

RECONHECIMENTO DE PLACAS DE VEÍCULOS POR IMAGEM

Aura Conci

PGMEC/UFF - Passo da Pátria 156 Niterói, RJ - 24210 -240 aconci@ic.uff.br

Leonardo Hiss Monteiro

PGCC/UFF - Passo da Pátria 156 Niterói, RJ - 24210 -240 aconci@ic.uff.br

RESUMO: *Atualmente o desenvolvimento tecnológico aliado ao relativo baixo custo das filmadoras e máquinas fotográficas digitais tornam viáveis utilizações de reconhecimento por imagens aplicadas à problemas do cotidiano como o reconhecimento de placas de veículos. Os dispositivos eletrônicos para aplicar multas nos veículos utilizados nos departamentos de trânsito da maioria dos estados do país, não dispõem de nenhum sistema de reconhecimento automático. Também os sistemas de controle eletrônico de estacionamento, onde o comprovante é emitido automaticamente, mas sem nenhuma identificação da placa do carro, são exemplos de aplicações imediatas do trabalho que aqui é apresentado, que mostra como é possível obter resultados corretos na identificação automática das placas de veículos, depois de sua imagem ser capturada e processada adequadamente por técnicas de processamento de imagens e reconhecimento ótico de caracteres.*

Palavras chave: *Processamento de imagens; Reconhecimento de padrões; Reconhecimento ótico de caracteres (OCR)*

1. INTRODUÇÃO

Cada vez mais, observa-se a utilização de computadores como auxiliar do homem na execução de tarefas que antes eram executadas visualmente. Na última década observamos um crescimento do uso de computadores em tarefas que antes pareciam ser impossíveis tais como o reconhecimento do conteúdo de imagens e cores. O desenvolvimento dessas áreas só foi possível devido ao fato de que só no últimos anos conseguimos obter máquinas com capacidade suficiente para executar essas tarefas ligadas ao reconhecimento de padrões visuais complexos.

Atualmente existem sistemas de são capazes de ajudar em reconhecimento de dados meteorológicos e econômicos a partir de fotos de satélite; no auxílio ao diagnóstico médico por imagens como nas tomografias computadorizados e no reconhecimento de elementos importantes em radiografias; na contagem de células em exames biológicos e de objetos em diversas aplicações industriais; na identificação de defeitos e no controle de qualidade e metrologia de indústrias automatizadas, no reconhecimento de faces, expressões na e leitura labial e muitas outras aplicações. Todas estas aplicações visuais são fundamentadas no desenvolvimento de técnicas que possibilitam usar os computadores trabalhando com imagens. Estas imagens depois de capturadas devem ser processadas e reconhecidas, possibilitando a execução de tarefas muitas vezes extremamente trabalhosas e repetitivas, ou que sem a visão de máquina seria impossível existir como as provenientes de sonares, radares, ultra-som, ultravioleta, infravermelho, térmicas, raios x, etc.

2. TÉCNICAS UTILIZADAS NO RECONHECIMENTO

No desenvolvimento utilizou-se várias técnicas para realizar o reconhecimento. Essas técnicas vão desde técnicas para o reconhecimento da posição da placa a se identificada até métricas desenvolvidas para estabelecer de qual padrão está mas próximo cada caracter da imagem que está sendo analisada. Este sistema poderia também ser utilizado para o reconhecimento de elementos alfanuméricos em qualquer outro tipo de imagens. O que seria útil, por exemplo, no reconhecimento de embalagens ou da numeração de *containers*. Neste último caso, poderia ser útil em estacionamentos portuários ou em pátios de carga e descarga onde se conseguiria um controle praticamente sem falhas e a possibilidade de associar o sistema a um banco de dados. Na forma atual o sistema é empregáveis nos DETRAN para aplicação de multas eletrônicas sem que seja necessário manter uma equipe realizando a mesma função. Uma qualidade superior se

Diversos métodos de **computação visual** são empregados, combinando aspectos de computação gráfica, processamento de imagem e inteligência artificial. A implementação do sistema compreende a implementação de **diversas etapas**. Na primeira etapa a imagem é **adquirida**. Na segunda, a imagem é **pré-processada**, visando identificar a região de possível placa nas imagens. Na terceira, só a região da placa (como mostrada na Fig. 1) é processada visando obter características de cada caracter que extraídas possibilitem sua análise. Finalmente os caracteres são **classificadas** para posterior identificação da placa. A seguir apresentaremos as técnicas que foram utilizadas no processamento das duas últimas etapas do processa, detalhes sobre as etapas iniciais do processamento destas imagens, estão disponíveis em outros trabalhos.

2.1 Binarização

Para fazer a separação do fundo da região da placa dos elemntos que desejamos analisar utilizamos a binarização ou limiarização (thresholding). A binarização é o método mais simples de segmentação de imagens. Resumidamente consiste em separar um imagem, originalmente em tons de cinza para que tenha só pixels pretos e brancos. Essa decisão da nova cor que terá o pixel é realizada de acordo com o ponto de corte (threshold). Qualquer pixel com intensidade menor ou igual ao ponto de corte passa a ser preto. Se o pixel tiver intensidade maior que o ponto de corte passa a ter a cor branca. Como sabemos uma imagem digital pode ser escrita como uma função $f(x,y)$, a resposta da binarização é a função $g(x,y)$, dada por:

$$g(x,y) = \begin{cases} R_1 & \text{se } f(x,y) \leq T \\ R_2 & \text{se } f(x,y) > T \end{cases}$$

onde R_1 e R_2 são os valores estipulados para os níveis de cinza da imagem binaria , no caso utiliza-se 0 (preto) e 255 (branco). Alguns autores se referem ao método de binarização como um método para retirar o fundo (background) da imagem. Por exemplo na primeira imagem da Fig.1 observemos uma imagem os objetos de interesse em preto. O ponto de corte nesse caso deve ser escolhido com o cuidado para que não capture elementos que não façam parte dos objetos. O valor escolhido automaticamente para essa imagem é 101, como mostrado.

