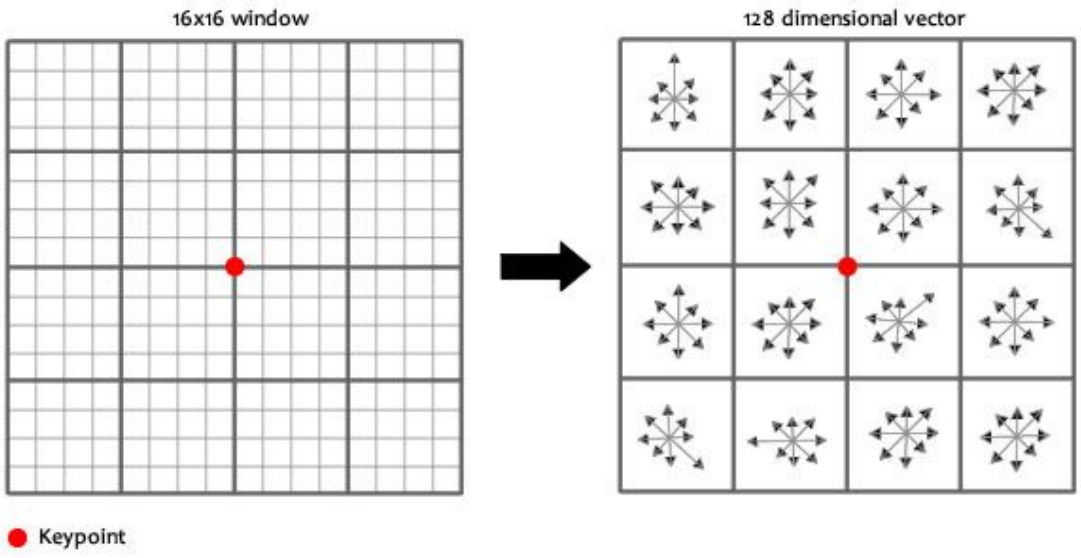
2.3.4 Orientação dos pontos de interesse

A cada um dos pontos de interesse é associado a orientação dominante dentro de uma pequena vizinhança. A magnitude e a orientação do gradiente é calcualada para todos os pixels na vizinhança do pontode interesse, e um histograma de 36 bins (10 graus por bin) é criado para identificar a orientação dominante nesta vizinhança.

A orientação do gradiente de um dado ponto define o bin em que este ponto será computado, e a quantidade a ser computada neste bin é ponderada pela magnitude do gradiente do ponto. O pico deste histograma determina a orientação dominante procurada.



2.3.5 Descritor SIFT (vetor de características)



de maneira bastante parecida ao descritor HoG, o descritor SIFT também é baseado em histogramas populados pelas orientações dos gradientes (ponderados pela magnitude) dos pontos vizinhos ao ponto de interesse, Nessa computação também é utilizada interpolação trilinear. Em sua configuração padrão, é utilizada uma janela (centralizada no ponto de interesse SIFT) de 16x16 dividida em blocos menores de 4x4. Um histograma de orientações/magnitude de 8 bins é associado a cada bloco de 4x4. Para garantir robustez a rotação, cada ponto vizinho é rotacionado da orientação dominante associado ao ponto de interesse central antes da alocação do bin, e a orientação dominante também é subtraída da orientação de cada ponto vizinho. Uma janela gaussiana é utilizada para atribuir maior peso para os pontos mais próximos ao centro da janela.

2.3.6 Adaptação do detector de Harris com o Descritor SIFT simplificado

Neste trabalho também foi realizada uma experiência, combinando o detector de cantos de Harris, com uma simplicação do descritor SIFT. A simplificação do descritor consistiu na remoção das compensações da orientação dominante associada ao ponto de interesse.

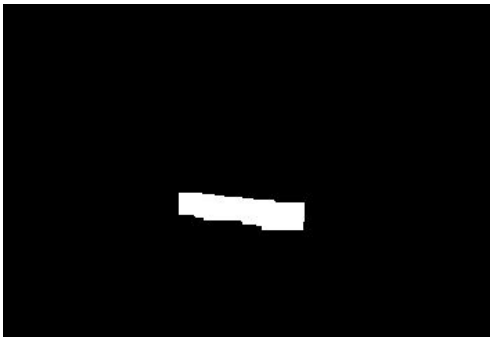
3. Segmentação com watershed

A delimitação da placa foi feito com o algoritmo de segmentação Watershed, utilizando componentes das operações morfológicas dilatação e erosão.

A primeiro foi utilizada a operação de dilatação para pegar a placa inteira mais alguma parte fora da placa e extrair o boxing com a placa Figura c), depois foi usado a operação de erosão para obter os sementes da placa Figura d), seguidamente para os sementes do fundo pegamos quase as

bordas do boxing Figura e), de este jeito esta garantido que os sementes do fundo não van a pegar alguma parte da placa, com a premissa que a detecção este certa. Na Figura f) pode-se ver os basins que se esta utilizando para a segmentação da placa, estes foram obtidos com o gradiente de Sobel, o que atingiu os melhores resultados, mais foi tentando diferentes métodos(gradiente morfológico, melhoramento de contraste com toogle). Depois de fazer a segmentação com Watershed se obteve o label Figura g) e por ultimo foi detectada a borda do label e marcada na imagem original Figura h).

a) Imagem original



b) Detecção da placa

c) Boxing da placa

d) Sementes da placa

e) Sementes do fundo

f) Basins da placa

g) Label

h) Placa segmentada Watershed.

Na figura superior apresenta as tarefas para a segmentação das placas, ainda é um intento porque achamos que pode ser melhorado.

O principal problema sobre o método de segmentação foi achar um bom abordagem para apresentar os basins, foi testado fazendo o gradiente morfológico, uma suavização com o método de Gauss, também o método de Sobel, e algumas combinações como equalização com gradiente morfológico, equalização com Sobel e melhoramento de contraste com Sobel, mas a que atingiu os melhores resultados foi Sobel só.

