

Aplicação da técnica de momentos invariantes no reconhecimento de padrões em imagens digitais

Germano Chacon, Elton Gastardelli, Fernanda D. Moraes, Gabriel
Oliveira, Márcio P. de Albuquerque e Marcelo Portes de Albuquerque.

Centro Brasileiro de Pesquisas Físicas (CBPF), Rua Dr. Xavier Sigaud, 150, Urca, Rio de Janeiro, Brasil

Este trabalho descreve a aplicação dos momentos invariantes HU como forma de reconhecimento de padrões em imagens digitais. Apresentamos uma análise da robustez do método em função da adição de ruído em uma forma padrão e da operação morfológica de erosão. Um classificador baseado na técnica de vetores de suporte (SVM) tendo como entrada os momentos invariantes também é apresentado, permitindo a classificação de dois padrões muito parecidos.

In this work we present the application of the HU invariant moments technique as pattern recognition tool in digital images. We present an analysis of the robustness of the method according to the addition of noise in a standard pattern and as function of the morphological operation of erosion. A classifier based on support vector machine (SVM) technique with invariant moments as input is also presented, allowing the classification of two similar patterns.

Sumário

1. Introdução	1
2. Conceitos Fundamentais de Processamento de Imagens	2
2.1. Formação e aquisição de Imagens	2
2.1.1. Pixel e Conectividade	2
2.2. Pré-Processamento	3
2.3. Segmentação	3
2.4. Pós-Processamento	3
2.5. Extração de Atributos	3
2.6. Classificação e Reconhecimento	4
3. Momentos de uma Imagem	5
3.1. Momentos HU	6
4. Eficiência da Técnica de Momentos HU	6
4.1. Análise pelos pixels de borda	6
4.2. Análise pelos pixels do interior	6
4.3. Análise pela erosão sucessiva	6
4.4. Avaliação dos resultados	7
5. Exemplo de um Classificador baseado em Momentos HU	8
6. Conclusão	8
7. ANEXO I	9
References	11

1. INTRODUÇÃO

Em muitos sistemas de análise e processamento de imagens o reconhecimento de padrões é a etapa final, visando extrair da imagem as informações nela presentes. Não é difícil perceber que um sistema de reconhecimento é um sistema cognitivo e fortemente associado ao que se pretende “conhecer”. Em sistemas computacionais o objetivo do processo de reconhecimento é classificar padrões baseado em informações estatísticas ou topológicas extraídas dos conjuntos de pixels na imagem.

Este trabalho aborda a técnica de extração de características em “objetos” presentes em uma imagem digital. Os objetos são descritos por agrupamentos de pixels previamente separados do fundo da imagem e serão analisados estatisticamente. Todo objeto dentro de um sistema de reconhecimento é descrito por suas características ou atributos. Estas características são representadas em um espaço N dimensional, onde N é o número de características. Cada objeto forma assim, um vetor dentro deste espaço de características, chamado de vetor de características.

As aplicações de sistemas de reconhecimentos de padrões estão hoje por toda parte. O mais comum são os sistemas de reconhecimentos de caracteres (OCR – *Optical Character Recognition*), porem outras aplicações são: reconhecimento de faces (presentes em sites de relacionamento sociais e câmeras digitais comerciais); reconhecimento de células ou tecidos com anomalias ou cancerosas [Mohanalin, 2010], identificação de fissuras, em imagens tomográficas ou radiográficas, ou ainda no reconhecimento de pontos estratégicos em imagens de satélite.

Existem atualmente vários algoritmos dedicados ao reconhecimento de padrões, entre eles podemos citar o método dos Descritores de Fourier [Soldea, 2010], Redes Neurais Artificiais [Lindblad e Kinser, 2005] e as Máquinas baseadas em Vetores de Suporte (*Support Vector Machines*), [Vapnik, 1992]. A figura 1 apresenta um exemplo típico e difícil de separação entre dois caracteres (letras “C” e “G”), representado pelas suas características nos dois eixos do gráfico. O desafio do sistema de classificação é utilizar essa visível separação no espaço de parâmetros para criar uma linha (ou outra função matemática) de decisão que será usada para discriminar entre esses dois “objetos”.

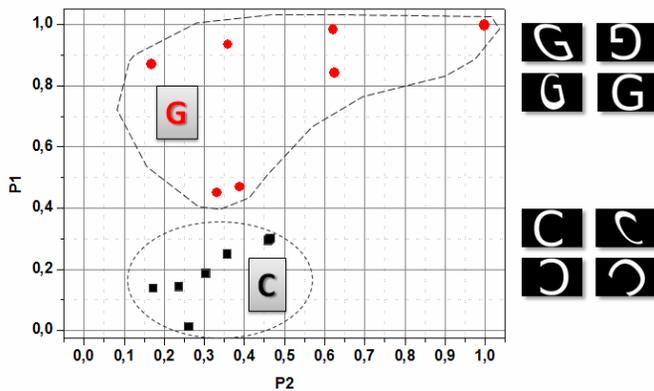


FIG. 1: Exemplo de separação de dois "objetos" (caracteres "C" e "G" em diferentes variações) representados por atributos (P1 e P2) obtidos a partir de uma imagem.

2. CONCEITOS FUNDAMENTAIS DE PROCESSAMENTO DE IMAGENS

O objetivo de um sistema de processamento de imagens é principalmente extrair informações relevantes e úteis para uma determinada aplicação. Um sistema deste tipo é composto de várias etapas e pode ser separado como ilustra a figura 2.

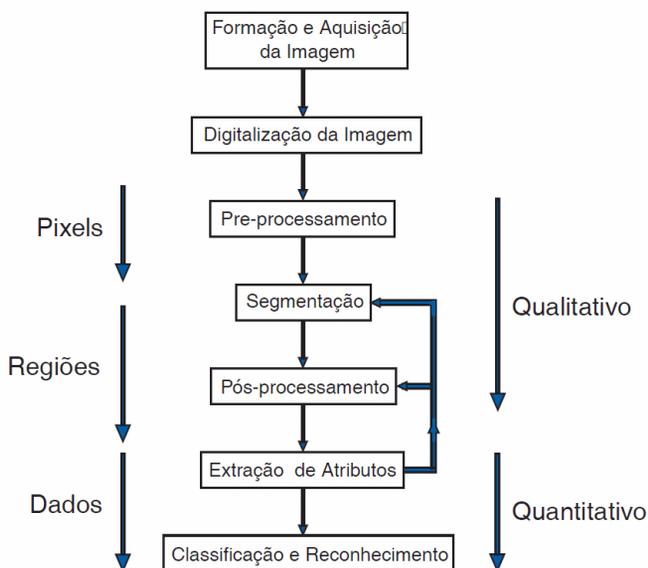


FIG. 2: Etapas de um sistema básico de processamento de imagens.

