

RECONHECIMENTO DE PLACAS DE VEÍCULOS UTILIZANDO PROCESSAMENTO DE IMAGEM

*Leonardo Hiss Monteiro¹
Aura Conci²*

Resumo: O relativo baixo custo das câmeras de vídeo, e o desenvolvimento tecnológico tornam, atualmente, viáveis utilizações de reconhecimento de padrões aplicadas à problemas do cotidiano como o reconhecimento de placas de veículos. Os dispositivos eletrônicos para aplicar multas nos veículos no estado do Rio de Janeiro e na maioria dos outros 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-se como é possível obter 100% de resultados corretos na identificação automática das placas de veículos, depois de sua imagem ser capturada digitalmente.

Palavras-chave: Processamento de imagens, Reconhecimento de padrões, Reconhecimento ótico de caracteres (OCR)

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 applications. The main objective of this work is show that pattern recognition can be used to solve the practical problem of car license identification. Electronic tickets for automobiles parking fee or vehicles velocity control is frequently used, but in many cases these type of mechanisms do not use any system to recognize automatically the car official registration. In this work we show that the usage of some image processing and the pattern recognition techniques make it possible to get results 100% corrects in license plate identification.

Key word: Image processing, Pattern recognition, Optical Character Recognition (OCR)

¹ Nokia S.A.

² Universidade Federal Fluminense

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 às máquinas, 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 computadorizadas 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 ou expressões, na 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, auxiliando assim o homem em tarefas muitas vezes extremamente trabalhosas e repetitivas, ou que seriam impossível de existir sem o computador como as provenientes de sonares, radares, ultrassom, ultravioleta, infravermelho, sensores térmicos, raios x, etc.

No desenvolvimento do sistema de reconhecimento de placas utilizamos várias técnicas, que serão discutidas e

testadas nas próximas seções, para realizar o processamento das cenas capturadas. Essas vão desde técnicas para a separação dos elementos da imagem até métricas desenvolvidas para estabelecer de qual padrão está mais próximo o caracter que está sendo analisado. Este sistema poderia também ser utilizado para o reconhecimento de elementos alfanuméricos em imagens. O que seria útil, por exemplo, no reconhecimento de embalagens, ou de placas de sinalização.

Na forma atual de identificação de placas de veículos, o sistema aqui apresentado e esquematizado na figura 1, pode ser utilizado para ampliar as possibilidades das câmeras de fiscalização de trânsito espalhadas pelas ruas das cidades, para agilizar e intensificar o controle em estacionamentos, ou até pelos DETRANS para conseguir-se mais rapidez na distribuição das multas eletrônicas em veículos infratores. Maior agilidade na arrecadação se reverteria em benefícios ao usuário e ao contribuinte, possibilitando uma redução nas taxas pela redução dos custos.

2. TÉCNICAS UTILIZADAS NO RECONHECIMENTO

Diversos métodos que combinam aspectos de computação gráfica, processamento de imagem e automação são empregados na implementação deste sistema de visão computacional, que em termos funcionais apresenta basicamente **quatro etapas**. Na primeira etapa a imagem é **adquirida**. Na segunda, a região da placa é **processada**. Na terceira, características são extraídas da placa em análise, que ao serem **classificadas** resultam no seu reconhecimento entre os dados armazenados em um banco de dados (na quarta e última etapa). A seguir apresentaremos as técnicas que foram utilizadas do processamento à classificação das imagens:

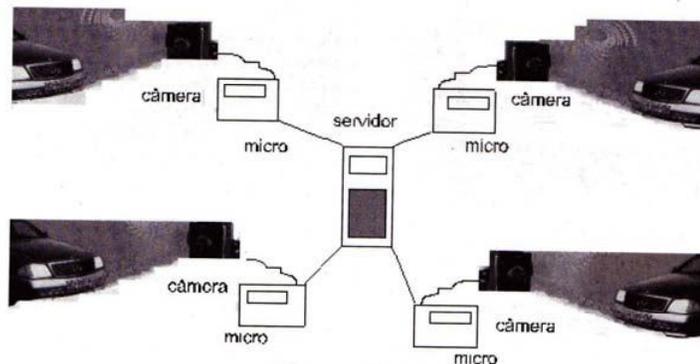


Fig. 1- Sistemas de captura e análises aplicado a Identificação de Placas

2.1 Binarização

Para fazer a separação entre o fundo da imagem e os elementos que desejamos analisar utilizamos a binarização ou *threshold*. A binarização é o método mais simples de segmentação de imagens. Resumidamente consiste em separar uma imagem, originalmente colorida ou em diversos níveis de cinza, em pixels pretos e brancos. A decisão da nova cor que terá o pixel é realizada de acordo com o ponto de corte ou *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 que uma imagem digitalizada pode ser escrita como uma função $f(x,y)$, a resposta da binarização é [1]:

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

Onde R_1 e R_2 são os valores estipulados para os níveis de cinza da imagem binarizada, no caso utiliza-se 0 (preto) e 255 (branco). O valor de T (ou valor de *threshold*) deve ser escolhido com o cuidado para que não capture elementos que não façam parte dos caracteres. O *threshold* escolhido em imagens coloridas

pode estar associado a média dos tons de cada canal de cor.

Uma ferramenta auxiliar na escolha deste ponto de corte (ou valor de *threshold*) é o histograma da imagem (figura 2). 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 por um objeto e um fundo podemos estabelecer o ponto de corte através de um isolamento das regiões que representem o fundo e o objeto. Para imagem compostas por mais de um objeto de cores diferentes sob um fundo, podemos utilizar uma técnica de binarização multi-nível (*multilevel thresholding*).

Quando não é possível estabelecer as regiões correspondentes a cada objeto no histograma se torna mais difícil encontrar o ponto de corte ideal. Existem métodos automáticos para a escolha e ajuste do ponto de corte que usam o histograma da imagem ou a imagem propriamente dita como referência. Métodos que comparam conhecimentos prévios sobre parâmetros da imagem e obtêm vários pontos de corte são também conhecidos.



Fig 2 - Histograma bi-modal onde fica simples a separação entre o fundo e o objeto

2.2 Erosão

Erosão foi a técnica utilizada para resolver um problema que apareceu freqüente nesta aplicação real de processamento de imagens. Imaginem que durante a captura da placa os caracteres que desejamos reconhecer pareçam estar em contato um com o outro de forma que pareçam um único objeto. Nesse caso temos sérios problemas para reconhecê-los pois o algoritmo provavelmente tratará esses dois objetos como um só. Fica evidente então a necessidade de desenvolver algo que seja capaz de separar objetos que possivelmente venham estar em contato entre si, devido a um maior sombreamento ou pelo movimento do carro [17]. A solução adotada para esse tipo de problema foi a utilização de um recurso conhecido como **erosão** (figuras 3 e 4). 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 escolhe-se o elemento que vai ser utilizado para realizar a "erosão da imagem" (figura 3). A escolha do elemento que vai fazer a operação é muito importante pois esse 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 garantia na imagem original suas características vitais. Apresentamos na figura 3 o elemento que foi utilizado [17].

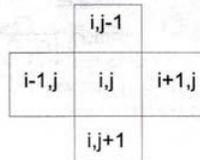


Fig. 3 - Elemento utilizado para fazer a erosão

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 (isto é com o valor 1) o pixel que está sendo analisado continua aceso. Caso contrário o pixel é apagado (tem seu valor modificado para 0). Apresentamos na figura 4 uma imagem genérica que foi submetida a 20 erosões utilizando o elemento apresentado acima.

Observando a figura 4 nota-se que os objetos foram realmente separados, mas sofreram uma apreciável mudança de tamanho, o que poderia acarretar num problema para a identificação do objeto. Na verdade isso não acontece se for utilizado um número de erosões pequeno e apenas o suficiente para identificar um determinado número de elementos, utilizando desta maneira a característica fundamental das placas brasileiras atualmente em uso que é ter sempre 3 letras e 4 números, isso é 7 elementos.

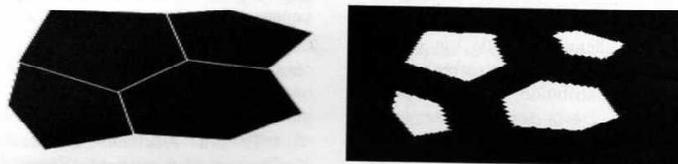


Fig. 4 - Imagem antes e depois da erosão

Uma outra característica muito 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 de vital importância para um bom funcionamento do programa.