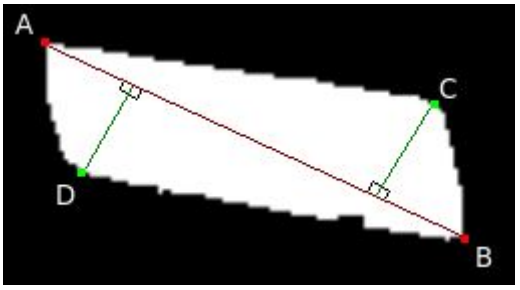
4. Correção da placa

Dado que as imagens nem sempre são tiradas desde uma perspectiva frontal, a posição das letras varia por cada imagem, segundo a perspectiva da qual a imagem foi tirada. Após a segmentação da placa, e antes da separação dos caracteres, é preciso projetar as imagens das placas para as letras ficarem retas e alinhadas horizontalmente.

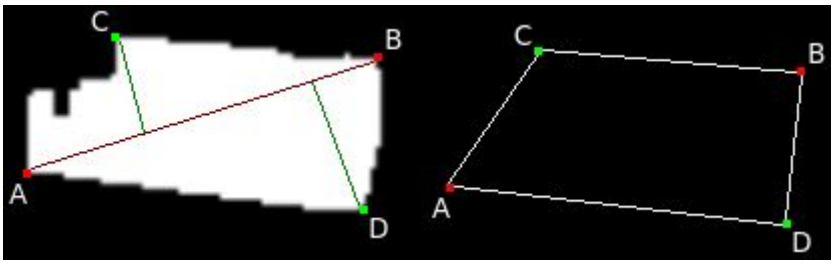
Duas abordagens foram utilizadas para isto: a primeira consiste no método de calibração de câmera e a segunda consiste no alinhamento do objeto por análise de componentes principais (PCA).

A correção por calibração de câmera precisa das coordenadas das quinas do paralelogramo que, em um caso ideal, representa o contorno da placa. Na realidade, o elemento segmentado não tem a forma de um paralelogramo homogêneo, logo uma aproximação é feita. Para achar as quinas do paralelogramo aproximado da placa, acha-se primeiro os dois pontos (A e B) mais afastados dentro do contorno da placa e posteriormente acha-se os dois pontos (C e D)

mais afastados da reta AB, considerando a distância euclidiana mínima de um ponto a uma reta (perpendicular).



Quando a forma da placa é regular, como é o caso da imagem de arriba, a aproximação do paralelogramo e bastante acurada. Porém, nos casos em que a região segmentada não é tão regular, o paralelogramo aproximado não representa a forma que tem a rotação de um objeto rectangular. Tentativas de deslocar alguns pontos para criar paralelogramos mais regulares acabaram com distorcer imagens que anteriormente tinham uma boa projeção.



Após as coordenadas das quatro quinas são encontradas, a matriz de transformação é calculada segundo o método de calibração de câmera, partindo dos quatro pontos iniciais até os quatro pontos alvos, os quais são: (0,0), (200,0), (200,50), (0,50). É importante notar que a ordem das quinas do paralelogramo calculado e do rectângulo final devem coincidir. Assim que a matriz de transformação é achada, ela é invertida para mapear cada pixel do rectângulo alvo (200 x 50) ao paralelogramo calculado; a coordenada encontrada no paralelogramo (a qual pode não ser inteira) é interpolada para encontrar a intensidade correspondente no rectângulo alvo.

Em alguns casos, a matriz de transformação não pode ser achada, devido a que o sistema de equações associado não pode ser resolto. Isto acontece quando a forma da placa e irregular suficiente. Para esses casos, a segunda abordagem é usada. Quando o determinante da matriz que descreve o sistema de equações é 0, o alinhamento por PCA é feito.

As imagens acima mostram a importância da forma do componente para a correção. Formas regulares causam uma boa projeção, enquanto irregularidades no componente são evidenciados no resultado final. A avaliação da projeção apenas pode ser feita qualitativamente: olhando cada resultado. Os casos onde os componentes segmentados não apresentaram uma forma regular, e por tanto causaram projeções erradas, foram poucos. Uma avaliação mais objetiva será feita na separação dos caracteres da placa.

5. Separação de Caracteres

A separação de caráteres depende da correção da placa, se a correção estiver errada, a separação se torna um processo difícil. Consideraram-se duas abordagens baseadas em perfis de projeção: abordagem baseada na aplicação de diferentes tipos de limiarização e no análise de projeção e outra abordagem baseada. As duas abordagens estão baseadas numa etapa de pré-processamento.

Pré-processamento

A etapa de pré-processamento considera os seguintes passos:

1. **Normalização da placa:** Em primeiro lugar, foi aplicado um ajuste de imagem para aumentar o contraste. Os valores de intensidade foram normalizados de modo que os seus valores estejam entre 0 e 255.

2. **Limiarização Local da placa**: Métodos de binarização podem ser classificados em duas classes: globais e locais. Métodos de limiarização globais utilizam um único limiar para a imagem inteira, não fornecem resultados satisfatórios quando a imagem tem iluminação irregular ou sombras. Por outro lado, os métodos de limiarização local calculam um limiar para cada pixel, inspecionando os pixels adjacentes. Considerou-se o método de limiarização global, Otsu, e as seguintes limiarizações locais: Niblack, Sauvola e Phansalkar, elas foram consideradas devido a que apresentam bons resultados em imagens de documentos. Como estamos trabalhando com os caracteres da placa, estas abordagens podem ser de muita ajuda na segmentação dos caracteres.

Podemos ver na seguinte figura exemplos dos resultados dos métodos de limiarização sobre algumas placas. A figura mostra como o método de Otsu não e muito bom. Os métodos de limiarização local apresentaram melhores resultados. Neste exemplo podemos ver o problema de ruído por causa da projeção da placa. O método de Phansalkar resultou com mais ruído do que os demais métodos.