2.1. Formação e aquisição de Imagens

Uma imagem digital é a representação de uma cena em uma organização em pixels colocados lado a lado em uma simetria quadrada. Do ponto de vista matemático é representado como uma função bidimensional $f(x,y)$ onde (x,y)

são coordenadas espaciais e inteiras, definidas por convenção como $x=[1,2,\dots,nx]$, $y=[1,2,\dots,ny]$ e o valor de f no ponto (x,y) é proporcional a intensidade luminosa (ou nível de cinza) da imagem neste ponto, onde nx e ny representam respectivamente a largura e a altura da imagem

2.1.1. Pixel e Conectividade

Um pixel é o elemento básico em uma imagem. A forma mais comum para o pixel é a quadrada, sendo ele um elemento de dimensões finitas na representação da imagem digital. Frequentemente, a organização de uma imagem sob a forma de uma matriz de pixels é feita em uma simetria quadrada. Isto se deve a facilidade de implementação eletrônica, seja dos sistemas de aquisição seja dos sistemas de visualização de imagens. Este tipo de organização provoca o aparecimento de duas características importantes nas técnicas de processamento. Em primeiro lugar ele é anisotrópico, i.e., um pixel não apresenta as mesmas propriedades em todas as direções. Esta propriedade faz com que um pixel tenha quatro vizinhos de borda e quatro vizinhos de diagonal, ilustrado na Figura 3. Esta propriedade obriga que seja definido o tipo de conectividade que será utilizada para o processamento, ou B4 (considerando apenas os vizinhos de borda) ou B8 (considerando os vizinhos de borda e os de diagonal). A segunda propriedade é consequência direta da primeira, ou seja, as distâncias entre um ponto e seus vizinhos não é a mesma para qualquer tipo de vizinho. Será igual a 1 para vizinhos de borda e $\sqrt{2}$ para aqueles na diagonal.

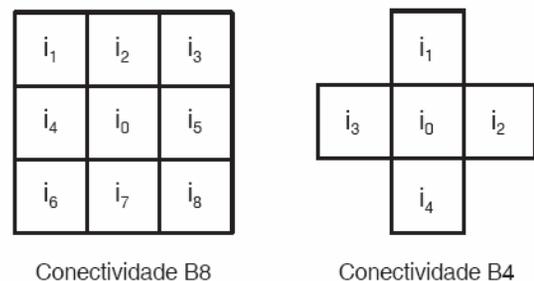


FIG. 3: Ilustração de tipos de conectividade dos pixels vizinhos ao pixel central i_0 . Conectividade B8 apresenta 8 vizinhos, sendo 4 de bordas e 4 diagonais. Conectividade B4 apresenta apenas os pixels de borda.

Atualmente existe uma diversidade de câmeras para uso em instrumentação científica. Por exemplo, em experimentos de fusão nuclear as imagens são obtidas por meio de uma câmera ultra rápida sensível ao espectro visível. Nesta aplicação, um dos modelos utilizado é a câmera ultima APX-RS da empresa PhotronTM [Photron, 2011]. Esta câmera pode adquirir imagens em uma taxa de até 250.000 quadros por segundo, em baixíssima resolução. No entanto é possível adquirir imagens de boa qualidade com resolução de 256 x 256 pixels, com uma taxa de 30.000 quadros por segundo.

2.2. Pré-Processamento

As técnicas de pré-processamento têm a função de melhorar a qualidade da imagem de forma a realçar características consideradas relevantes. Estas técnicas se dividem em duas categorias principais: métodos que trabalham no domínio espacial e métodos no domínio da frequência. Os primeiros se baseiam no processamento das coordenadas de posição (x,y) utilizando filtros que manipulam o plano da imagem, enquanto o segundo método se baseia em filtros que utilizam o espectro da imagem obtido após a aplicação da transformada de Fourier na imagem.

Um exemplo de pré-processamento em experimentos científicos é a obtenção (ou produção) de uma imagem de fundo (*background image*) que permite a correção dos efeitos de não homogeneidade de iluminação ou a detecção de movimento. Normalmente o grande desafio está na escolha apropriada da imagem de background. A abordagem mais comum é obter a imagem no próprio experimento, através de procedimentos que permitam chegar à imagem de fundo, figura 4. Em situações onde são adquiridos imagens em modo dinâmico, é possível considerar as imagens anteriores [Murari, 2010].

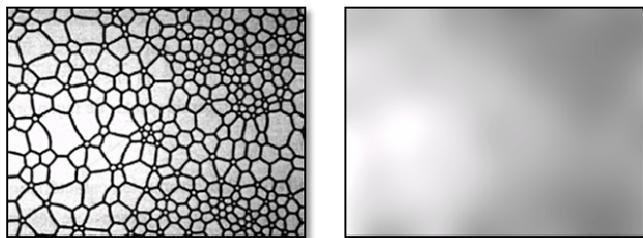


FIG. 4: Exemplo de obtenção da imagem de fundo em um experimento científico de visualização de domínios magnéticos. A imagem é obtida por meio de microscopia óptica e a imagem de fundo através da desfocalização do objeto.

2.3. Segmentação

Segmentar uma imagem significa separá-la em suas partes constituintes e que se diferenciam entre si. É usual denominar “objetos” da imagem os grupos de pixels de interesse, ou que fornecem alguma informação para o processamento. Da mesma forma, a denominação “fundo”, nesta etapa, é utilizada para os grupos de pixels que podem ser desprezados ou que não tem utilidade para o processamento. Essas denominações “objeto” e “fundo” possuem uma conotação bastante subjetiva, podendo se referir a grupos de pixels que formam determinadas regiões na imagem sem que representem um objeto, de modo literal, presente na imagem processada. A segmentação é considerada, dentre todas as etapas do processamento de imagens, a mais crítica no tratamento da informação. É na etapa de segmentação que são definidas as regiões de interesse para processamento e análise posteriores. Como consequência deste fato, quaisquer erros ou distorções presentes nesta etapa se refletem nas demais eta-

pas, de forma a produzir ao final do processo resultados não desejados e que poderão contribuir de forma negativa para a eficiência de todo o processamento.

Deve ser ressaltado que não existe um modelo formal para a segmentação de imagens. A segmentação é um processo empírico e adaptativo, procurando sempre se adequar as características particulares de cada tipo de imagem e ao objetivo final que se pretende alcançar. Apesar de existir uma grande diversidade de técnicas de segmentação de imagens, ainda assim existe atualmente, um grande interesse no estudo e desenvolvimento de novas técnicas [Portes de Albuquerque, 2004 e Zhang, 2011].