2.3 Segmentação da imagem

A segmentação da imagem consiste num processo cujo objetivo é separar em imagens distintas objetos que não se encontram conectados. Essa etapa é de importância primordial pois permitirá a identificação em separado de cada letra ou número que compõem a placa. A contagem de objetos conectados faz parte de um pré-processamento para a segmentação das imagens. Essa parte do processamento é fundamental para o reconhecimento de cada caracter. Para isso foi desenvolvido um algoritmo que basicamente verifica a vizinhança de cada pixel (figura 5) 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 $(i-1,j-1)$, $(i-1,j)$, $(i-1,j+1)$, $(i+1,j-1)$, $(i+1,j)$, $(i+1,j+1)$, $(i,j-1)$, $(i,j+1)$. Desta forma o algoritmo vai verificando cada um dos pixels. Sabemos que a imagem após a binarização possui apenas pixels brancos e pretos, representando os objetos e o fundo. Os brancos são representados pelo valor 1 e os 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 1. Achado esse 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 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 pega 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 não se encontre nenhum outro pixel subsequente para verificar na vizinhança, então o algoritmo reconhece o fim de um objeto e reinicia uma varredura na busca de um pixel acesso e reinicia todo o processo. Desta forma conseguimos com que seja contabilizado o número de objetos conectados.

| | | |
|-----------|---------|-----------|
| $i-1,j-1$ | $i,j-1$ | $i+1,j-1$ |
| $i-1,j$ | i,j | $i+1,j$ |
| $i-1,j+1$ | $i,j+1$ | $i+1,j+1$ |

Fig. 5 - Coordenadas das imagens

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 desenvolveu-se um único algoritmo que faz a contagem e a segmentação das imagens. Esse processo

funciona utilizado o algoritmo original de contagem sendo que quando for terminada a identificação de um objeto para ser contabilizado, se armazena o valor que foi atribuído ao último pixel desse objeto. Dessa forma consegue-se conhecer todos os valores dos últimos pixels referente a cada objeto. Então separa-se as imagens através desses valores. Depois de conseguir-se que os objetos sejam devidamente segmentados e separados em imagens distintas, pode-se submetê-las ao processo de extração de características para o reconhecimento.

3. MOMENTOS INVARIANTES

Esta forma de reconhecimento usa fundamentos de Estática. Supondo que a imagem binária, a ser analisada, é representada por um arranjo retangular de $M \times N$ pontos, que podem assumir as cores *preta* ou *branca*. Chamaremos de $B(i,j)$ o valor dos pontos da imagem que seja uma silhueta da forma do objeto representado pela imagem e tenha valores 1 (um) ($B(i,j)$ é um subconjunto de $g(x,y)$ da seção 2.1). As 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, existe apenas um objeto B que pode produzir o mesmo valor para os momentos 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_0, j_0) de modo que o objeto possa ser descrito em relação a um sistema de coordenadas que passa pelo centróide, definido pelas coordenadas:

$$\begin{aligned} i_0 &= \sum_{k=1}^n B(i_k, j_k) i_k / \sum_{k=1}^n B(i_k, j_k) = m_{10} / m_{00} \\ j_0 &= \sum_{k=1}^n B(i_k, j_k) j_k / \sum_{k=1}^n B(i_k, j_k) = m_{01} / m_{00} \end{aligned} \quad (2)$$

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

$$m_{p,q} = \sum_{k=1}^n B(i_k, j_k) (i_k - i_0)^p (j_k - j_0)^q \quad p, q = 0, 1, 2, \dots \quad (3)$$

É um resultado conhecido da Estática que os momentos centrais são invariantes a translação. O momento $(m_{00} = m_{000})$ 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 independente da orientação dos eixos usados para descrever estes momentos. Assim outro invariante a rotação dos eixos é $m_{02} + m_{20}$.

Também é conhecido que a excentricidade de uma área é um valor invariante à rotação dos eixos. A excentricidade é medida por:

$$F_2 = 4 (m_{011})^2 + (m_{20} - m_{02})^2 \quad (4)$$

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 este cálculo, deve-se determinar o ângulo, θ , que identifica a orientação dos eixos principais, este ângulo é:

$$\tan 2\theta = 2 m_{011} / (m_{20} - m_{02}) \quad (5)$$

Na determinação do ângulo em (5), deve-se usar, para garantir a unicidade dos resultados, as restrições: $m_{20} > m_{02}$ e $m_{011} > 0$. 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 deste trabalho. 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:

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

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 II_M , que é o determinante do tensor de segunda ordem, $det M_2$, e o segundo invariante ou invariante quadrático, III_M [10]:

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

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 , em relação ao sistema original de eixos (i, j) , podem ser escritos como:

$$m_{03}' = m_{03} \cos^3\theta - 3 m_{12} \cos^2\theta \sin\theta + 3 m_{21} \cos\theta \sin^2\theta - m_{30} \sin^3\theta$$

$$m_{12}' = m_{12} \cos^3\theta - 2 m_{12} \sin^2\theta \cos\theta + m_{03} \sin\theta \cos^2\theta + m_{30} \sin^2\theta \cos\theta - 2 m_{21} \sin\theta \cos^2\theta + m_{21} \sin^3\theta \quad (9)$$

$$m_{21}' = m_{21} \cos^3\theta - 2 m_{21} \sin^2\theta \cos\theta - m_{30} \sin\theta \cos^2\theta + m_{03} \sin^2\theta \cos\theta + 2 m_{12} \sin\theta \cos^2\theta - m_{12} \sin^3\theta$$

$$m_{30}' = m_{30} \cos^3\theta + 3 m_{21} \cos^2\theta \sin\theta + 3 m_{12} \cos\theta \sin^2\theta + m_{03} \sin^3\theta$$

Os tensores de terceira ordem também apresentam diversos invariantes. Hu [11] utiliza a combinação:

$$f_1 = m_{20} + m_{02}$$

$$f_2 = (m_{20} - m_{02})^2 + 4m_{11}^2$$

$$f_3 = (m_{30} - 3m_{12})^2 + (3m_{21} - m_{03})^2$$

$$f_4 = (m_{30} + m_{12})^2 + (m_{21} + m_{03})^2$$

$$f_5 = (m_{30} - 3m_{12})(m_{30} - m_{12}) [(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2] + (3m_{21} - m_{03})(m_{21} + m_{03}) [3(m_{30} + m_{12}) - (m_{21} + m_{03})]$$

$$f_6 = (m_{20} - m_{02}) [(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2] + 4m_{11}(m_{30} + m_{12})(m_{21} + m_{03})$$

$$f_7 = (3m_{21} - m_{03})(m_{30} + m_{12}) [(m_{30} + m_{12})^2 - 3(m_{21} + m_{03})] - (m_{30} - 3m_{12})(m_{21} + m_{03}) [3(m_{30} + m_{12}) - (m_{21} + m_{03})]$$

Liu e Tsai [7] utilizam m_{00} , f_1 e f_2 . Banieh e de Figueiredo [12] usam $j_1 = III_M$ e:

$$j_2 = (m_{30} m_{03} - m_{21} m_{12})^2 - 4 (m_{03} m_{12} - m_{21}^2) (m_{21} m_{30} - m_{12}^2)$$

$$j_3 = m_{40} m_{04} - 4 m_{31} m_{13} + 3 m_{22}^2$$

$$j_4 = m_{40} m_{22} m_{04} - 2 m_{31} m_{22} m_{13} - m_{40} m_{13}^2 - m_{04} m_{31}^2 m_{22}^3$$

Neste trabalho, além dos F_1 , F_2 já descritos são usados os invariantes:

$$F_3 = (m_{30} - 3m_{12})^2 + (3m_{21} - m_{03})^2$$

$$F_4 = (m_{30} + m_{12})^2 + (m_{21} + m_{03})^2$$

$$F_5 = (3m_{21} \cdot m_{03})(m_{30} + m_{12}) [(m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2] - (m_{30} - 3m_{12})(m_{21} + m_{03}) [3(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2] \quad (10)$$

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 podem ser modificados pelas relações:

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

A utilização de momentos invariantes a escala é útil na identificação de objetos quando a distância de captura pode variar, como no caso deste trabalho. Assim um grupo de invariantes completos pode ser obtido usando as equações (11).

$$\begin{aligned} M0 &= F_1 / m_{00}^2 \\ M1 &= F_2 / m_{00}^4 \\ M2 &= F_3 / m_{00}^5 \\ M3 &= F_4 / m_{00}^5 \\ M4 &= F_5 / m_{00}^{10} \end{aligned} \quad (12)$$

Na identificação dos caracteres que compõem as placas utilizamos além dos momentos invariantes a escala, as características intuitivas relacionadas aos de números de "buracos", números de extremidades e posição destas extremidades em relação a um sistema de coordenadas locais.