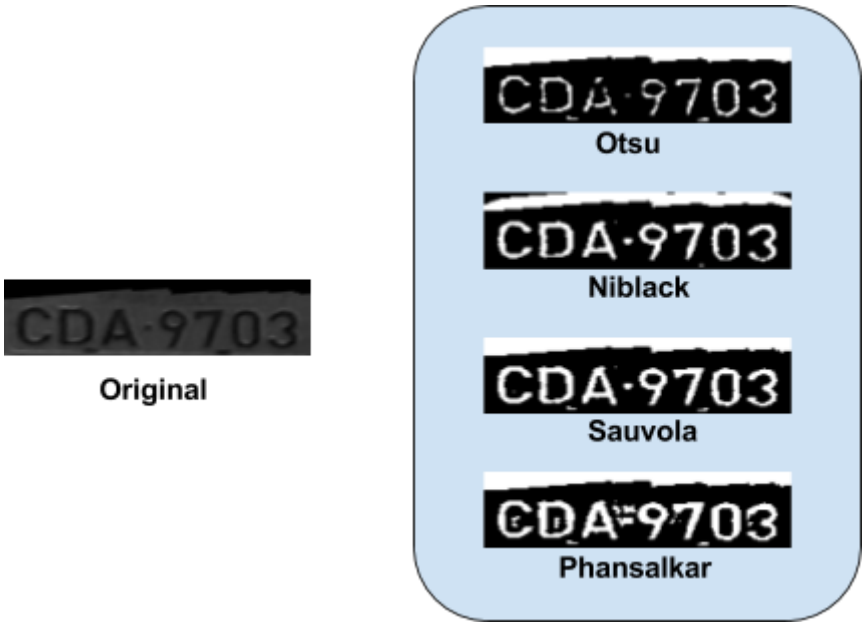


Figura: Limiarização da placa

Porém a seguinte figura apresentam casos onde só alguns dos métodos de limiarização local apresentam melhores resultados. Neste exemplo temos uma placa de baixo contraste, o método de Otsu não detectou os caracteres, o método de Niblack resultou com mas ruído, um problema que pode ser visto é que as vezes alguns caracteres são separados, isso pode afetar a separação geral devido a que um caracter pode ser considerado como dois caracteres.

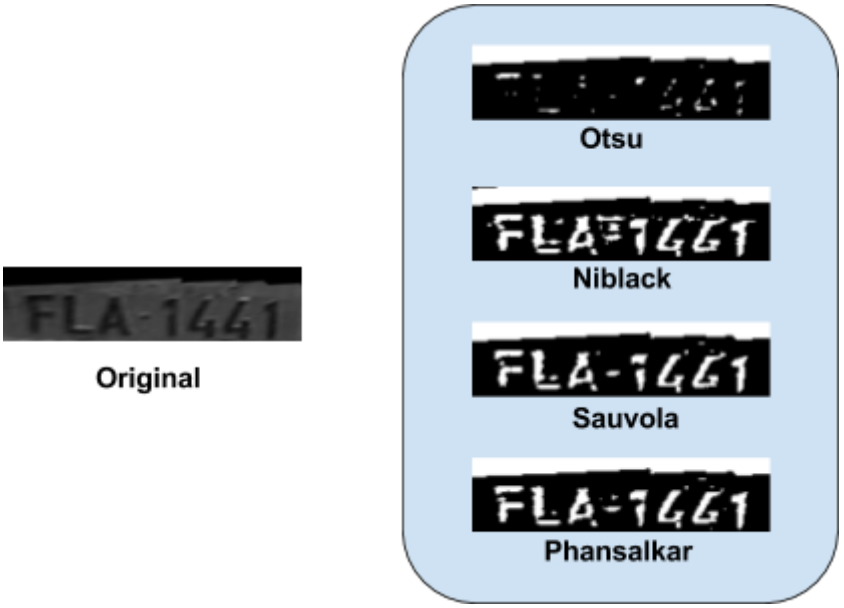
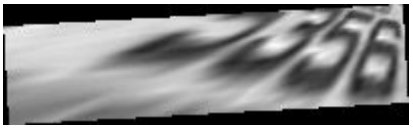


Figura: Limiarização da placa

Ainda na etapa de pré-processamento da placa buscou-se por métodos que poderiam ser aplicados para corrigir o formato da mesma, visto a importância desse no tratamento da

separação dos caracteres.

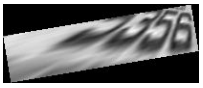
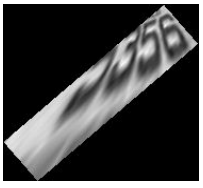
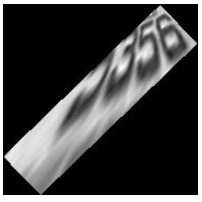
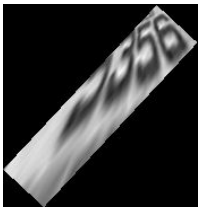
Considere então a imagem da placa abaixo, que é o caso mais ruim da segmentacao:



Pela orientação dos caracteres a projeção vertical da imagem não oferecia resultados satisfatórios que permitissem ao algoritmo particionar os caracteres de forma apropriada.

Tentou-se então a rotação da imagem, buscando uma melhor orientação da placa e portanto, valores da projeção vertical que pudessem demonstrar de forma mais precisa, a existência de um caracater na região analisada.

Segue abaixo alguns exemplos da rotação realizada com os valores da projeção vertical.