De um modo geral, as técnicas de segmentação utilizam duas abordagens principais: a similaridade entre os pixels e a descontinuidade entre eles. A técnica baseada na similaridade mais utilizada é a chamada de binarização. A binarização de imagens ou *image thresholding* é uma técnica eficiente e simples do ponto de vista computacional, sendo, portanto largamente utilizada em sistemas de visão computacional. Este tipo de segmentação é utilizado quando as amplitudes dos níveis de cinza são suficientes para caracterizar os “objetos” presentes na imagem. Na binarização, um nível de cinza é considerado como um limiar de separação entre os pixels que compõem os objetos e o fundo. Nesta técnica, se obtém como saída do sistema uma imagem binária, i.e., uma imagem com apenas dois níveis de luminância: preto e branco. A determinação deste limiar de modo otimizado é o objetivo principal dos diversos métodos de binarização existentes.

As técnicas baseadas em descontinuidade entre os pixels procuram determinar variações abruptas do nível de luminância entre pixels vizinhos. Estas variações, em geral, permitem detectar o grupo de pixels que delimitam os contornos ou bordas dos objetos na imagem. A técnica de segmentação baseada em descontinuidade mais utilizada é a chamada detecção de bordas.

2.4. Pós-Processamento

O objetivo desta fase é a correção dos principais defeitos ou imperfeições provenientes da etapa de segmentação. Normalmente são utilizadas técnicas de morfologia matemática que realizam uma análise quantitativa dos pixels da imagem em relação ao elemento estruturante adotado, figura 5 [Najman, 2010].

Exemplo de aplicação de uma técnica de pós-processamento para filtrar regiões pequenas. (a) imagem original. (b) imagem binária e (c) regiões que sobraram na imagem após pós-processamento pela operação morfológica de fechamento. Fonte: NIH-ImageJ.

2.5. Extração de Atributos

O objetivo desta fase é extrair as informações relevantes para o processamento. Esta extração é normalmente realizada a partir da imagem binária. Existem basicamente duas classes de medidas que são possíveis obter, são elas: i) atributos da imagem como um todo (“*field features*”), como por

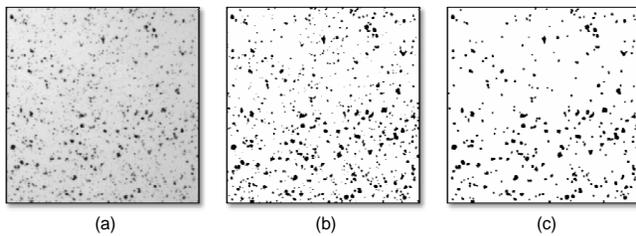


FIG. 5: Exemplo de aplicação de uma técnica de pós-processamento para filtrar regiões pequenas. (a) imagem original. (b) imagem binária e (c) regiões que sobraram na imagem após pós-processamento pela operação morfológica de fechamento. Fonte: NIH-ImageJ.

exemplo: Número de objetos presentes na imagem, área total de objetos na imagem, etc; e ii) atributos de Região ("Region Features") que se referem aos conjuntos de pixels binários (regiões) independentemente, como por exemplo: Área, perímetro, fator de forma, orientação etc.

Uma das etapas intermediárias na extração de atributos é chamada *Labelização* ou *Rotulação*. Após a etapa de segmentação obtemos uma imagem onde as regiões correspondentes aos "objetos" estão separadas daquelas correspondentes ao "fundo" da imagem. Neste ponto do sistema de processamento, as regiões de interesse estão contiguamente agrupadas por pixels que se tocam (blobs). O próximo passo é dar um rótulo (ou *label*) para cada um desses grupos de pixels. Esta identificação permitirá posteriormente parametrizar os objetos segmentados calculando para cada região de pixels contíguos um parâmetro específico, como área ou perímetro por exemplo. A figura 6 apresenta um exemplo desta técnica para uma imagem constituída de células bem delimitadas entre si. O processo de segmentação separa as regiões pertencentes às células daquelas pertencentes às regiões entre células (fundo), criando um delimitador entre elas. A etapa de "*labelização*" cria um rótulo que identifica cada uma dessas regiões para que os processos seguintes de tratamento da informação sejam concentrados em cada uma das regiões quer receberam um rótulo.

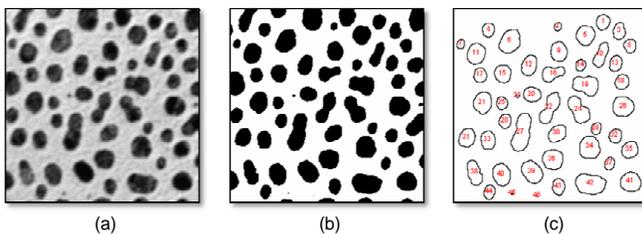


FIG. 6: Exemplo de criação dos rótulos em uma imagem. (a) imagem original. (b) imagem binária e (c) imagens com regiões rotuladas: resultado final eliminando-se as regiões que apresentam ao menos um pixel na borda da imagem. No exemplo foram encontradas 46 partículas (blobs) onde podem ser realizadas pelas etapas posteriores as mais diversas análises.

Diversos atributos podem ser obtidos para cada uma das regiões, como por exemplo: área, perímetro, área convexa, relações entre área e perímetro, retângulo (ou elipse) mínimo

que envolve o objeto, orientação, etc. Um dos possíveis atributos é caracterização das regiões pelos momentos HU, que será discutido na seção 4.

Um exemplo de uma biblioteca de processamento de imagens que realiza a extração de atributos é a *cvBlobsLib*. Esta integra a biblioteca completa de análise de imagens (OpenCV) desenvolvida em linguagem C/C++, para processamento em tempo real [OpenCV, 2011 e Bradski, 2008]. A *cvBlobsLib* atua nas imagens binárias com o objetivo de rotular todas as regiões de pixels e realizar duas funcionalidades básicas: i) extrair as regiões conectadas (com conectividade B8 em imagens binárias ou em níveis de cinza) e ii) realizar medidas nos blobs encontrados. Este algoritmo é apresentado detalhadamente em [Chang et al, 2004].

2.6. Classificação e Reconhecimento

O objetivo do reconhecimento é realizar, de forma automática, a "identificação" dos objetos segmentados na imagem. Existem duas etapas no processo de classificação de formas: o aprendizado e o reconhecimento propriamente dito. Na maior parte dos sistemas de reconhecimento de formas, os parâmetros provenientes da etapa de extração de atributos são utilizados para construir um espaço de medida à N dimensões. Os sistemas de aprendizado irão definir uma função discriminante que separe eficientemente todas as formas representadas neste espaço de medida.

O espaço de medidas é um espaço de dimensão N onde cada dimensão corresponde a um parâmetro (atributo). A escolha e a qualidade dos parâmetros para construir este espaço são essenciais para a realização de um bom processo de reconhecimento de forma. A escolha de um grande número de parâmetros leva a um espaço de grande dimensão e a uma difícil etapa de aprendizado. Um pequeno espaço de medidas pode acarretar em uma baixa caracterização da forma e a muitos erros no processo de reconhecimento. Em alguns casos pode ser interessante normalizar cada eixo para construir um espaço que seja bem distribuído e facilite o processo de classificação.