A escolha do ponto de corte T é fundamental. Uma ferramenta auxiliar na escolha deste ponto de corte é o histograma da imagem. A observação desse gráfico permite a localização do melhor valor de T para a imagem. Esta localização é tão mais fácil quanto mais bimodal for o histograma. Quando a imagem, por exemplo, for composta objetos e fundo de cores distintas mas uniformes, podemos estabelecer através de técnicas que ajudam a definir o ponto de corte através de um isolamento das regiões que representem o fundo e os objetos. Para imagem compostas por mais de um objeto de tons diferentes sob um fundo constante, pode-se utilizar técnica de binarização multi-nível (multilevel thresholding).

No caso real de placas a iluminação não uniforme, impossibilita ter tons de fundo e objetos constantes, e não é possível estabelecer as regiões de tons precisas de cada objeto no histograma, de modo que se torna mais difícil encontrar o ponto de corte ideal. Como o trabalho precisava de métodos automáticos para a escolha e ajuste do ponto de corte, depois de diversos testes usando histogramas da imagem reais como referência comparados com conhecimentos prévios sobre parâmetros da imagem e pontos de corte ideais também conhecidos utilizamos o método de Otsu para identificação automática do ponto de corte (Hiss Monteiro,2002).



Figura 1. Exemplo de uma imagem inicial a ser analisada nesta etapa, seu histograma (bimodal) à direita com a linha vertical mostrando o ponto de corte usado para a separação entre o fundo e os objetos. Na imagem inferior a direita observa-se o resultado: a imagem bitonal.

Esse método é baseado na partição do histograma da imagem com L níveis de cinza em duas classes C_0 e C_1 , ou duas regiões que definam um limiar ótimo. Esta otimização é definida pela maximização da função critério, descrita pela razão entre a variância entre as classes que o limiar definir no histograma da imagem usando o ponto de corte a ser definido e o histograma total da imagem:

$$\eta = \frac{\sigma_b^2}{\sigma_t^2}$$

onde a variância entre classes, será expressa por $\sigma_b^2 = w_0(\mu_0 - \mu_t)^2 + w_1(\mu_1 - \mu_t)^2$, sendo μ_0 a média, ou valor esperado para a classe C_0 , descrita pelo histograma com tons de cinza variando de 1 a t , e μ_1 é a média, ou valor esperado para a classe C_1 , com tons variando de $t+1$ a $M-1$, μ_t é a média do histograma total. Fazendo-se $n_i = h(i)$, aonde $h(i)$ é a frequência do tom de cinza i no histograma e considerando N ou total de pixels da imagem. A fração $p_i = (n_i / N)$ representa o percentual de cada tom de cinza no total da imagem. Assim w_0 será o somatório de p_i na classe C_0 , ou seja, variando-se i de 1 até t , e w_1 o somatório de p_i na classe C_1 , ou seja, variando-se i de $t+1$ até $M-1$. Finalmente o denominador da expressão anterior indica a variância do histograma total da imagem a ser binarizada. O resultado obtido pelo método pode ser observado na imagem esquerda inferior da Fig.1.

2.2 Erosão

Erosão foi a técnica utilizada para resolver um problema que apareceu freqüente nesta aplicações real. Imagine que durante a captura da placa os caracteres que desejamos reconhecer pareçam estar contato um com o outro de forma que pareçam um único objeto, ou ligados a ruídos que modifiquem suas formas (como os mostrado pelas pequenas elipses marcadas na imagem superior esquerda da Fig. 2). Nesse caso teríamos sérios problemas para reconhecê-los.

Ficou evidente então a necessidade de desenvolver algum pré-processamento que seja capaz de separar objetos que possivelmente venham estar em contato entre si, devido ao uma maior sombreamento, ou pelo movimento do carro. A solução adotada para esse tipo de problema foi a utilização de um recurso conhecido como **erosão**. Esse recurso faz com que uma imagem original sofra diminuição de tamanho sem que perca suas características geométricas que são de suma importância para o reconhecimento da mesma.

Para o processo de erosão primeiramente escolhemos o elemento estruturante que vai ser utilizado para realizar a "erosão da imagem". A escolha do elemento que vai fazer a operação é muito importante pois esse elemento é que vai determinar o quanto a imagem vai reduzir de tamanho cada vez que é submetida ao processo. Outro fato relevante é a perda das características geométricas da imagem que pode ser causada por uma má escolha do elemento. No programa foi adotado um elemento simples mas que depois de todos os testes se mostrou capaz de garantir que a imagem original não perdesse suas características. A Figura 2 apresenta uma placa antes e depois do processo de erosão e o elemento que foi utilizado na operação.