				
503.01501	484.88614	481.19116	478.08292	476.42233
				
476.35922	478.05853	483.36765	483.60397	491.83920
				
497.33163	508.42487	524.34576	530.28259	**547.73035**

Infelizmente, vê-se que a projeção não melhorou com a rotação, em parte pelo redimensionamento da imagem através a rotação, mas em parte pela característica do problema, onde mesmo com a rotação da imagem, não se pode identificar mudança sensível nos valores da projeção. Veja que a imagem selecionada pelo algoritmo continuou sendo a imagem **15** ao invés da imagem **11**, que apresentava uma melhor orientação, ao menos no caracter "6" de onde poderíamos encaixar um template, por exemplo.

Duas tentativas foram realizadas para tentar isolar o problema de redimensionamento da imagem:
(1) Tentou-se alterar a escala da imagem rotacionada, no entanto, os ruídos introduzidos pela distorção do caracter vieram a inviabilizar essa tentativa
(2) Tentou-se também usar como critério de escolha da melhor rotação a razão entre a projeção horizontal e vertical da imagem, gerando-se os valores acima mencionados.

Viu-se então que tal abordagem não produz resultados que possam contribuir com o pós-processamento da placa para a identificação da imagem e esse passo foi desconsiderado para a sequência de identificação

Separação de caracteres

Após a etapa de preprocessamento as abordagens consideradas foram:

1. **Abordagem baseada na combinação dos métodos de limiarização e projeção**

   Como vimos anteriormente, os métodos de limiarização podem apresentar diferentes resultados para diferentes placas. Nesta abordagem foram considerados todos os métodos de limiarização local, os passos da abordagem são os seguintes:

   a. **Projeções horizontais e verticais:** As projeções consistem na contagem de pixels existentes numa determinada linha ou coluna. O resultado da projeção horizontal e vertical de uma placa é mostrado na seguinte figura.

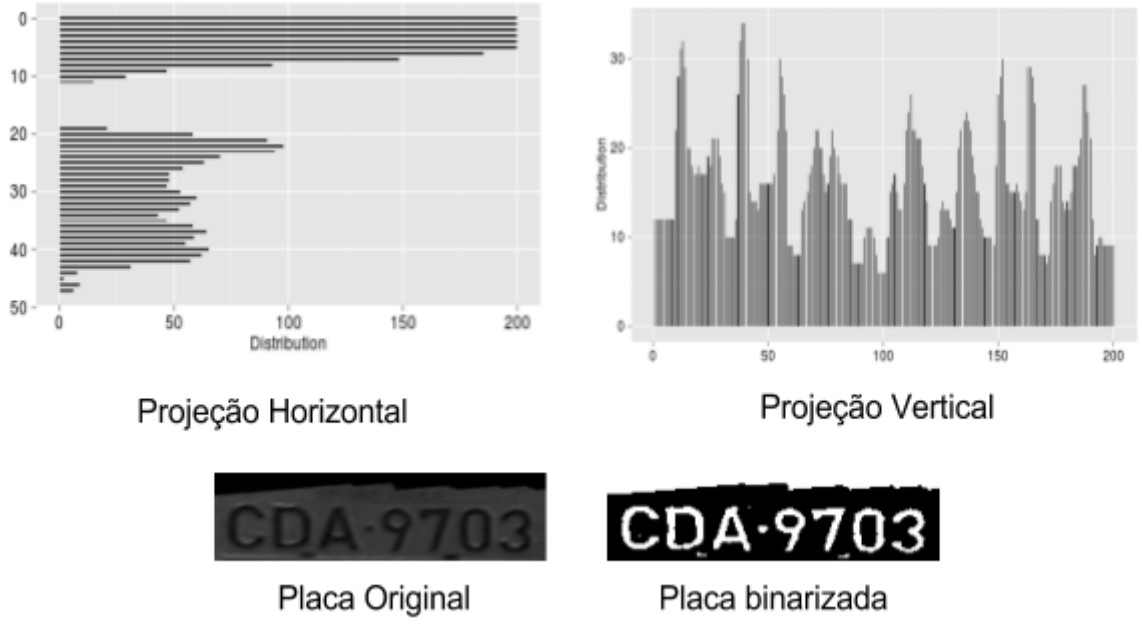


Figura: Projeções horizontais e verticais da imagem original

A figura mostra o comportamento das projeções da imagem binarizada. A projeção horizontal permite que se determine a posição inicial e final das letras no eixo y. A projeção vertical permite separar os caracteres.

b. Eliminação do ruido superior: De acordo com a figura anterior, a projeção horizontal mostra um pico grande na parte inicial por causo do ruído. Essa quantidade de pixels afetou a projeção vertical, isto é, os mínimos globais nunca terão um valor próximo a 0. A seguinte figura mostra as projeções horizontais de diferentes placas.