Podemos dividir o processo de aprendizado em dois tipos: os métodos supervisionados e não supervisionados. No método supervisionado, o classificador, em sua fase de aprendizado, recebe informações de como as classes devem ser identificadas. Por exemplo, em um sistema de reconhecimento de caracteres, existirão classes independentes para todas as letras do alfabeto. Por exemplo, a classe das letras "A" será representada pela ocorrência da letra A em suas diferentes variações. O aprendizado consiste então em apresentar ao sistema uma série de objetos "teste" no qual suas classes são previamente conhecidas e definir uma função que separe todas as classes entre si.

Podemos dizer que o sistema de aprendizado supervisionado age sob a supervisão de outro sistema de reconhecimento (externo, por exemplo) que identificou anteriormente os objetos testes e permitirá a construção correta de seu espaço de medida e sua função discriminante. Durante este processo devemos modificar os parâmetros que compõem o espaço de medida e permitir um melhor ajuste da função discriminante, objetivando sempre que o sistema possa realizar com mais

eficiência o processo de classificação. Ao final, é possível determinar a função discriminante responsável pela separação das diversas classes. Este processo pode ser lento e de elevado custo computacional. Somente então os objetos desconhecidos serão fornecidos a este classificador, na fase de reconhecimento.

No caso em que a classificação não é supervisionada, o classificador receberá os objetos desconhecidos e, a partir da medida dos diferentes parâmetros (atributos dos objetos presentes na imagem), ele tentará alocá-los em diferentes classes. A identificação de classes é usualmente realizada a partir da identificação de agrupamentos de objetos no espaço de medidas (o termo em inglês utilizado é "cluster").

Existem atualmente vários métodos de reconhecimento de formas. Entre eles podemos citar os métodos baseados em propriedades estatísticas dos objetos (classificador de Bayes), os métodos baseados em técnicas de distância entre os objetos na imagem e suas formas padrões (como as redes neurais artificiais) ou ainda a descrição da forma através de um dicionário ou uma linguagem básica. Neste último caso é definido uma sequência de elementos básicos que representam as formas dos objetos. Em seguida é construída uma linguagem e formada uma gramática. O reconhecimento é neste caso um reconhecimento sintático baseado nesta gramática [Gonzalez, 2002]. Existe ainda o método supervisionado de Máquina de Vetores de Suporte (*Support Vector Machines*). SVMs fazem parte de um conjunto de técnicas de aprendizado supervisionado. A separação ótima obtida pelo método é definida como sendo a melhor margem posicionada entre os pontos mais próximos entre elas, chamados de pontos de suporte.

Esta técnica é amplamente usada em reconhecimento de padrões e busca a construção de um hiperplano como superfície de decisão no espaço de medidas. No caso de padrões não linearmente separáveis, o conjunto de dados originais é mapeado em um espaço de maior dimensionalidade através de uma função chamada *kernel*. Neste novo espaço o problema se torna linearmente separável como ilustra a figura 7.

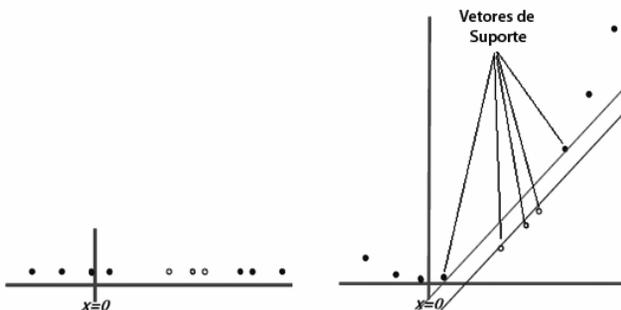


FIG. 7: Exemplo de separação de duas regiões, caracterizadas por círculos sólidos ou abertos, pela técnica de SVM. A função *kernel* aplica aos pontos originais uma transformação com o objetivo de separá-las por um único plano.

A figura 8 apresenta um exemplo de uso do método de vetores de Suporte (SVM) para a detecção automática de um padrão em imagens de fusão nuclear [Murari, 2010].

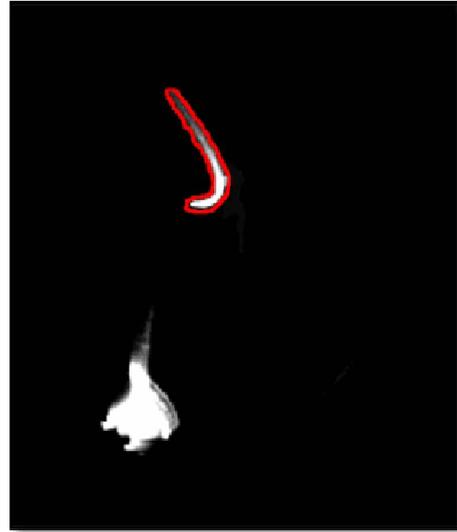


FIG. 8: Exemplo de detecção em uma imagem de fusão nuclear da região de MARFE [Murari, 2010].

3. MOMENTOS DE UMA IMAGEM

A técnica de momentos, também chamados momentos estatísticos, é um dos vários métodos utilizados para extração de características de uma imagem. Estes momentos e as funções derivadas deles se caracterizam por valores numéricos calculados a partir da imagem previamente segmentada e que descrevem a distribuição espacial dos pontos contidos na imagem ou em uma região.

Os momentos de imagem mais utilizados são os momentos regulares definidos a partir da seguinte fórmula:

$$m_{pq} = \sum_{x=1}^{nx} \sum_{y=1}^{ny} x^p y^q f(x,y)$$

Nessa fórmula m_{pq} é o momento de ordem $(p+q)$ da função intensidade $f(x,y)$ onde nx e ny representam respectivamente a largura e a altura da imagem digital. Uma imagem binária terá valores da função $f(x,y)$ iguais a 0 ou 1.

A partir dos momentos regulares podemos definir algumas medidas importantes sobre os objetos de interesse, e que são úteis na identificação de diferentes formas, por exemplo, os momentos regulares de ordem 0 e 1 são usados para o cálculo do baricentro ou centro de massa do objeto, através das seguintes fórmulas:

$$x_c = \frac{m_{10}}{m_{00}} \quad y_c = \frac{m_{01}}{m_{00}}$$

Além do baricentro é possível perceber que alguns momentos possuem um significado físico relacionado ao objeto,

como por exemplo, o momento m_{00} . Este representa a área do objeto, enquanto os momentos m_{10} e m_{01} representam as projeções nos eixos x e y respectivamente.