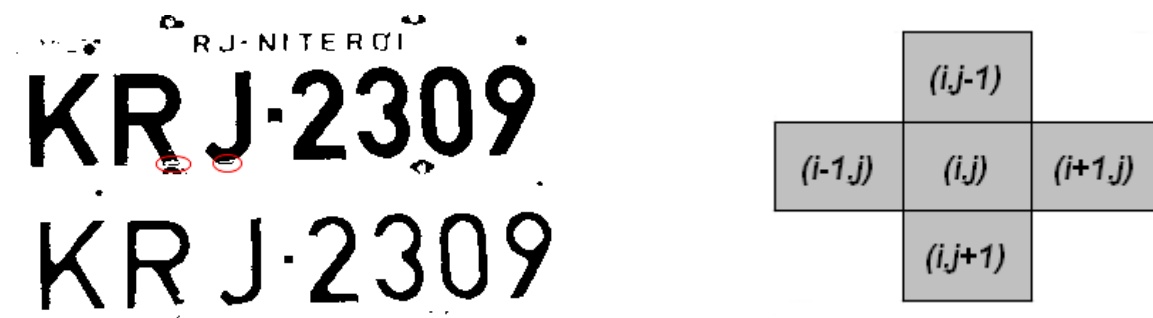


Figura 2. Uma placa antes (imagem superior esquerda) e depois (imagem inferior direita) de submetida ao processo de erosão, e elemento (à direita) utilizado para fazer a erosão.

Depois da escolha do elemento a ser utilizado devemos analisar a vizinhança de cada um dos pixels da imagem através do elemento. Fazemos essa análise da seguinte forma: Colocamos o pixel a ser analisado na coordenada (i,j) e a partir daí analisamos os outros pixels do elemento. Se todos eles estiverem acesos (com o valor 1) o pixel que esta sendo analisado continua aceso. Caso contrário o pixel é apagado (tem seu valor modificado para 0).

Observando a Fig. 2 notamos que os objetos foram realmente separados dos ruídos, mas sofreram uma apreciável mudança de tamanho, o que poderia acarretar num problema para a identificação do objeto. Na verdade isso não acontece se utilizarmos o número de erosões suficiente apenas para identificar um determinado número de elementos posicionados entre limites superiores e inferiores que alinharmos objetos (Fig. 3). Utiliza-se desta maneira a característica fundamental das placas brasileiras que é ter sempre 3 letras e 4 números, isso é 7 elementos.

Um outra característica importante observada durante a erosão das imagens foi o fato desse processamento "limpar" ruídos prévios na imagem para que não haja problemas com o reconhecimento das mesmas. Pequenos objetos que contenham até 5 pixels são simplesmente apagados pelo processo de erosão. Esse recurso é importante para um bom funcionamento das etapas posteriores.

2.3 Segmentação dos Caracteres

A segmentação da imagem consiste num processo cujo objetivo é separar em imagens distintas cada caracter. Essa etapa permitirá a identificação em separado de cada letra ou número que compõem a placa. A identificação de pixels conectados faz parte de um pré-processamento

para a segmentação das imagens. Para isso foi desenvolvido um algoritmo que basicamente verifica a vizinhança de cada pixel e a partir daí vai identificando quem faz parte, ou não, do mesmo objeto, isso é está com a mesma cor que ele.

O algoritmo analisa a partir de um pixel de coordenadas (i,j) sua vizinhança que são os pixels de coordenadas $(i-1,j-1)$, $(i-1,j)$, $(i-1,j+1)$, $(i,j-1)$, $(i,j+1)$, $(i+1,j-1)$, $(i+1,j)$, $(i+1,j+1)$. Desta forma o algoritmo vai verificando cada um dos pixels. Sabemos que a imagem após a binarização possui apenas pixels pretos e brancos, representando os objetos e o fundo. Os pixels brancos são representados pelo valor 1 e os pixels pretos são representados pelo valor 0. O algoritmo de contagem, primeiramente faz uma varredura, na imagem, a procura de um pixel que represente a imagem, ou seja tenha o valor 0. Achado esse primeiro pixel o algoritmo troca o valor dele para 2 e faz uma verificação da vizinhança conforme foi descrito anteriormente. Se durante essa verificação for encontrado algum outro pixel acesso esse pixel terá seu valor trocado para 3 e os que forem sendo encontrados terão o valor 4, 5, 6 e assim por diante. Quando terminar a verificação da vizinhança o algoritmo procura o pixel com o valor 3 e faz a verificação de sua vizinhança atribuindo aos pixels que forem encontrados valores subsequentes e assim o processo continua até que o algoritmo não encontre nenhum outro pixel subsequente para verificar a vizinhança, reconhecendo assim o fim de um objeto e iniciando uma varredura na busca de um pixel acesso e reiniciando todo o processo. Desta forma conseguimos com que seja contabilizado o número de objetos conectados, ao mesmo tempo que se separa cada objeto.



Figura 3. Coordenadas limites horizontais das imagens (à esquerda) e o resultado da segmentação com cada objeto representado por uma cor diferente (à direita).