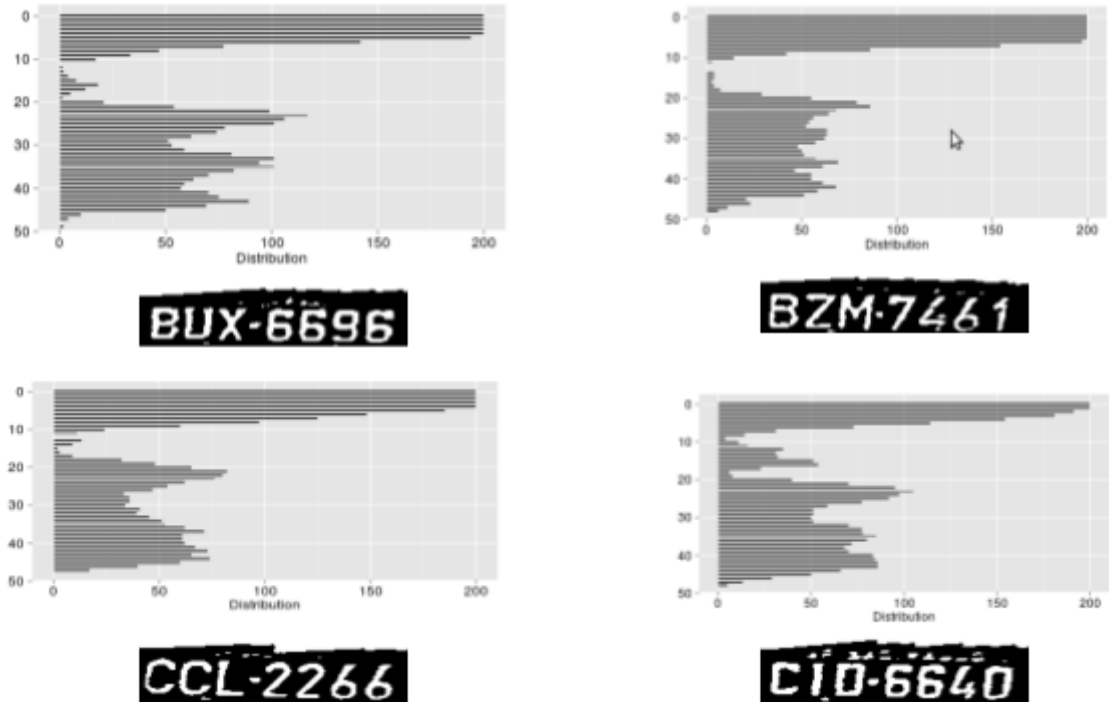


Figura: Visualização de ruido superior nas projeções horizontais

Podemos ver que as imagens quase sempre apresentam um ruído na parte superior por tanto foi determinado um limiar fixo de aproximadamente 16 pixels na parte superior, isso foi feito devido a que algumas vezes o ruído é variável, isto é, algumas vezes a projeção horizontal apresenta mais de um pico na parte do ruído. A seguinte figura mostra os resultados após a correção do ruído.

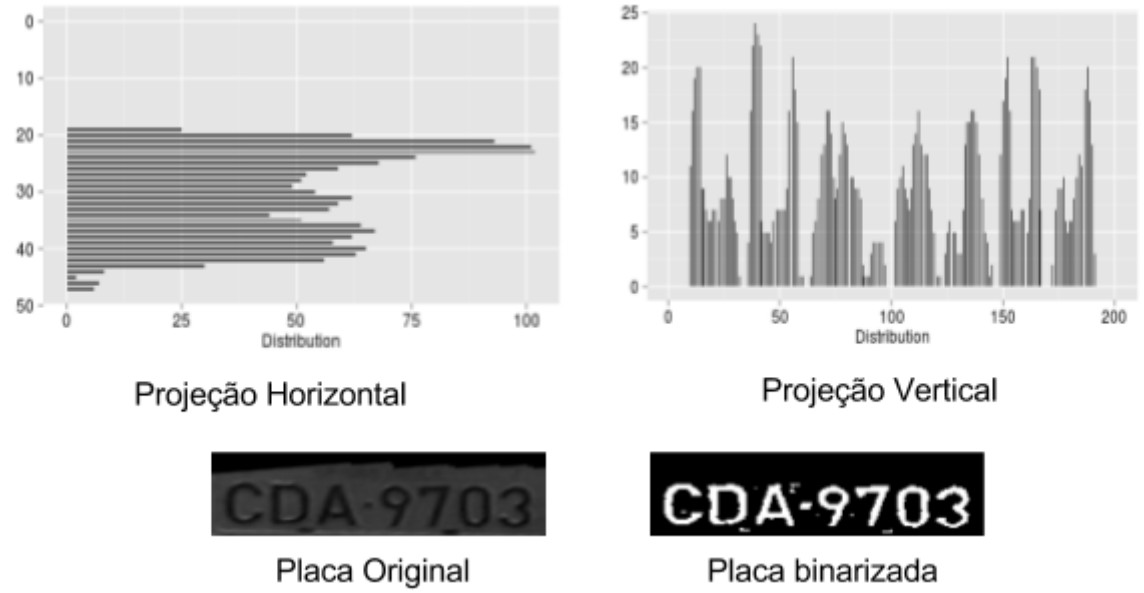


Figura: Projeções horizontais e verticais da imagem sem ruído superior

c. Calculo de mínimos baseado na projeção vertical: O seguinte passo foi o cálculo dos valleys na projeção vertical, como é possível ter valleys próximos a zero, um limiar foi fixado para determinar si um valley será considerado como candidato na separação de caracteres. Depois foi considerado outro limiar para filtrar os valleys entre picos muito pequenos, isto foi feito para eliminar pequenos ruídos que ainda estejam presentes entre os caracteres e para somente considerar aqueles valleys que separem picos das alturas maiores que normalmente são os caracteres da placa. Finalmente foi considerado outro limiar sobre a largura mínima de um caracter. A seguinte figura mostra um exemplo de possíveis candidatos a valley.

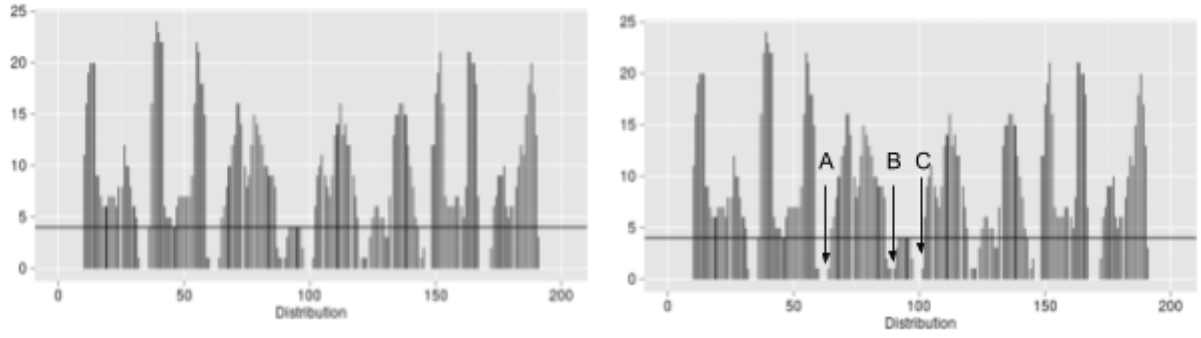


Figura: Valleys candidatos

A figura anterior mostra os candidatos do primeiro limiar, todos os mínimos que estão abaixo do limiar são os primeiros candidatos. Entre os valleys candidatos, consideraram-se aqueles que estão entre picos altos. por exemplo A, B e C. Candidato *A* esta no meio de dois picos altos, por tanto ele é considerado um valley que começa num pico e termina em outro pico. O candidato B é um valley que tem um pico alto na

esquerda, então ele é considerado um valley que termina em um pico. Finalmente C é um valley que tem um pico alto na direita, então ele é considerado um valley que começa num pico.