4. RECONHECIMENTO

O reconhecimento é o último processo que a imagem é submetida. Nessa etapa são utilizados todos os elementos que foram calculados anteriormente para identificar o objeto. O processo de reconhecimento das imagens é dividido em duas etapas. A primeira etapa é constituída na formação ou adição de elementos ao banco de dados. O banco de dados é o elemento fundamental dessa etapa de processamento. É nele que ficam

ENGEVISTA, v. 5, n. 10, p. 31-43, dez. 2003

armazenados todos os elementos referentes aos caracteres das imagens que foram previamente processadas e o nome que foi atribuído a cada um deles. São utilizados cinco elementos para efetuar o cálculo referente ao reconhecimento das imagens por momentos invariantes. Esses elementos são constituídos pela área do objeto e mais quatro momentos invariantes.

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 registro. Esse nome será utilizado para identificar o número ou letra em processamentos futuros. Podemos utilizar também vários bancos de dados diferentes para vários conjuntos de fontes distintas, ou registros dos principais clientes de um estacionamento, por exemplo. Para utilizar esse artifício basta re-nomear o banco de dados que o programa esta usando e criar um novo banco de dados para esse novo conjunto que vai ser usado no reconhecimento.

O processo de reconhecimento é mais o simples. O programa nessa etapa calcula para cada elemento a ser identificado todos os dados que utiliza na primeira etapa da mesma forma e compara esses números com os dados que possui no banco de dados. Esse elemento será reconhecido como o que mais se aproximar de um elemento que já exista no banco de dados. Considerando a comparação dos números obtidos no caso dos momentos invariante, se M_i for o momento invariante de ordem i e $M_{i,db}$ o momento invariante de ordem i no banco de dados, é realizado o da porcentagem de quão próxima está a letra (ou numero) que se quer identificar de alguma outra já catalogada no banco de dados. Após realizarmos esse cálculo para todas as peças e todos os momentos devemos contabilizar os dados através de:

$$P_j = \sum_i^N D_{ij} M_i$$

onde N é o número de momentos que estão sendo utilizados. Após esse cálculo vamos possuir vários P_j cada um referente a uma peça que existe no banco de dados. A peça que possuir o maior P_j será reconhecida como a peça que está sendo submetida ao programa.

Para começar a processar as imagens pelo programa devemos introduzi-las digitalmente no mesmo. As imagens podem ser adquiridas diretamente de uma câmera digital ou via arquivo dependendo da necessidade encontrada. O processamento via arquivo pode ser útil quando o programa for

utilizado em um ambiente de rede rodando em um servidor. Nessa situação o programa pode ficar analisando as imagens que recebe através de um ou mais postos de observação conectados a câmeras e dessa forma fazer uma rede de processamento de imagens. O processamento do programa é dividido em duas partes, sendo a primeira utilizada para realizar a montagem do banco de dados e a segunda para iniciar o reconhecimento das peças. Para iniciar cada um desses processos devemos clicar no menu **Processa**, como mostrado na figura 6.

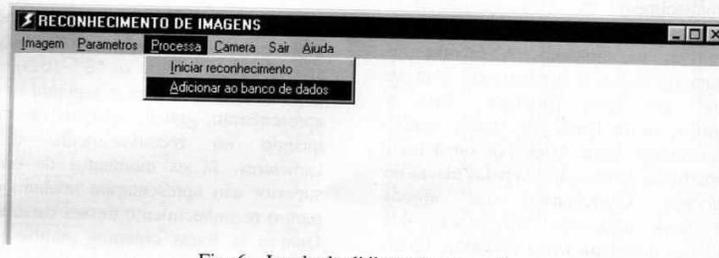


Fig. 6 - Janela de diálogo do menu **Processa**

O processo que a imagem é submetida é igual em ambas opções desse menu: **iniciar reconhecimento** (que inicia o processo de reconhecimento da imagem) e **adicionar ao banco de dados** (que adiciona esse elemento ao banco de dados), apenas diferindo na parte final. O processo consiste primeiramente na binarização da imagem através do *threshold*, depois a imagem é submetida a erosão quantas vezes forem especificados para chegara 7 elementos. Após a erosão é iniciado o processo de contagem de elementos conectados. Ao fim dessa etapa é colocada uma janela na tela com o número ou as letras identificadas, que fica aguardando um sinal de OK para que prossiga na de análise ou se coloque mais um elemento no banco de dados. Também pode ser apresentado as características calculadas de cada um deles, ou o processamento de cada etapa passo a passo, para análise ou futura modificação de algumas das etapas do sistema. Se, ao invés de escolhermos o