O processo de segmentação que foi desenvolvido utiliza a forma final com que o processamento de contagem disponibiliza a imagem para realizar a segmentação. Para otimizar o processo de segmentação um único algoritmo que faz a contagem e a segmentação das imagens. Esse processo foi nomeado como processo de contagem de elementos modificado. O algoritmo funciona da seguinte forma: Utilizamos o algoritmo original de contagem sendo que quando for terminada a identificação de um objeto para ser contabilizado, seja armazenado em alguma matriz o valor que foi atribuído ao último pixel desse objeto. Dessa forma conseguimos uma matriz que contenham todos os valores dos últimos pixels referente a cada objeto. Então separamos as imagens através desses valores da seguinte forma. Finalmente conseguimos que os objetos sejam devidamente segmentados e separados e imagens distintas. Dessa forma conseguimos obter imagens em condições de serem submetidas a etapa de reconhecimento de caracteres. Cada subimagem colorida mostrada na Fig. 3 vai ser submetida a última etapa do processamento, sendo composta apenas pelo objeto e pelo fundo.

2.4 Momentos Invariantes

Como o sistema desenvolvido para o reconhecimento objetiva 100% de acerto, foram usados duas formas de extração de características geométricas de cada um dos caracteres. A primeira forma utiliza diretamente o objeto separado na forma como se apresenta em cada subimagem da Fig. 3. A segunda forma considera apenas as características essenciais dos

caracteres e necessita que eles passam por um processamento prévio de afinamento ou esquematização, esse processo devido à limitações de espaço não será discutido neste trabalho.

A forma de extração de característica discutida nesta seção usa os momentos de área de Estática. Supondo que as subimagens binárias, a serem analisada e já separadas na etapa anterior, sejam consideradas individualmente e representadas por um arranjo retangular de $M \times N$ pontos, que podem assumir cores ou serem brancos (Fig. 3 à direita). Chamaremos de $B(i,j)$ o valor dos pontos da imagem. O conjunto dos pontos da imagem binária só terá valores 1 ou zero, e esses valores são definidos da seguinte forma: os *pixels* de cada uma das cores dos caracteres representado pela imagem, terá valores 1 (um). A cada ponto do fundo das subimagens branco é associado o valor 0 (zero). Às coordenadas de cada um dos $n = M \times N$ pontos de B são (i, j) . Nesta notação o momento geométrico (ou de área) do conjunto B , de ordem (p, q) , pode ser definido como:

$$m_{p,q} = \sum_{k=1}^n B(i_k, j_k) (i_k)^p (j_k)^q \quad p, q = 0, 1, 2, \dots \quad (1)$$

Segundo a teoria dos momentos de área, existe apenas um objeto B que pode produzir o mesmo valor para os momento de todas as ordens, ou em outras palavras se dois objetos têm os mesmos momentos em todos as ordens, estes objetos são idênticos. Considerando uma translação das coordenadas (i, j) de um valor (i_o, j_o) de modo que o objeto possa ser descrito em relação a um sistema de coordenadas que passa pelo centróide, definido pelas coordenadas:

$$\begin{aligned} i_o &= \sum_{k=1}^n B(i_k, j_k) i_k / \sum_{k=1}^n B(i_k, j_k) = m_{10} / m_{00} \\ j_o &= \sum_{k=1}^n B(i_k, j_k) j_k / \sum_{k=1}^n B(i_k, j_k) = m_{01} / m_{00} \end{aligned} \quad (2)$$

então pode-se definir os momentos centrais (em torno do centróide) como:

$$m_{o,p,q} = \sum_{k=1}^n B(i_k, j_k) (i_k - i_o)^p (j_k - j_o)^q \quad p, q = 0, 1, 2, \quad (3)$$

É um resultado conhecido que os momentos centrais são invariantes a translação. O momento $(m_{00} = m_{o00})$ de ordem zero, representa a área de B , e é um invariantes a translação e rotação. Sabe-se também que a soma dos momentos de área de segunda ordem em relação ao centróide, que corresponde ao momento polar de inércia $(I_x + I_y)$, é constante, independente da orientação dos eixos usados para descrever estes momentos. Assim outro invariante a rotação dos eixos é $m_{o02} + m_{o20}$. Também é conhecido que a excentricidade de uma área é um valor invariante à rotação dos eixos. A excentricidade é medida por:

$$F_2 = 4 (m_{o11})^2 + (m_{o20} - m_{o02})^2 \quad (4)$$

De uma maneira geral, a invariância em relação a rotação do objeto pode ser obtida, utilizando-se no cálculo dos momentos eixos que coincidam com os eixos principais do objeto passando pelo centróide, ao invés de um sistema de eixos (i, j) qualquer. Para esta cálculo, deve-se determinar o ângulo, θ , que identifica a orientação dos eixos principais, este ângulo é:

$$\tan 2\theta = 2 m_{o11} / (m_{o20} - m_{o02}) \quad (5)$$

Na determinação deste ângulo, deve-se usar, para garantir a unicidade dos resultados, as restrições: $m_{o20} > m_{o02}$ e $m_{o11} > 0$. Qualquer momento obtido em relação a este par de eixos é invariante a rotação. No caso de imagens digitalizadas, descritas através de $M \times N$ pixels, a utilização de eixos diferentes dos eixos verticais e horizontais originais da câmara de captura pode causar muitos erros de arredondamento. Uma forma de minimizar estes erros seria um pré-

direcionamento da câmara, antes da captura, de modo que a imagem fosse capturada paralelamente a direção dos eixos principais do objeto. Essa técnica é impossível de ser usada na aplicação pretendida. No desenvolvimento deste trabalho as imagens digitais serão processadas depois de sua captura. Neste caso a busca de outros invariantes à transformação de translação e rotação é bem mais adequada. Utilizando matrizes de rotação, qualquer ponto (i',j') , do objeto rodado de um ângulo θ , pode ser escrito em termos do sistema original de eixos (i,j) , como:

$$\begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{bmatrix} \cos\theta & \text{sen}\theta \\ -\text{sen}\theta & \cos\theta \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} = [R] \begin{bmatrix} i \\ j \end{bmatrix} \quad (6)$$