d. Contagem de caracteres: Após o processamento de mínimos, a contagem de caracteres foi feita para cada método de limiarização, Foram consideradas contagens de caracteres maiores a 6 por que e possível ter placas com 6 caracteres e 7 caracteres. A prioridade foi para 7 caracteres, isto é, primeiro se faz uma verificação, se algum método de limiarização conseguiu encontrar essa quantidade de caracteres, então ele um candidato para o resultado final. Se tivermos mais de um candidato, a eleição e feita sobre aquele que tenha maior variação na projeção vertical, isto foi feito porque o candidato com maior variação nos picos, terá uma maior probabilidade de uma melhor separação.

e. Obtenção de caracteres: Após da eleição do método de limiarização final e a contagem de caracteres, a obtenção de caracteres foi feita separando cada caracter de acordo com o limiar superior e os valleys achados anteriormente. A seguinte figura mostra os resultados da separação final de caracteres**.**

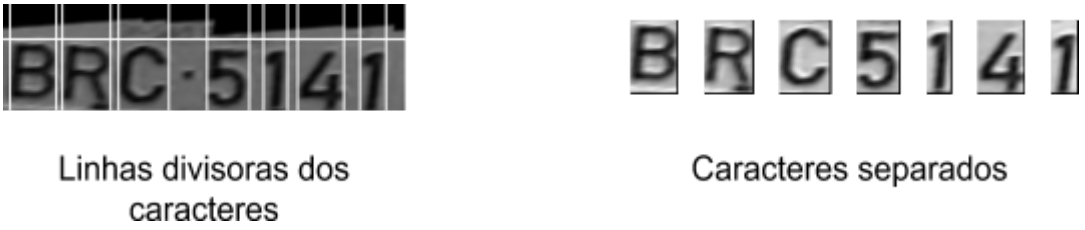


Figura: Separação de caracteres

2. Abordagem baseada em projeções e limiares para detecção de caracteres e eliminação de caracteres muito "estreitos" e "baixos".

A seguir temos a descrição de uma segunda abordagem para segmentação de caracteres implementada no código entregue para avaliação. As etapas de pré processamento de normalização e binarização são basicamente as mesmas utilizadas na abordagem anterior (vide passos 1 e 2 na descrição a seguir). Os conceitos de projeções horizontal e vertical também são utilizados, e as diferenças estão na implementação da determinação das fronteiras dos caracteres através de detecção de transições nos vetores de projeções e processamentos adicionados para eliminação de caracteres muito "estreitos" e muito "baixos".

- **Passo 1: Normalização da imagem**
  - Foi utilizada uma normalização com adjacência circular de raio 5.
- **Passo 2: Binarização da imagem**
  - Neste passo não foi utilizado uma combinação de algoritmos de linearização. Um entre os 4 algoritmos listados a seguir foi escolhido e testado separadamente. Os melhores resultados foram obtidos usando o algoritmo Niblack:
    - Otsu

        - Niblack: vizinhança retangular de 25x25, k = -0.2
        - Savoula: vizinhança retangular de 25x25, k=0.1 e R=128
        - Phansalkar: vizinhança circular com raio 11, k= 0.05, R=128, p=2 e q=10
- **Passo 3: Projeção Horizontal**
    - 3.a) Computamos o vetor de projeção horizontal, "h_proj[j=0 .. j=ysize-1]" onde cada elemento h_proj[j] armazena a soma da intensidade dos pixels da linha j da imagem.
    - 3.b) O vetor de projeção horizontal, "h_proj", é varrido na direção "de baixo para cima" (j=y_size-1, j=y_size-2, j=y_size-3, ...), buscando-se a primeira transição "de baixo para cima" (ou seja, procura-se o ponto em que valor de hproj[j] se torna maior que um threshold th_0). Esta transição é armazenada e identificada como o limite de busca inferior da imagem: "bottom_limit".
    - 3.c) O vetor de projeção horizontal, "h_proj", continua sendo varrido na direção "de baixo para cima", mas agora buscando-se a próxima transição "de cima para baixo" (ou seja, procura-se o ponto em que valor de hproj[i] se torna inferior a um threshold th_1 = th_0/2). Esta transição é armazenada e identificada como o limite de busca superior da imagem, "upper_limit". Após alguns experimentos, verificou-se que o valor do upper_limit poderia ser fixado sempre 17 linhas acima do bottom_limit (ou seja, upper_limit = bottom_limit - 17), e para se obter melhor robustez a ruídos presentes na parte superior das placas, este foi a abordagem utilizada
- **Passo 4: Projeção Vertical**
    - 4.a) Computamos o vetor de projeção vertical, "v_proj[j=0 .. j=xsize-1]" onde cada elemento v_proj[j] armazena a soma da intensidade dos pixels da coluna j da sub-imagem limitada pelas linhas de j=upper_limit até j=bottom_limit.
    - 4.b) O vetor de projeção vertical, "v_proj", é varrido na direção "da esquerda para a direta" (i=0,1,..), buscando-se uma transição "de baixo para cima" (ou seja, superação de threshold tv_0). Esta transição é armazenada e identificada como o inicio de um caracter da placa: "char_begin".
    - 4.c) O vetor de projeção vertical, "v_proj", continua sendo varrido na direção "da esquerda para a direta" (i=0,1,..), mas agora buscando-se a próxima transição "de cima para baixo" (ou seja, procura-se o ponto em que valor se torna inferior a um threshold tv_1). Esta transição é armazenada e identificada como o final de um caracter da placa, "char_end".
    - 4.d) Os limites "char_begin" e "char_end" definem um candidato a caracter da placa, para que ele seja assumido como tal, dois requisitos devem ser atendidos:
        - a “largura” do caracter candidato deve ser maior que um valor tw.
        - a "altura" do caracter candidato deve ser maior que um valor th. Esta altura é basicamente calculada usando uma projeção horizontal sobre a sub-imagem definida pelo caracter candidato
    - 4.e) O vetor de de projeção vertical, "v_proj", continua sendo varrido na direção "da esquerda para a direta" (i=0,1,..), e as tarefas mencionadas nos passos 4.b a 4.d, são repetidas até que 7 caracteres sejam encontrados ou vetor de projeção seja totalmente percorrido.