ENGEVISTA, v. 5, n. 10, p. 31-43, dez. 2003

reconhecimento do objeto, tivéssemos escolhido a opção de adicionar elemento ao banco de dados, surgirá uma tela pedindo que seja inserido um nome para esse caracter da imagem. Esse nome vai ser armazenado no banco de dados associado aos elementos previamente calculados daquele caracter. O banco de dados é composto por um arquivo texto que pode ser editado por qualquer editor de texto atual. Se só forem utilizados os momentos invariantes é composto por seis colunas; a primeira coluna indica os nomes das peças, a segunda coluna é composta pela área da peça, a terceira pelo primeiro momento invariante, a quarta pelo segundo invariante e assim por diante até a última coluna. Então, se desejarmos re-nomear o elemento, ou retirá-lo do banco de dados, basta editar o arquivo texto e apagar a linha referente ao que se deseja excluir do registro do banco de dados. O nome do arquivo que contém o banco de dados fica armazenado no mesmo diretório do

programa . Para se construir outro banco de dados podemos re-nomear esse arquivo e criar um novo banco através do programa. Dessa forma pode-se atuar com vários bancos de dados , de acordo com a necessidade encontrada.

5. CONCLUSÕES

Os teste foram realizados utilizando imagens capturadas por uma câmara digital coletadas de forma aleatória durante o dia e durante a noite. Utilizamos nos teste 31 imagens de placas de veículos e conseguimos um reconhecimento de 100%, quando alem dos momentos usamos as formas intuitivas mencionadas. A escolha do banco de dados é fundamental para se obter um bom resultado. Para a construção do banco de dados usados inicialmente fonte Arial, por ser a mais semelhante da fonte usada pelas placas de veículos. Observamos que alguns caracteres utilizados nas placas dos veículos possuíam fonte diferente. Então para esses caracteres foi colocada mais de uma entrada no banco de dados.

Apresentaremos na tabela 1 os valores obtidos no processamento de cada uma das placas testadas utilizando o banco de dados da seção anterior para realizar o reconhecimento dos caracteres. A segunda coluna da tabela representa quantos caracteres foram reconhecidos sem a necessidade de utilização dos momentos invariantes (usando apenas os dados intuitivos mencionados no final da seção 3). A terceira coluna representa quantos caracteres foram reconhecidos utilizando a técnica dos momentos invariantes. As demais colunas representam o número de reconhecimentos errados que cada uma das placas obteve separado por momentos.

Os momentos de ordem inferiores apresentaram mais falhas no reconhecimento dos caracteres do que os momentos de ordem superior. Foram processadas ao todo 217 caracteres de 31 placas de automóveis. Foram reconhecidos 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. Apresentamos nas figuras 7 e 8 um gráfico com as estatísticas dos erros obtidos usando cada um dos momentos invariantes e os pares de caracteres que foram reconhecidos com erro, respectivamente.

Quanto aos números o maior problema de reconhecimento foi quanto ao "7" (sete) com o "3" (três). Os momentos de primeira e segunda ordem apresentaram grande número de erros quando ao reconhecimento desses caracteres. Já os momentos de ordem superior não apresentaram nenhum erro para o reconhecimento desses caracteres. Quanto às letras obtemos muitos erros nos caracteres "H" e "N" bem como "D" e "O". Os caracteres "H" e "N" apresentaram problemas de reconhecimento em todos os momentos invariantes exceto o ultimo. Já os caracteres "D" e "O" apresentaram problemas reconhecimento apenas nos dois primeiros momentos de mais baixa ordem.

Apesar de alguns momentos invariantes apresentarem problemas no reconhecimento de alguns caracteres os outros momentos invariantes eram capazes de reconhecer com precisão o mesmo. Esse fato assegurou um reconhecimento de 100% dos caracteres. É claro que se for feito um estudo detalhado do banco de dados os problemas encontrados por alguns momentos no reconhecimento de caracteres pode ser muito minimizado.

