

Uma ferramenta morfométrica

MAC 499 - Trabalho de formatura supervisionado

aluno: Heitor Lellis Nicolielo No. USP 4941762

orientador: Roberto Marcondes Cesar Junior

colaboradores: Fabio Machado e Erika Hingst-Zaher

Sumário

1	Introdução	2
1.1	Motivação e objetivos	2
1.2	O processo de análise de imagens	2
1.3	Pré-processamento	3
1.4	Transformações	3
1.5	Classificação	3
2	Conceitos	3
2.1	Imagem digital	3
2.2	Vizinhança entre pontos	4
2.3	Segmentação	4
2.3.1	<i>Thresholding</i>	5
2.4	Área	5
2.5	Contorno	6
2.6	Perímetro e maior diâmetro	10
2.7	Centróide	10
2.8	Raio médio	11
2.9	Proporção de escala	11
2.10	Série de Fourier	11
2.11	Transformada de Fourier	12
2.11.1	FFT	14
2.12	Curvatura	14
2.12.1	Curvograma	14
2.13	Energia de dobramento	15

3	Atividades realizadas	16
4	Resultados	16
5	Conclusão	16

1 Introdução

1.1 Motivação e objetivos

A análise de formas é uma importante área da visão computacional, que tem aplicações em diversos campos, como biologia, medicina, geologia, indústria etc. Seu objetivo é caracterizar formas espaciais, extrair medidas precisas e representativas sobre elas e classificá-las. Muitas dessas medidas são, na prática, ao menos inviáveis de serem extraídas sem a captura de imagens e o auxílio do computador. A forma dos objetos, sua disposição e as texturas contém todas as informações explícitas de uma imagem, que podem ser capturadas através da análise de formas em conjunto com o estudo de padrões. Tal processo envolve várias tarefas. Uma delas é formalizar conceitos, muitas vezes subjetivos, em medidas e procedimentos bem definidos, passíveis de implementação.

Este trabalho serviu de auxílio ao estudo que o mestrando Fabio Machado propôs em conjunto com sua orientadora Erika Hingst-Zaher, ambos do Instituto de Biologia da USP. Tal estudo pretende quantificar a forma de ovos de maneira biologicamente significativa, para investigar a discriminação em grupos de distintas hierarquias taxonômicas (variação intra e interespecífica, entre famílias etc).

O objetivo desse trabalho foi desenvolver uma ferramenta que extraísse medidas das formas de objetos a partir de imagens digitais para auxiliar estudos morfométricos em geral.

Para contextualizar a área em que este trabalho se insere, descreveremos em linhas gerais o processo de análise de imagens.

1.2 O processo de análise de imagens

As tarefas do processo de análise de imagens podem tipicamente ser divididas em três grandes etapas sequências: pré-processamento da forma, transformações de forma e classificação da forma, como mostra a figura ???. O foco

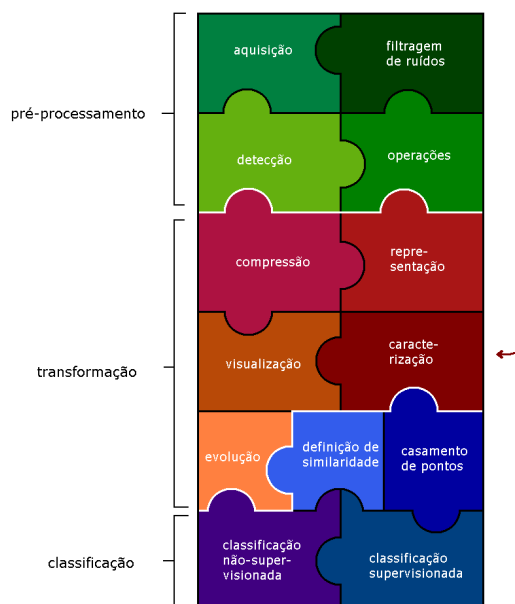


Figura 1: Tarefas típicas do processo de análise de imagens. Figura extraída de [Costa]

deste trabalho é a caracterização (ou descrição) de formas. Definiremos a seguir algumas das tarefas do processo de análise de imagens, enfatizando as mais relevantes para este estudo.

1.3 Pré-processamento

O pré-processamento da forma envolve a obtenção e digitalização da imagem, a filtragem de ruídos, a detecção da forma de interesse na imagem (por exemplo, via segmentação, definida a seguir), a normalização da forma para viabilizar comparação, a localização de *landmarks* (conjunto de pontos que definem correspondência entre as formas) e outras operações sobre a forma.

1.4 Transformações

Foco do trabalho. Onde é feita a extração de características da forma, isto é, onde se obtém informações sobre a imagem.

1.5 Classificação

Quando se quer separar as formas em classes, baseando-se nas descrições obtidas na etapa de transformações, há duas situações típicas: quando as classes são pré-definidas e possivelmente há exemplos delas, ou existe uma fase de treinamento, dizemos que a classificação é **supervisionada**. Quando não se sabe *a priori* quais são as classes de uma população de formas, dizemos que a classificação é **não-supervisionada**, envolvendo algoritmos de *clustering* (ou agrupamento). Ambos os casos envolvem a comparação entre formas e a definição de quão similares são duas formas.