Os momentos de segunda ordem, formam um tensor de segunda ordem. Pela rotação dos eixos $[R]$ estes serão rescritos como:

$$\begin{bmatrix} m_{20}' & m_{11}' \\ m_{11}' & m_{02}' \end{bmatrix} = [R^T] \begin{bmatrix} m_{20} & m_{11} \\ m_{11} & m_{02} \end{bmatrix} [R] \quad (7)$$

O traço de um tensor de segunda ordem M_2 , denotado $tr M_2$, é o primeiro invariante escalar I_M (tem o mesmo valor numérico em qualquer sistema de coordenadas). Os outros invariantes escalares são o III_M , que é o determinante do tensor de segunda ordem, $det M_2$, e o segundo invariante ou invariante quadrático, II_M :

$$\begin{aligned} I_M &= tr M_2 = F_1 = m_{020} + m_{002} \\ II_M &= -m_{020} m_{002} + m_{011}^2 \\ III_M &= det M_2 \end{aligned} \quad (8)$$

Os momentos de terceira ordem, formam um tensor de terceira ordem. Estes, se representados em um sistema de eixos (i',j') , rotacionado de um ângulo θ , do sistema original de eixos (i,j) , podem ser escritos como:

$$\begin{aligned} m_{03}' &= m_{03} \cos^3\theta - 3 m_{12} \cos^2\theta \text{sen}\theta + 3 m_{21} \cos\theta \text{sen}^2\theta - m_{30} \text{sen}^3\theta \\ m_{12}' &= m_{12} \cos^3\theta - 2 m_{12} \text{sen}^2\theta \cos\theta + m_{03} \text{sen}\theta \cos^2\theta + m_{30} \text{sen}^2\theta \cos\theta \\ &\quad - 2 m_{21} \text{sen}\theta \cos^2\theta + m_{21} \text{sen}^3\theta \\ m_{21}' &= m_{21} \cos^3\theta - 2 m_{21} \text{sen}^2\theta \cos\theta - m_{30} \text{sen}\theta \cos^2\theta + m_{03} \text{sen}^2\theta \cos\theta \\ &\quad + 2 m_{12} \text{sen}\theta \cos^2\theta - m_{12} \text{sen}^3\theta \\ m_{30}' &= m_{30} \cos^3\theta + 3 m_{21} \cos^2\theta \text{sen}\theta + 3 m_{12} \cos\theta \text{sen}^2\theta + m_{03} \text{sen}^3\theta \end{aligned} \quad (9)$$

Os invariantes usadas neste trabalho, além dos F_1, F_2 já descritos são:

$$\begin{aligned} F_3 &= (m_{30} - 3m_{12})^2 + (3m_{21} - m_{03})^2 \\ F_4 &= (m_{30} + m_{12})^2 + (m_{21} + m_{03})^2 \\ F_5 &= (3m_{21} - m_{03})(m_{30} + m_{12})[(m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2] - \\ &\quad (m_{30} - 3m_{12})(m_{21} + m_{03})[3(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2] \end{aligned} \quad (10)$$

Considerando uma transformação de escala, de modo que $i'' = a i'$ e $j'' = a j'$ (onde a é uma constante) a área mudará de $m_{00}'' = a^2 m_{00}'$. Os momentos de ordem superiores poderiam ser modificados pelas relações abaixo:

$$(m_{00}')^{0.5(p+q)+1} \sum i''^p j''^q = (m_{00}'')^{0.5(p+q)+1} \sum i'^p j'^q \quad (11)$$

A utilização de momentos invariantes a escala é útil na identificação de objetos quando a distância de captura pode variar. Um grupo de invariantes completos (isso é a rotação, translação e mudanças de escala) podem ser obtidos usando a Eq. (11).

$$\begin{aligned}
 V(0) &= F_1 / m_{00}^2 \\
 V(1) &= F_2 / m_{00}^4 \\
 V(2) &= F_3 / m_{00}^5 \\
 V(3) &= F_4 / m_{00}^5 \\
 V(4) &= F_5 / m_{00}^{10}
 \end{aligned}
 \tag{12}$$

Nesse trabalho vamos utilizar os momentos invariantes a escala, isto é as Eq. (12) porque não queremos fazer distinção dos caracteres devido a tamanhos diferentes mas idênticos. Isso pode não ser feito se a câmera utilizada para capturar as imagens ficar em uma posição fixa em relação ao objeto, o que pode ser comum em parques de estacionamento.