As técnicas que foram utilizadas para reconhecer placas de automóveis podem ser utilizadas para inúmeras outras aplicações de reconhecimento de caracteres. Com a implementação destas técnicas se torna viável um sistema

automático de reconhecimento de placas humana.
de veículos sem qualquer intervenção

Tabela 1 - Placas processadas

| Placa | Carac. Geom | Momentos | M0 | M1 | M2 | M3 | M4 |
|---------|-------------|----------|----|----|----|----|----|
| GWK2323 | 4 | 3 | 1 | 0 | 0 | 0 | 0 |
| KMG9066 | 4 | 3 | 0 | 0 | 0 | 0 | 0 |
| KMH9737 | 2 | 5 | 1 | 1 | 0 | 0 | 0 |
| KML4188 | 5 | 2 | 0 | 0 | 0 | 0 | 0 |
| KNA2764 | 4 | 3 | 0 | 1 | 0 | 1 | 0 |
| KNA4521 | 6 | 1 | 0 | 0 | 0 | 0 | 0 |
| KOC8204 | 5 | 2 | 1 | 1 | 0 | 0 | 0 |
| KQM1513 | 4 | 3 | 1 | 1 | 0 | 0 | 0 |
| KRE4315 | 5 | 2 | 0 | 1 | 0 | 0 | 0 |
| KRJ2309 | 4 | 3 | 1 | 0 | 0 | 0 | 0 |
| KRE5912 | 4 | 3 | 1 | 0 | 0 | 0 | 0 |
| KMJ0305 | 4 | 3 | 1 | 1 | 0 | 0 | 0 |
| KMM2090 | 6 | 1 | 0 | 0 | 0 | 0 | 0 |
| KNA7675 | 5 | 2 | 1 | 0 | 0 | 0 | 0 |
| KOD7728 | 2 | 5 | 2 | 1 | 0 | 0 | 0 |
| KRJ7761 | 2 | 5 | 2 | 0 | 0 | 0 | 0 |
| KSY4558 | 4 | 3 | 0 | 0 | 0 | 0 | 0 |
| KUB6657 | 2 | 5 | 1 | 0 | 0 | 0 | 1 |
| LAB7582 | 4 | 3 | 0 | 0 | 0 | 0 | 0 |
| LAE0231 | 5 | 2 | 0 | 1 | 0 | 0 | 0 |
| LBE9571 | 5 | 2 | 1 | 0 | 0 | 0 | 0 |
| LBM7321 | 4 | 3 | 1 | 0 | 0 | 0 | 0 |
| LCC4261 | 3 | 4 | 2 | 0 | 0 | 0 | 0 |
| LCU3687 | 2 | 5 | 2 | 1 | 0 | 0 | 0 |
| LIY3173 | 1 | 6 | 1 | 2 | 2 | 0 | 0 |
| LJF2087 | 4 | 3 | 2 | 0 | 0 | 0 | 0 |
| LNC3270 | 4 | 3 | 1 | 1 | 1 | 0 | 0 |
| LN17263 | 1 | 6 | 1 | 2 | 1 | 0 | 0 |
| LNJ7519 | 2 | 5 | 1 | 0 | 0 | 0 | 0 |
| LNT0251 | 3 | 4 | 1 | 0 | 0 | 0 | 0 |
| LNU4768 | 2 | 5 | 2 | 1 | 0 | 0 | 0 |

6. AGRADECIMENTOS

Ao Laboratório de Computação Visual do CTC da UFF, (<http://visual.ic.uff.br>) que proporcionou os recursos computacionais para execução deste trabalho. E aos revisores técnicos pelas sugestões que muito colaboraram para a melhoria deste trabalho

7. REFERÊNCIAS

[1] A. Watt, F. Policarpo - The Computer Image , Addison-Wesley Pub Co (Net); 1998.

[2] S-KChang, *Principles of Pictorial Information Systems Design*, Prentice-Hall, 1989.

[3] 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.

[4] C-H Teh , R. T. Chin, "On Image Analysis by Methods of Moments", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10 , No. 4, July, 1988, pp. 495-513.