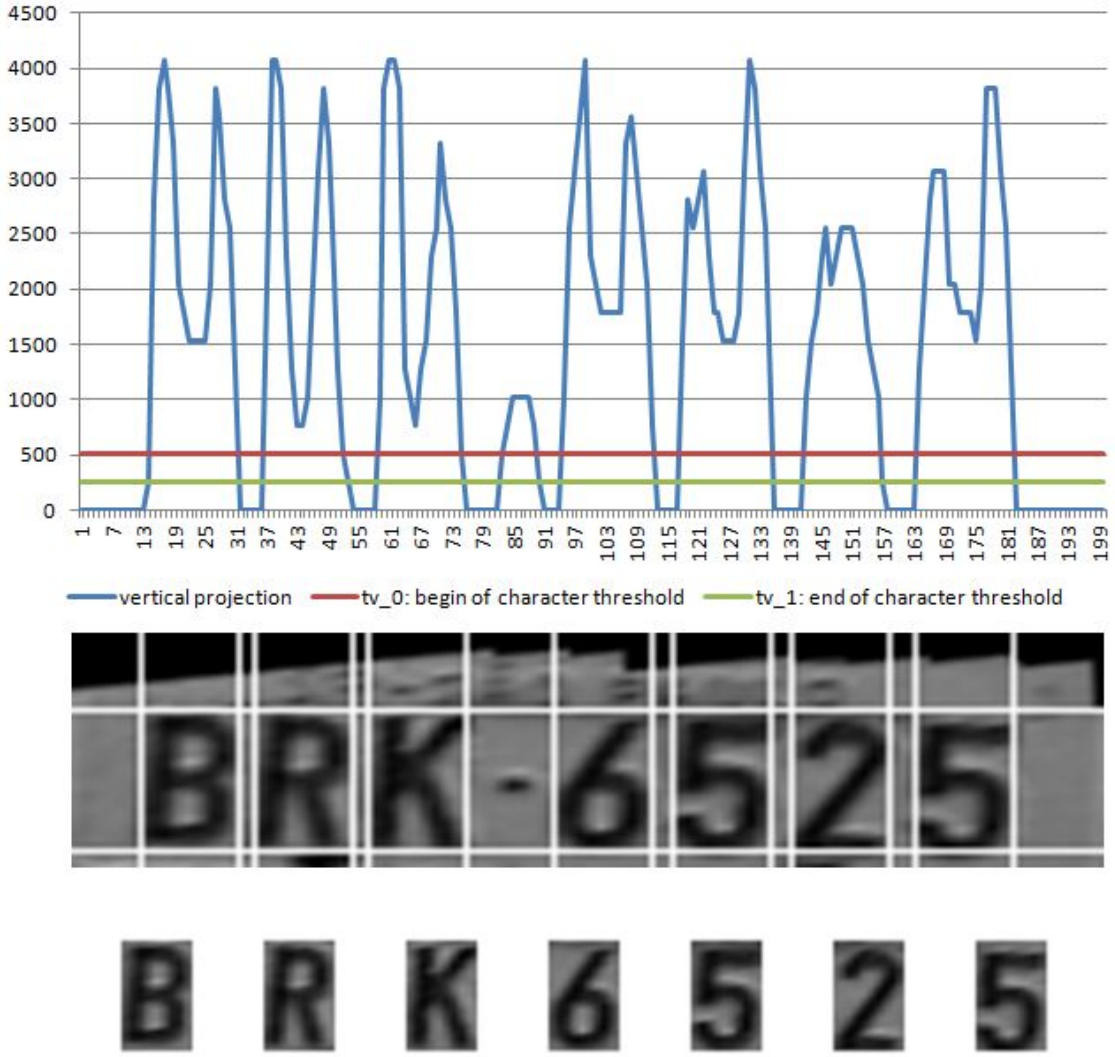


Figura - Projeção vertical e limiares usados na detecção das fronteiras dos caracteres.

- **Passo 5: Alargamento de caracteres estreitos**
  - 5.a) Para caracteres com largura menor que a largura média Lm, são tem sua largura expandida de Lm/2 para esquerda e para direita, respeitando-se as fronteiras com os caracteres vizinhos