3. RECONHECIMENTO E TESTES

O reconhecimento é o último processo que as subimagens são submetida depois de todos os outros. Nessa etapa são utilizados todos os elementos que foram calculados anteriormente para identificar o caracter. O processo de reconhecimento é dividido em duas etapas. A primeira etapa se constitui na formação ou adição de características dos caracteres ao banco de dados. O banco de dados é o elemento fundamental dessa etapa de processamento. É nele que ficam armazenadas todas as características referentes a cada caracter que foram previamente processadas e o nome que foi atribuído a cada um deles. O reconhecimento utiliza duas etapas para agilizar o processo. A primeira, é empregado as características relacionadas às cavidades e extremidades das subimagens esquematizadas, pois estas características são computacionalmente muito menos custosas em termos de tempo de processamento, o que faz a maior diferença neste tipo de aplicação que requer uma resposta imediata do sistema. Só no caso de ambigüidade nesta primeira forma de identificação é que são utilizadas as características relacionadas aos momentos invariantes mencionadas na seção anterior. São utilizados cinco características para efetuar o calculo referente ao reconhecimento dos caracteres se a identificação baseadas nos momentos for necessária. Essas são constituídos pelos cinco valores obtidas pelos momentos invariantes da última equação da seção anterior.

Primeiramente quando iniciamos o programa não existe nenhum banco de dados e o mesmo só será inicializado depois que realizarmos a primeira entrada de dados ao seu registro. Nessa parte do processamento é necessário que seja inserido um nome para cada caracter. Esse nome será utilizado para identifica-lo em processamentos futuros. Podemos utilizar também vários bancos de dados diferentes para vários conjuntos distintos, o que pode ser útil para aplicações de gerência de pátios de estacionamento com áreas restritas, por exemplo. Para utilizar esse recurso basta re-nomear o banco de dados que o programa esta usando e criar um novo banco de dados para esse novo conjunto de peças que vai ser submetido ao reconhecimento.

O processo de reconhecimento baseado nos momentos é bastante simples. O programa nessa etapa calcula todos os cinco elementos que calculou na primeira etapa da mesma forma e compara esses números com os dados que possui no banco de dados. Essa peça será reconhecida como a que mais se aproximar de um elemento que já exista no banco de dados. Essa comparação dos números obtidos é feita usando a seguinte avaliação de distância entre cada momento de um caracter já definido no bando e o caracter a ser identificado da seguinte forma:

$$D_i = \sum_{i=0}^4 (V(i) - V(i)_{iBD})^2$$

onde $V(i)$ é a cada uma das combinação de momento invariante de ordem i e $V(i)_{iBD}$ é a mesma combinação de cada um dos elementos i já incluídos no banco de dados. Esse cálculo indica quão próxima está a peça que se quer identificar de alguma outra já catalogada no banco de dados. O caracter peça que possuir o menor D_j será reconhecida como o que está sendo submetida ao programa. Para começar a processar as imagens pelo programa devemos capturar as imagens pelo mesmo. As imagens podem ser adquiridas via câmera e via arquivo dependendo da necessidade encontrada. O processamento via arquivo pode ser muito útil quando utilizamos o programa para montar o banco de dados. Ao final do calculo dos momentos invariantes , acontece uma variação entre os processamento para geração do banco de dados ou para a identificação dos caracteres. Se ao invés de escolhermos o reconhecimento dos caracteres, tivéssemos escolhido a opção de adicionar elemento ao banco de dados, surgirá uma tela pedindo que seja inserido um nome para essa peça. Depois de ser inserido o nome dessa peça, esse nome vai ser armazenado no banco de dados associado aos momentos invariantes previamente calculados.

Foram realizados testes com 200 imagens de placas de veículos e conseguimos um reconhecimento de 198 com o banco de dados que foi utilizado. A escolha do banco de dados é fundamental para se obter um bom resultado. Para a construção do banco de dados observamos que alguns caracteres utilizados nas placas dos veículos possuíam fonte diferente. Então para esses caracteres foram colocadas mais de uma entrada no banco de dados. Nessa seção vamos apresentar os resultados obtidos quando após construirmos o banco de dados o submetemos os seus caracteres ao reconhecimento pelo programa. A seguir vamos apresentar uma tabela que contem os dados referentes a cada caracter que foi inserido no banco de dados e um gráfico para uma melhor visualização dos dados da tabela.

Apresentaremos na Tab. 1 o subconjunto das placas testadas que precisou da análise por momentos para sua identificação. Mostra-se na colunas 2 o número de caracteres reconhecidos usando-se só as características geométricas (método não discutido neste trabalho). Na coluna 3 da Tab. 1 é mostrado o número de caracteres identificados nestas placas usando-se o momentos invariantes. Ou seja a segunda coluna da tabela representa quantos caracteres foram reconhecidos sem a necessidade de utilização dos momentos invariantes e a terceira coluna representa quantos caracteres foram reconhecidos utilizando a técnica dos momentos invariantes. As demais colunas representam o numero de reconhecimento errado que cada uma das placas obteve associado ao momentos que promoveu esse reconhecimento errôneo, para possibilitar melhorias futuras na implementação e uma análise da utilização dos mementos no reconhecimento de caracteres deste tipo de aplicação.

4. CONCLUSÃO

Observado a Tab. 1 nota-se que o momento M0, apresentou mais falhas no reconhecimento dos caracteres, seguindo do momento M1, enquanto que os momentos de ordem superiores (M3 e M4) foram os que apresentaram menor falhas. Na Tab. 1 foram processados 217 caracteres de 31 placas de automóveis, e reconhecido 112 caracteres sem a necessidade de utilização dos momentos invariantes. Sendo assim 105 caracteres foram reconhecidos utilizando a técnica dos momentos invariantes. Foram realizados 505 cálculos de momentos onde apenas 49 apresentaram erro de reconhecimento. Esses erros não prejudicaram o reconhecimento como um todo pois 100% das placas processadas foram reconhecidas com sucesso (utilizando a combinação de técnicas implementadas).