[5] F. W. Smith , M. H. Wright, "Automatic ship photo interpretation by

- method of moments", *IEEE Transactions on Computer*, vol. C-20, pp. 1089-1094, September, 1971.
- [6] S. A. Dudani, K. J. Breeding, R. B. McGhee, "Aircraft identification by moment invariants", *IEEE Transactions on Computer*, vol. C-26, pp. 39-26, January, 1977.
- [7] C.-H. Liu, W.-H. Tsai, "3D Curved Object Recognition from Multiple 2D Camera Views", in *Advances in Computer*, Academic Press, 1990, pp. 177-187.
- [8] Y. S. Abu-Mostfa, D. Psaltis, "Recognitive aspects of moment invariants", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI -6, No. 7, November, 1984, pp. 698-706.
- [9] M.-K. Hu, "Pattern Recognition by Moment Invariant", *Proceedings of the IRE*, vol. 49, N. 9, September, 1961, p. 1428.
- [10] L. E. Malvern, *Introduction to the Mechanics of a Continuous Medium*, Prentice-Hall, Inc. 1969.
- [11] M.-K. Hu, "Visual Pattern Recognition by Moment Invariant", *IRE Transactions on Information Theory*, vol. 49, N. 9, September, 1961, p. 1428.
- [12] B. Bamieh, R. J. P. de Figueiredo, "A general moment-invariant/attributed graph method for three-dimensional object recognition from a single image.", *IEEE Journal of Robotics and Automation*, vol. RA-2, March, 1986, pp. 31-41.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] Persoon, K.-S. Fu, "Shape Discrimination Using Fourier Descriptors", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI -8, No. 3, May, 1986, pp. 388-397.
- [17] Hiss Monteiro, L. "Utilização de Técnicas de Processamento de Imagens para o Reconhecimento de Placas de Veículos", dissertação de mestrado departamento de Ciência da Computação IC/ UFF, defendida em 22/08/2002.
- [18] Conci, A., Hiss Monteiro L., "Utilização de Momentos de Área na Identificação de Veículos", anais do CILAMCE 97 - XVIII Congresso Ibero Latino Americano de Métodos Computacionais em Engenharia, de 29 a 31 de outubro de 1997 em Brasília - DF, vol. IV, pp. 2019-2025.
- [19] Conci, A., Hiss Monteiro L., "Utilização de Momentos Geométricos na Inspeção Industrial", paper no. 3-268, CD do CIDIM 97- III Conferencia Ibero-americana de Ingenieria Mecanica, de 23 a 26 de setembro de 1997 em Havana - Cuba.
- [20] Conci, A., Hiss Monteiro L., "Utilização de momentos na identificação automática de elementos variados adquiridos com ruídos", vol. 3 dos anais do IV Congreso Iberoamericano de Ingeniería Mecánica, CIDIM '99, Departamento de Ingeniería Mecánica de la Universidad de Santiago de Chile, ISBN 956-7069-41-7 e 959-7069-44-1, p.100-104.

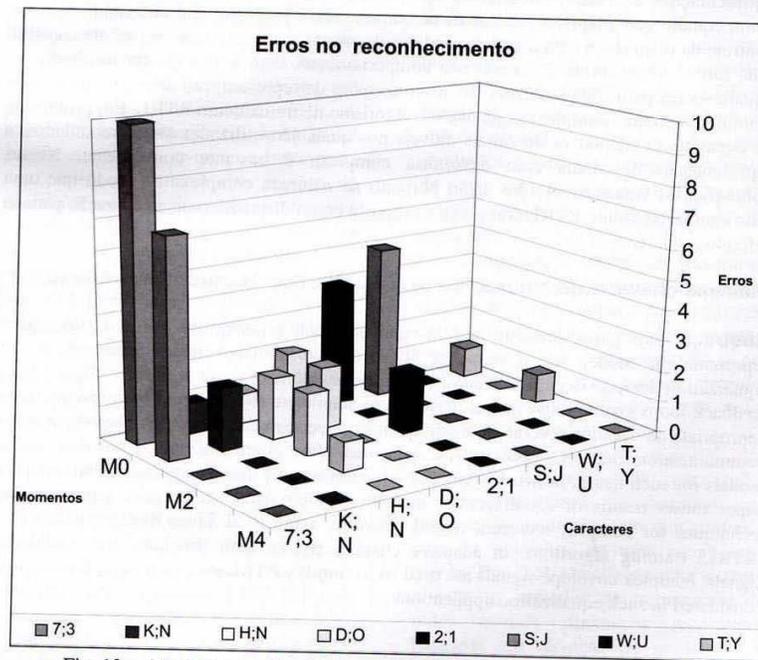


Fig. 13 . - Erros no reconhecimento usando apenas momentos invariantes