Com a informação obtida dos baricentros obtemos o que chamamos de momentos centrais que são definidos para imagens digitais pela fórmula:

$$\mu_{pq} = \sum_1^{nx} \sum_1^{ny} (x - x_c)^p (y - y_c)^q f(x, y)$$

Finalmente existem os momentos centrais normalizados representados por n_{pq} e definidos pela seguinte fórmula:

$$n_{pq} = \frac{\mu_{pq}}{m_{00} \left(\frac{p+q}{2} + 1\right)}$$

3.1. Momentos HU

Um dos principais problemas em um sistema de reconhecimento de padrões é a variação das características dos objetos através de uma rotação ou escala. Um sistema classificador deve identificar os objetos para o qual foi treinado da mesma forma, não importa se o objeto se apresenta transladado, rotacionado ou teve alteração na escala, ou seja, as características extraídas das imagens devem ser invariantes às alterações acima. HU [Hu-1962] propôs uma combinação de momentos que são invariantes à escala, rotação e translação. Estes momentos são representados por sete equações chamadas de momentos HU ou momentos invariantes, que são:

$$\begin{aligned} I_1 &= n_{20} + n_{02} \\ I_2 &= (n_{20} - n_{02})^2 + (2n_{11})^2 \\ I_3 &= (n_{30} - 3n_{12})^2 + (3n_{21} - n_{03})^2 \\ I_4 &= (n_{30} + n_{12})^2 + (n_{21} + n_{03})^2 \\ I_5 &= (n_{30} - 3n_{12})(n_{30} + n_{12})[(n_{30} + n_{12})^2 - 3(n_{21} + n_{03})^2] \\ &+ (3n_{21} - n_{03})(n_{21} + n_{03})[3(n_{30} + n_{12})^2 - (n_{21} + n_{03})^2] \\ I_6 &= (n_{20} - n_{02})[(n_{30} + n_{12})^2 - (n_{21} + n_{03})^2] \\ &+ 4n_{11}(n_{30} + n_{12})(n_{21} + n_{03}) \\ I_7 &= (3n_{21} - n_{03})(n_{30} + n_{12})[(n_{30} + n_{12})^2 - 3(n_{21} + n_{03})^2] \\ &+ (n_{30} - 3n_{12})(n_{21} + n_{03})[3(n_{30} + n_{12})^2 - (n_{21} + n_{03})^2] \end{aligned}$$

A etapa de análise de características é utilizada para o cálculo dos momentos HU dos objetos separados pela etapa de segmentação e rotulação. Estes momentos junto com o baricentro formam, por exemplo, o vetor de características dos objetos.

4. EFICIÊNCIA DA TÉCNICA DE MOMENTOS HU

A fim de caracterizar a técnica de momentos HU, quanto a sua robustez e imunidade a ruídos, definimos como objetos de estudo uma forma padrão. Para este caso escolhemos um círculo de raio unitário ao qual adicionamos ruído uniformemente distribuído e com variância controlada, figura 9. A relação sinal ruído é obtida através da equação:

$$SNR(dB) = -20 \log_{10} \left(\frac{1}{\sigma} \right)$$

Sobre cada forma resultante, em uma imagem digital, calculamos os sete momentos HU. Dividimos os mesmos em dois tipos: uma parte é formada apenas pelos objetos compostos dos pixels de borda e a outra possui todo o seu interior preenchido (forma sólida). A diferença entre os dois nos permitirá realizar uma análise da influência dos pixels de borda e de interior na determinação dos momentos.

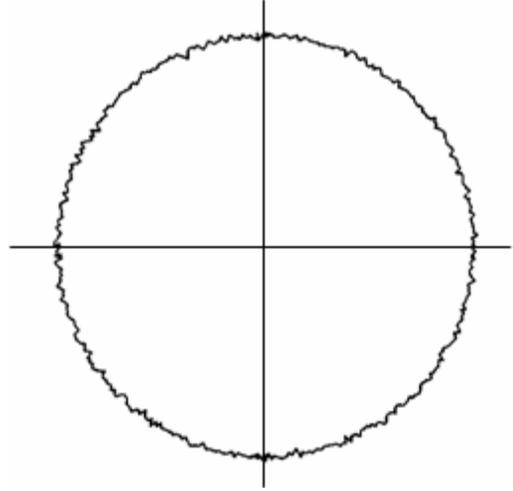


FIG. 9: Objeto base para a análise da técnica de momentos Invariantes. À forma inicial (círculo) foi adicionado um ruído de 0.01 de amplitude, tendo uma relação sinal/ruído (SNR) igual a 40 dB.

Os resultados são apresentados em um gráfico dos sete Momentos x SNR para os dois tipos de análises apresentadas anteriormente.

4.1. Análise pelos pixels de borda

Nesta abordagem apenas aqueles pixels pertencentes à borda do objeto são analisados. Conforme há o aumento do ruído adicionado à borda a figura se torna completamente irregular, se afastando da imagem original (círculo de raio unitário), figura 10. O gráfico comparativo dos sete momentos HU, normalizados pelo valor dos momentos obtidos na figura inicial (círculo com relação SNR de 60 dB) e apresentado na figura 11.

4.2. Análise pelos pixels do interior

A mesma análise foi realizada para o objeto completamente preenchido para a mesma variação do ruído na borda, figura 12. O resultado é apresentado na figura 13.

4.3. Análise pela erosão sucessiva

Nós realizamos a análise dos momentos invariantes após n sucessivas erosões morfológicas de uma determinada forma, apresentado na figura 14. Da mesma forma o objetivo foi realizar uma análise de distanciamento da forma em relação

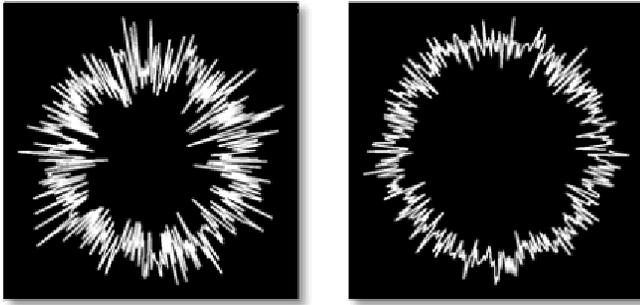


FIG. 10: Exemplo de dois círculos com diferentes ruídos adicionado aos pixels de borda. O objeto da esquerda tem uma relação SNR de 10 dB. O da direita é igual a 15 dB.

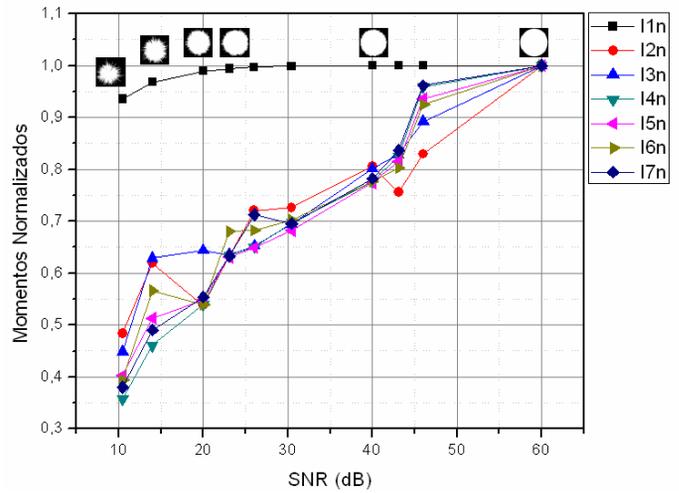


FIG. 13: Comparativo da variação dos sete momentos HU para objetos sólidos em função do crescimento do ruído.