6. Resultados

5.1. Detecção da placa

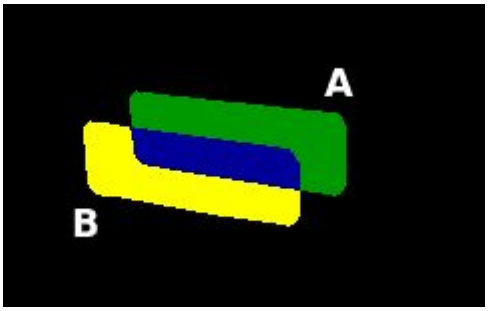
Método	Parâmetros	Total Candidatos	Falsos Positivos	Falsos Negativos	Precisão
LBP	r = 1.0 p = 8	1043	17 (1,2%)	7 (0,5%)	98,72 %
	r = 3.0 p = 10		11 (0,78%)	17 (1,2%)	98 %
HOG com normalização padrão	células = 8x8 blocos = 2x2 sobreposição = 1x1 bins = 9		4 (0,28%)	9 (0,64%)	99.07%
HOG com normalização proposta			7 (0.49%)	4 (0.28%)	99.22%
HOG com equalização			5 (0,35%)	8 (0.57%)	99.07%
SIFT	4 pontos de interesse	198 (treinamento) 792(teste)	164 (14%)	137(12%)	77%
	5 pontos de interesse		124 (11%)	116(10%)	80%
HARRIS+ SIFT descritor simplificado	5 pontos de interesse	198 (treinamento) 792(teste)	90(8%)	121(11%)	79%
	6 pontos de interesse		170(15%)	43(4%)	88%
	7 pontos de interesse		158(14%)	28(2%)	90%

5.2. Segmentação da Placa

A segmentação foi feita com os resultados da detecção com HOG, já que este classificador foi o que obteve os melhores resultados.

Para a medição dos resultados foi utilizado o coeficiente de Jaccard que é a medida da intercepção dos pixel entre a união.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}.$$



Onde A é o conjunto dos pixels depois de segmentação e B são os pixel da base de dados label.

Jaccard com LBP : 94.28%
Jaccard com HOG: 96.25%

5.3 Separação de caracteres

Testaram-se as duas abordagens, alguns problemas foram devido a que a projeção da placa não foi muito boa para alguns casos. A seguinte figura mostra placas que não foram corretamente projetadas.

Figura: Problemas de projeção

Para esses casos a separação de caracteres somente tenta separar os caracteres que estão presentas na placa.

Não se tem uma forma exata de medição para saber se os caracteres separados foram os corretos, é possível que se tenham falsos positivos devido a que na limiarização algum caracter pode ter sido dividido em dois caracteres ou por algum caracter adicional como o hífen que separa alguns caracteres. A primeira abordagem apresenta problemas desse tipo, é por isso que essa abordagem gera falsos positivos, os problemas dessa abordagem estão no uso de muitos limiares estáticos na definição de valleys, por exemplo o limiar do ruído da parte superior da imagem, a máxima altura possível entre valleys e outros. Assim como o tempo de execução pelo uso de diferentes métodos de limiarização.

Sobre a abordagem 2, ela também e fortemente baseada em parâmetros de limiar que necessitam ser ajustados experimentalmente. Uma das vantagens da segunda abordagem sobre a primeira é o tempo de execução. Os resultados foram obtidos usando-se imagens segmentadas e rotacionadas com procedimentos específicos que geram distorções e ruídos particulares que foram compensadas através do ajuste dos referidos limiares. Uma mudança nestes procedimentos ou outras alterações na qualidade das imagens testadas certamente requereriam um reajustamento destes parâmetros.

A medição usada neste trabalho esta baseada na quantidade de caracteres detectados. Porém, como o dito anteriormente não se tem uma medição exata dos caracteres detectados.

Método	**Parâmetros**	**Caracteres Detectados**				**Precisão**
		8	**7**	**6**	**Outros**	
Abordagem 1	- Vizinhança Rectangular: 25x25 Niblack: k=-0.2 Sauvola: k=0.1, R=128 - Vizinhança Circular: 17 Phansalkar: k= 0.05, R=128, p=2, q=10	52	747	101	90	85.65%
Abordagem 2	Otsu	0	333	184	473	52.22%
Abordagem 2	**Niblack: k = -0.2** **Vizinhança retangular: 25x25**	**0**	**829**	**88**	**73**	**92.63%**
Abordagem 2	Sauvola: k=0.1, R=128 Vizinhança retangular: 25x25	0	795	114	81	91.82%
Abordagem 2	Phansalkar: k= 0.05, R=128, p=2, q=10 Vizinhança Circular: 17	0	822	79	89	91.01%

7. Contribuição do trabalho

- Marcos Piaia: implementação da abordagem de seleção de placas usando a DFT/DCT. Implementação do procedimento de rotação para correção da placa como fase de pré-processamento à segmentação de caracteres.
- Jhosimar Arias: implementação de HOG e testes com diferentes parâmetros, implementação dos métodos de limiarização local: Niblack, Sauvola e Phansalkar, implementação da abordagem de separação de caracteres baseado na combinação dos métodos de limiarização e 31projeção.
- Juan Hernández: Implementação de LBP e testes correspondentes. Correção da placa: calibração de câmera com mapeamento inverso e teste da abordagem de alinhamento por PCA.

- Darwin Saire: Pré-processamento com morfologia matemática, implementação e possíveis sementes para a delimitação e segmentação com watershed, poscorrecção na placa.
- Ricardo Nishihara: Estudo do detector e do descritor SIFT (implementação inicial). Implementação de uma adaptação: detector de corners de Harris + descritor SIFT. Implementação da abordagem 2 para segmentação de caracteres usando projeções e limiares para detecção das fronteiras dos caracteres e eliminação de caracteres muito "estreitos" e "baixos".

8. Referências

[1] R.Parisi, E.D.Di Claudio, G.Lucarelli and G.Orlandi, "Car plate recognition by neural networks and image processing", Conference Paper, 1998, DOI: 10.1109/ISCAS.1998.703970 Source: IEEE Xplore

[2] Distinctive Image Features from Scale-Invariant Keypoints, David G. Lowe (link: https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf)

[3] Implementing the Scale Invariant Feature Transform(SIFT) Method - Yu Meng , Dr. Bernard Tiddeman (link: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.102.180)

[4] Deng Hongyao, Song Xiuli "License Plate Characters Segmentation Using Projection and Template Matching" - 2009 International Conference on Information Technology and Computer Science