Apresentamos no gráfico da Fig. 4 detalhes de caracteres que foram reconhecidos com erro e suas relações com os momentos que possibilitaram este erro. Observamos que nos números o maior problema de reconhecimento foi em confundir o “7” (sete) com o “3” (três). Os momentos de primeira e segunda ordem apresentaram grande numero de erros quando ao reconhecimento desses caracteres. Já os momentos de ordem superior não apresentaram nenhum erro para o

reconhecimento desses caracteres. Dessa forma o reconhecimento final do caracter foi correto mas esses momentos para esses caracteres não apresentaram resultados adequados.

Nas letras, os caracteres “H” e “N” bem como “D” e “O” foram os mais problemáticos. Os caracteres “H” e “N” apresentaram problemas de reconhecimento em todos os momentos invariantes exceto o último, V(4). Já os caracteres “D” e “O” apresentaram problemas de reconhecimento apenas nos dois primeiros momentos de mais baixa ordem, V(0) e V(1).

Mas sempre, a pesar de alguns momentos invariantes apresentarem problemas no reconhecimento de alguns caracteres os outros momentos foram capazes de reconhecer com precisão o mesmo. Esse fato assegurou um reconhecimento de 100% dos caracteres. Isso mostra que essa técnica, aqui utilizada para reconhecer placas de automóveis pode ter muitas outras aplicações de reconhecimento de caracteres sem qualquer intervenção humana.

Tabela 1. Conjunto de Placas Reconhecidas Corretamente apenas com o auxílio adicional dos de momentos invariantes

Placa Capturada	Carac. Geom	Momentos	M0	M1	M2	M3	M4	Placa Identificada
GWK2323	4	3	1	0	0	0	0	GWK2323
KMG9066	4	3	0	0	0	0	0	KMG9066
KMH9737	2	5	1	1	0	0	0	KMH9737
KML4188	5	2	0	0	0	0	0	KML4188
KNA2764	4	3	0	1	0	1	0	KNA2764
KNA4521	6	1	0	0	0	0	0	KNA4521
KOC8204	5	2	1	1	0	0	0	KOC8204
KQM1513	4	3	1	1	0	0	0	KQM1513
KRE4315	5	2	0	1	0	0	0	KRE4315
KRJ2309	4	3	1	0	0	0	0	KRJ2309
KRE5912	4	3	1	0	0	0	0	KRE5912
KMJ0305	4	3	1	1	0	0	0	KMJ0305
KMM2090	6	1	0	0	0	0	0	KMM2090
KNA7675	5	2	1	0	0	0	0	KNA7675
KOD7728	2	5	2	1	0	0	0	KOD7728
KRJ-7761	2	5	2	0	0	0	0	KRJ7761
KSY4558	4	3	0	0	0	0	0	KSY4558
KUB6657	2	5	1	0	0	0	1	KUB6657
LAB7582	4	3	0	0	0	0	0	LAB7582
LAE0231	5	2	0	1	0	0	0	LAE0231
LBE9571	5	2	1	0	0	0	0	LBE9571
LBM7321	4	3	1	0	0	0	0	LBM7321
LCC-4261	3	4	2	0	0	0	0	LCC4261
LCU3687	2	5	2	1	0	0	0	LCU3687
LIY-3173	1	6	1	2	2	0	0	LIY3173
LJF2087	4	3	2	0	0	0	0	LJF2087
LNC3270	4	3	1	1	1	0	0	LNC3270
LNI7263	1	6	1	2	1	0	0	LNI7263
LNJ7519	2	5	1	0	0	0	0	LNJ7519
LNT0251	3	4	1	0	0	0	0	LNT0251
LNU4768	2	5	2	1	0	0	0	LNU4768