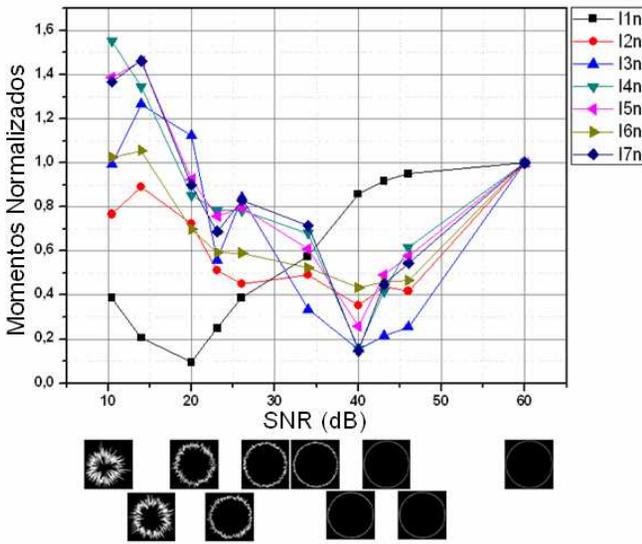


FIG. 11: Comparativo da variação dos sete momentos HU em função do crescimento do ruído na borda do objeto.

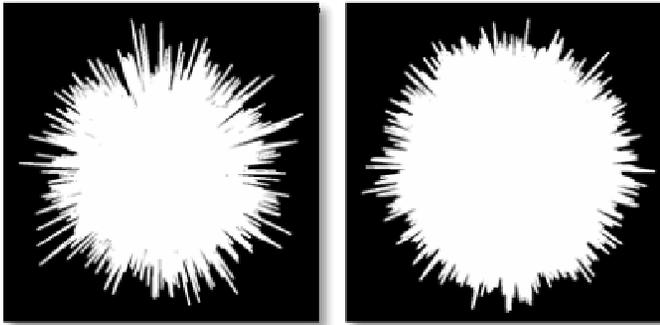


FIG. 12: Exemplo dos objetos (círculos) sólidos com diferentes ruídos adicionado aos pixels de borda. Objeto da esquerda: SNR = 10 dB. Objeto da direita: SNR = 15 dB.

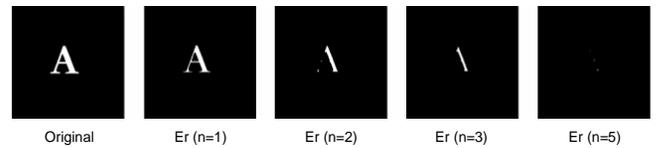


FIG. 14: Análise dos momentos invariantes após a realização de sucessivas operações de erosão morfológica para o caractere "A".

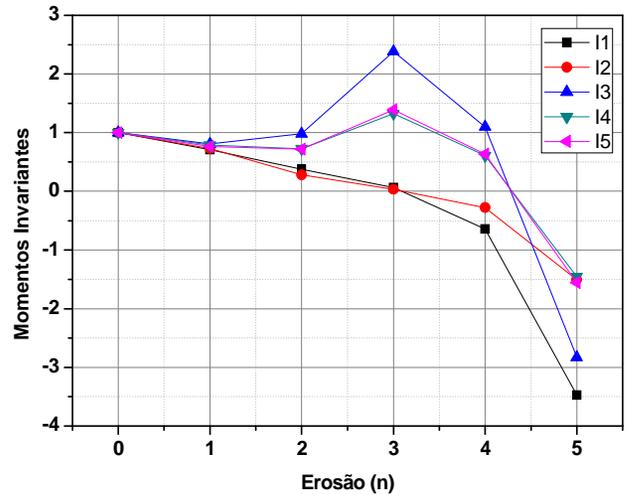


FIG. 15: Evolução dos momentos HU para um objeto (caractere A) após a realização do processo de erosão. Os momentos I6 e I7 não apresentaram variações.

à imagem original ($n=0$). Os valores apresentados foram normalizados por aqueles encontrados na imagem original, figura 15.

4.4. Avaliação dos resultados

Esta análise permite uma verificação comparativa das abordagens apresentadas que são comumente utilizadas em processamento de imagens após a segmentação dos objetos. Avaliando os gráficos anteriores podemos fazer as seguintes

considerações:

1. Na análise pelos pixels de borda, todos os momentos passam por um mínimo, sendo que para o I1 este é em 20 dB e para os outros em 40 dB.
2. Na análise por todos os pixels pertencentes ao objeto (sólido), é possível verificar que todos os momentos decaem com o aumento do ruído, não há uma passagem por um mínimo. O decaimento relativo é mais suave quando comparada com abordagem pela borda do objeto. Isso se deve principalmente devido a participação no cálculo de todos os pixels dos objetos que não sofreram modificação.
3. Nos dois casos, o momento I1 é menos sensível ao ruído, com uma evolução mais suave em função da amplitude do ruído. Todos os outros (2 ao 7) tem um decaimento muito rápido comparado com a análise dos objetos preenchidos.
4. No caso da avaliação pela erosão, é possível perceber um distanciamento contínuo do objeto de sua forma original. Como a remoção sucessiva dos pixels de borda é possível verificar que o método pode ser bastante robusto com as pequenas variações inseridas na etapa de segmentação.

5. EXEMPLO DE UM CLASSIFICADOR BASEADO EM MOMENTOS HU

Com o objetivo de exemplificar o uso da técnica dos Momentos HU foi desenvolvida uma aplicação gráfica em Matlab¹. Para a implementação deste código utilizamos também a técnica de reconhecimento de padrões SVM apresentada anteriormente. O programa é composto por três painéis, definidos como:

1. Principal: permite a configuração dos parâmetros necessários à classificação.
2. Imagens: usado para visualização das imagens analisadas.
3. Gráfico: apresenta os resultados dos momentos previamente selecionados com a superfície de separação entre as classes e um gráfico adicional com a porcentagem de acerto do classificador.

Como teste utilizamos o classificador para identificar os caracteres “C” e “G”, devido principalmente as semelhanças entre esses dois símbolos. As imagens foram elaboradas através do software de processamento de imagens GIMP. Em um conjunto de 28 imagens (14 para cada caractere) foi realizado o procedimento de deformação (por perspectiva e livre). A partir desta base de imagens o programa executa três etapas no processo de classificação, a primeira é

a extração de atributos obtida com o cálculo dos sete momentos HU para cada imagem da base. Estes cálculos foram executados por uma função escrita em Matlab (ver lista completa do código no Anexo 1 deste documento). A segunda e terceira etapas do processo correspondem respectivamente ao treinamento e classificação e foram realizadas por dois programas desenvolvidos em C/C++, que utilizam uma biblioteca de código aberto chamada LibSVM (implementação da técnica de máquinas de vetores de suporte, especializada em classificação de múltiplas classes e validação cruzada – [Chih-Chung e Chih-Jen, 2011]).

O programa separa aleatoriamente 70% da base dados para o treinamento e 30% para classificação e são executados a partir do código Matlab. Os resultados obtidos podem ser visualizados no diferentes painéis, conforme apresentado na figura 16.

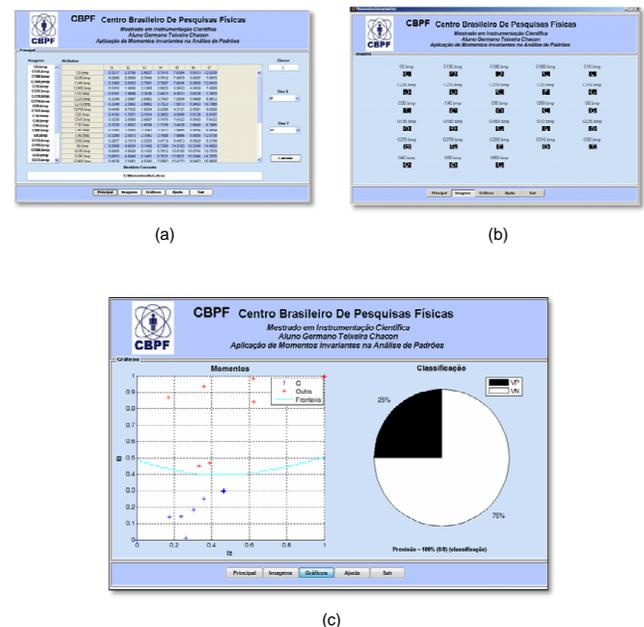


FIG. 16: (a) Painel Principal após o cálculo dos Momentos invariantes. (b) Painel com as imagens da base utilizada pelo classificador. (c) Resultado da separação dos caracteres com a técnica dos momentos HU e classificador SVM (utilização dos momentos I2 e I3).

6. CONCLUSÃO

Neste trabalho detalhamos as potencialidades de uso dos momentos invariantes para a caracterização de padrões em imagens digitais. Apesar de ter sido proposta inicialmente em 1962, por Ming-Kuei Hu do Departamento de Engenharia Elétrica da Universidade de Syracuse em Nova York, esta técnica tem sido bastante utilizada atualmente devido ao pré-processamento realizado por bibliotecas específicas de processamento de imagens. Recentemente, outras técnicas com o mesmo objetivo têm sido propostas e devem também ser avaliadas com relação à robustez (e.g. a técnica SIFT - Scale-invariant feature transform – [Lowe, 2004]).

¹ Uma versão completa do código em MATLAB está disponível no sítio internet <http://www.cbpf.br/cat/pdsi/hu> para versão do MATLAB R2008A com toolbox de processamento de imagens.

Com o classificador desenvolvido em Matlab apresentamos as funcionalidades das inúmeras aplicações possíveis do reconhecimento de padrões, através do reconhecimento automático de caracteres. Este pode ser estendida de forma simplificada para uso com qualquer forma desde que previamente segmentada e tratada.

Com relação à caracterização dos Momentos HU como técnica para extração de atributos, a aplicação nos permitiu observar que, em função dos momentos utilizados, podemos obter uma grande variedade de combinações de atributos. É possível organizar o espaço de características com a finalidade de encontrar um plano de separação e simplificar as etapas de treinamento e classificação.

A aplicação também nos permitiu avaliar a técnica de aprendizado supervisionado de máquina de vetores de suporte (SVM), esta mostrou a maioria das vezes, uma eficiência no nosso exemplo de 100% de acertos inclusive quando o espaço de características apresentava uma disposição não separável linearmente. O grande benefício desta técnica é que dispomos de uma série de funções Kernel para o cálculo do hiperplano de decisão, como por exemplo, funções polinomiais, radial ou tangente hiperbólica, este fato confere grande flexibilidade a técnica pois nos permite escolher entre as funções aquela que apresenta uma maior separação entre as classes e o menor custo computacional.

Uma complementação deste trabalho será comparar a eficiência deste classificador com outro, em desenvolvimento, utilizando outra técnica de classificação, como por exemplo, a análise por componentes principais (PCA) ou Redes Neurais Artificiais.

7. ANEXO I

Código para realização dos Momentos Invariantes em MATLAB

Este código em MATLAB está disponível na página eletrônica do CBPF na seguinte URL:
<http://www.cbpf.br/cat/lpdsi> (Ago 2011)
<http://www.cbpf.br/~chacon/mhu/>

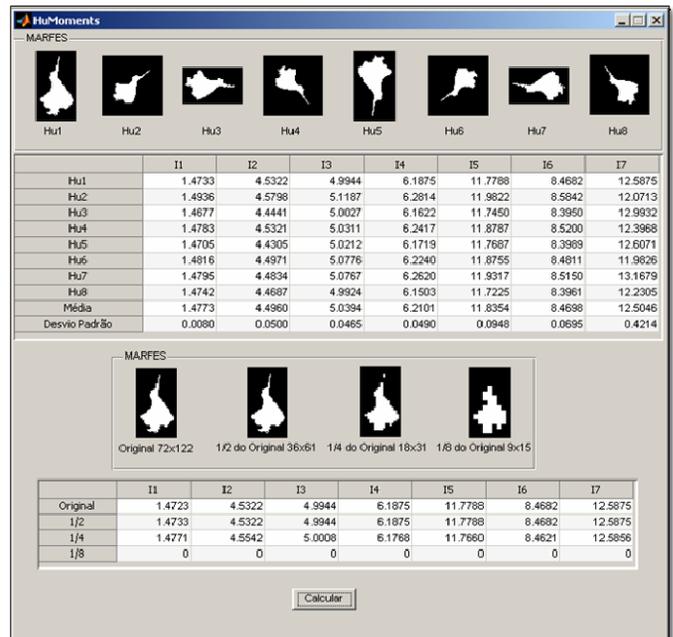


FIG. 17: Interface do Programa "HuMoments.m" desenvolvido em MATLAB.