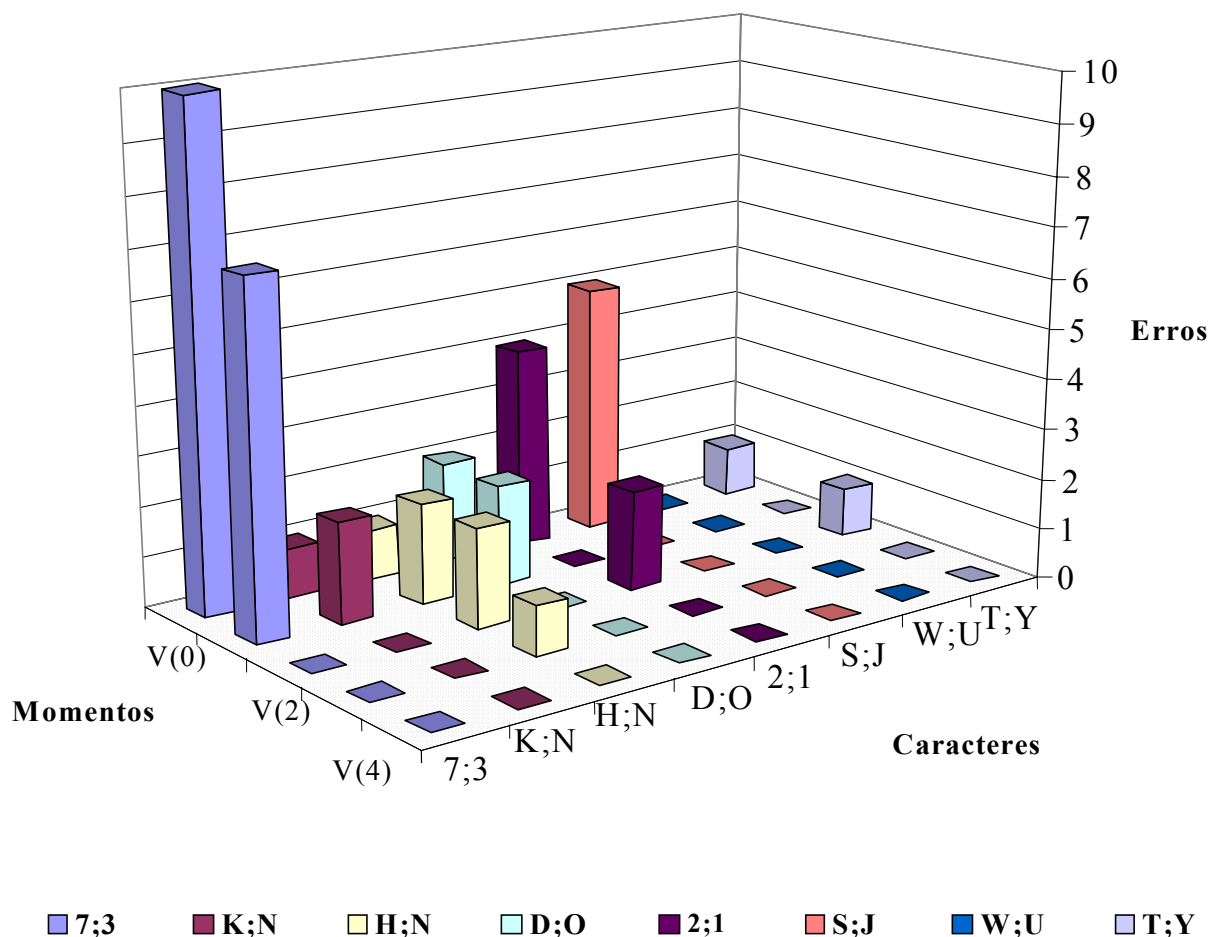


Figura 4. Análise dos erros no reconhecimento por momentos

5. AGRADECIMENTOS

Ao Laboratório de Computação Visual do CTC da UFF, (<http://visual.ic.uff.br>) que proporcionou os recursos computacionais para execução deste trabalho.

6. REFERÊNCIAS

- R. C. Gonzalez, P. Wintz, *Digital Image Processing*, Addison-Wesley, 1987.
- K-A Moon, S.-Y. Chi, J.-W Park, W.-G. Oh, “ Multilingual Character Segmentation Using Matching Rate”, *Proceeding IWISP’96*, November 1996, Manchester, Elsevier Science , pp. 295- 298.
- L. E. Malvern, *Introduction to the Mechanics of a Continuous Medium*, Prentice-Hall, Inc. 1969.
- M. Gruber, K.-Y. Hsu, “Moment Based Image Normalization with High Noise Tolerance”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19 , No. 2, February, 1997, pp. 136-138.
- T. B. Nguyen, B. J. Oommen, “Moment-Preserving Piecewise Linear Approximation of Signal and Images” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19 , No. 1, January, 1997, pp. 84-91.
- K. Voss , H. Suesse, “Invariant Fitting of Planar Objects by Primitives”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19 , No. 1, January, 1997, pp. 80-84.
- MONTEIRO L.H., Utilização de Técnicas de Processamento de Imagens para o

Reconhecimento de Placas de Veículos", por Leonardo Hiss Monteiro, departamento de Ciência da Computação IC/ UFF - defesa em 22/08/2002.

CONCI, A., MONTEIRO, L.H, "Utilização de momentos na identificação automática de elementos variados adquiridos com ruídos ", vol. 3 dos anais do IV Congresso Iberoamericano de Ingeniería Mecánica, CIDIM '99, Santiago do Chile, 23 a 26 de Novembro de 1999 Universidad de Santiago de Chile , ISBN 956-7069-41-7 e 959-7069-44-1, p.100-104.

5. DIREITOS AUTORAIS

Os autores são os únicos responsáveis pelo conteúdo do material incluído no seu trabalho.

BRAZILIANS VEHICLE PLATE RECOGNITION BY IMAGE

Aura Conci

PGMEC/UFF - Passo da Pátria 156 Niterói, RJ Brazil - 24210 -240 aconci@ic.uff.br

Leonardo Hiss Monteiro

PGCC/UFF - Passo da Pátria 156 Niterói, RJ Brazil - 24210 -240 aconci@ic.uff.br

Abstract: *Nowadays we can see that the usability of the computer vision is increasing very fast, due to the decrease of digital camera cost. With the fast development in this area it is possible to use image processing techniques in reasonable commons areas. The main objective of this work is show that pattern recognition can be used to solve the practical problem of recognize car licenses. Electronic tickets for parking fee or velocity control is frequently used, but in many cases these type of mechanisms do not use any system to recognize automatically the car license. In this work we show that the usage of some image processing and optical character recognition techniques make it possible to get correct results on automatic identification of vehicle plates.*

Keywords: *Image Processing, Pattern Recognition, Optical Character Recognition - OCR.*