Listagem da função MomentosHu.m desenvolvida em MATLAB

(Esta função é responsável pelo cálculo dos momentos HU, e integra a aplicação gráfica chamada MomentosInvariantes.m escrita em MATLAB R2008a instalado com o toolbox image processing).

```
function [m00,c,mus,I]= MomentosHu(img)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Esta Função Calcula os sete Momentos HU de uma imagem %
% Binária no formato bmp %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Parâmetros %
%img - String com o nome da imagem %
%m00 - Momento de ordem 0 que representa a área do objeto%
%c - Baricento do objeto %
%mus - Momentos centrais %
%I - Momentos Hu %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
F=imread(img);
%format double
m00=double(0);
m10=double(0);
m01=double(0);
mus=zeros(1,7);
%Momento de ordem zero e um
tamanho=size(F); %Tamanho da figura
V=tamanho(1); %Tamanho do eixo vertical
H=tamanho(2); %Tamanho do eixo Horizontal

for y=1:V
    linha=y;
    for x=1:H
        coluna=x;
        m00=double(m00 + F(linha,coluna));
        m10=double(m10 + (x*F(linha,coluna))); %p=1 q=0
        m01=double(m01 + (y*F(linha,coluna))); %p=0 q=1
    end
end
xm=double(m10/m00);
ym=double(m01/m00);
c=[xm ym];

%Momentos Centrais Normalizados
m11=double(0); %p e q =1
m20=double(0); %p=2 q=0
m02=double(0); %p=0 q=2
m12=double(0); %p=1 e q=2
m21=double(0); %p=2 e q=1
m30=double(0); %p=3 q=0
m03=double(0); %p=0 q=3

for y=1:V
    linha=y;
    for x=1:H
        coluna=x;
        m20=double(m20 + ( ((x-xm)^2)* F(linha,coluna) ));
        m02=double(m02 + ( ((y-ym)^2)* F(linha,coluna) ));
```

```

m11=double(m11 + ( ((x-xm) * (y-ym))* F(linha,coluna) ));
m30=double(m30 + ( ((x-xm)^3)* F(linha,coluna) ));
m03=double(m03 + ( ((y-ym)^3)* F(linha,coluna) ));
m12=double(m12 + ( ((x-xm) * ( (y-ym)^2) ) * F(linha,coluna) ));
m21=double(m21 + ( ( ( (x-xm)^2 ) * (y-ym) ) * F(linha,coluna) ));

end
end

mus = [m20 m11 m02 m30 m21 m12 m03];
%O expoente gama é definido como ((p+q)/2)+1 e assume dois valores
gama=double(2);
m00G=double(m00^2);
N20=double(m20/m00G);
N02=double(m02 /m00G);
N11=double(m11 /m00G);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
gama=5/2;
m00G=double(m00^(5/2));
N30=double(m30 / m00G);
N03=double(m03 / m00G);
N12=double(m12 / m00G);
N21=double(m21 / m00G);
I=zeros(1,7);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
S1=(N30 + N12);
S2=(N21 + N03);
S3=(N30 - 3*N12);
S4=(N20 - N02);
S5=(3*N21 - N03);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
S6 = ( (S1^2) - (3*(S2^2)) );
S7 = ( (3*(S1^2)) - (S2^2) );
S8 = (S3*S1);
S9 = (S5*S2);
S10= (S5*S1);
S11= (S3*S2);
%Momentos HU
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
I(1)= (N20 + N02);
I(2)= ( S4^2 ) + ( (2*N11)^2 );
I(3)= ( S3^2 ) + ( S5^2 );
I(4)= ( S1^2 ) + ( S2^2 );
I(5)= ( S8 * S6 ) + (S9*S7);
I(6)= ( S4*((S1^2)-(S2^2)) ) + (4*N11*S1*S2);
I(7)= ( S10*S6 )+ (S11*S7);
I=abs(log(I));

```

-
- [1] G. Bradski and A. Kaehler, “*Learning OpenCV: Computer Vision with the OpenCV Library*”, 978-0596516130, O’Reilly Media – 2008.
- [2] Fu Chang, Chun-Jen Chen, Chi-Jen Lu, “A linear-time component-labeling algorithm using contour tracing technique”, *Computer Vision and Image Understanding*, vol. 93, 2, pp.206-220, Feb, 2004.
- [3] Chih-Chung Chang and Chih-Jen Lin, *LIBSVM library for support vector machines*, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
- [4] Rafael Gonzalez, Richard Woods e Steven Eddin; “*Digital Image Processing*”, Editora: Prentice Hall, ISBN 020-11-80-758, 2002.
- [5] M. K. Hu, “*Visual Pattern Recognition by Moment Invariants*”, *IRE Trans. Info. Theory*, vol. IT-8, pp.179–187, 1962.
- [6] - Thomas Lindblad e Jason M. Kinser, “Image processing using pulse-coupled neural networks”, Editora: Springer-Verlag, 2005.
- [7] Lowe, D. G., “*Distinctive Image Features from Scale-Invariant Keypoints*”, *International Journal of Computer Vision*, vol.60, n.2, pp. 91-110, 2004.
- [8] Software aberto para manipulação de imagens GNU. URL para acesso na Internet: <http://www.gimp.org/>
- [9] Mohanalin, Prem Kumar Kalra, Nirmal Kumar, “*An automatic method to enhance microcalcifications using Normalized Tsallis entropy*”, *Signal Processing*, vol.90, Issue 3, pag. 952-958, Março, 2010.
- [10] A. Murari et al; “Algorithms for the automatic identification

- of MARFEs and UFOs in JET database of visible camera videos”, IEEE Trans. on Plasma Science, 2009.
- [11] L. Najman e H. Talbot, “*Mathematical morphology: from theory to applications*” ISTE-Wiley, ISBN 978-18-48-21215-2, (520 pp.) Junho, 2010.
- [12] OpenCV (Open Source Computer Vision) web site: <http://opencv.willowgarage.com/wiki/> 2011.
- [13] Photron, Fastcam APX-RS, página visitada na Internet em 4/7/2011, url: http://www.photron.com/index.php?cmd=product_general&product_id=11&product_name=FASTCAM+APX+RS&home=1
- [14] M. Portes de Albuquerque, I. A. Esquefb, A. R. Gesualdi Mell e Marcelo Portes de Albuquerque, “*Image thresholding using Tsallis entropy*”, Pattern Recognition Letters, vol.25, issue 9, pag.1059-1065, Julho, 2004.
- [15] O. Soldea, M. Unel e A. Ercil; “*Moments of Elliptic Fourier Descriptors*”, Proceedings of the 20th International Conference on Pattern Recognition, Istambul/Turquia, 2010.
- [16] B. E. Boser, I. Guyon, and V. Vapnik, “*A training algorithm for optimal margin classifiers*”, In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pp. 144-152. ACM Press, 1992.
- [17] Yudong Zhang e Lenan Wu, “*Optimal Multi-Level Thresholding Based on Maximum Tsallis Entropy via an Artificial Bee Colony Approach*”, vol.13, n.4, pag. 841-859; Entropy, 2011.