2 Conceitos

Nesta seção, apresentamos os conceitos estudados necessários para realização e compreensão do trabalho.

2.1 Imagem digital

Uma imagem digital é uma representação de uma imagem bidimensional como um conjunto finito de valores digitais, chamados de *pixels* (contração no inglês de *picture element*). Um *pixel* é o menor elemento de uma imagem, contendo a intensidade da cor da imagem no ponto ao qual corresponde.

Uma imagem digital pode ser representada digitalmente através de uma matriz, onde cada elemento é um valor no sistema aditivo de cores RGB (acrônimo do inglês, *Red*, *Green* e *Blue*), isto é, um número correspondente a uma cor formada por uma ponderação (variando de zero a 255) de vermelho, verde e azul. Cada uma dessas cores (vermelho, verde e azul) é chamada de banda.

Outra maneira comum de representação de imagem digital é em níveis de cinza. Uma matriz de inteiros variando entre zero e 255 representa os tons de cinza da imagem: quanto mais baixo é o valor de um pixel, mais escura é sua tonalidade.

Um terceiro tipo de imagem digital é a imagem binária, que só possui valores zero ou um (usualmente visualizados como branco e preto, respectivamente). Esse tipo de imagem em geral é o produto da segmentação, definida mais adiante, e é muito útil para a análise de formas.



Figura 2: Ilustração das duas definições de vizinhança. À esquerda são destacados os 4-vizinhos do ponto central. À direita, os 8-vizinhos.

2.2 Vizinhança entre pontos

Duas definições de vizinhança entre pontos podem ser adotadas (figura 1):

- 4-vizinhança: os vizinhos do ponto (i, j) são os pontos $(i, j-1)$, $(i-1, j)$, $(i+1, j)$ e $(i, j+1)$.
- 8-vizinhança: os vizinhos do ponto (i, j) são os pontos da 4-vizinhança e também os pontos $(i-1, j-1)$, $(i-1, j+1)$, $(i+1, j-1)$ e $(i+1, j+1)$.

2.3 Segmentação

A segmentação é a delimitação de o que é objeto de interesse e o que é apenas fundo na imagem. É um procedimento que recebe uma imagem colorida ou em níveis de cinza e produz uma imagem binária onde os pixels acesos (com valor 1) são apenas aqueles que pertencem ao(s) objeto(s) de estudo.

É um dos mais importantes passos para bons resultados. Erros na segmentação podem levar a problemas difíceis de serem tratados na análise da imagem.

Há diversas abordagens, entre elas algumas baseadas em limiarização, em crescimento de regiões, em redes neurais, em métodos estatísticos, na transformada de Fourier, na transformada de *wavelet*, na transformação *Watershed* e em autômatos celulares (exemplo: *GrowCut*). A mais simples delas é mostrada a seguir.

2.3.1 *Thresholding*

Thresholding ou limiarização é um método de segmentação de imagens que seleciona apenas os pontos cujo valor é maior que um certo limiar. O valor do ponto depende da representação; pode ser o nível de cinza, o valor RGB ou uma ponderação do valor RGB (em particular, pode-se tomar apenas uma banda para aplicar a limiarização). O valor do limiar pode ser definido de várias maneiras: interativamente por uma pessoa que tenta encontrar um valor adequado; automaticamente analisando-se, por exemplo, o histograma de frequência de intensidades da imagem; ou pode ser pré-definido caso as imagens a serem processadas seguirem um padrão conhecido. Esta técnica se baseia na hipótese de que os níveis de intensidade do objeto e do fundo são diferentes e portanto, existe um valor capaz de separá-los.

2.4 Área

Tendo o objeto de estudo segmentado na imagem, o cálculo da área é feito apenas contando-se o número de pixels que pertencem ao objeto. Obviamente esta medida varia conforme a escala. Só é uma medida comparável entre duas imagens se lhe for aplicada uma proporção de escala ou se ambas as imagens forem obtidas mantendo a mesma distância entre objeto de estudo e dispositivo de aquisição da imagem.

2.5 Contorno

O contorno de um objeto pode ser definido de duas formas:

1. conjunto de pixels pertencentes ao objeto que fazem fronteira com os pixels pertencentes ao fundo, chamado de contorno interno
2. conjunto de pixels pertencentes ao fundo que fazem fronteira com os pixels pertencentes ao objeto, chamado de contorno externo

A figura 2 ilustra um exemplo das duas definições.

O algoritmo usado para encontrar o primeiro tipo de contorno é descrito a seguir. A idéia é começar percorrendo a imagem no sentido da leitura ocidental até encontrar o primeiro pixel do objeto, encontrando assim o primeiro pixel do contorno (à esquerda do primeiro pixel encontrado do objeto). A partir deste primeiro ponto, procura-se por transições fundo-objeto entre

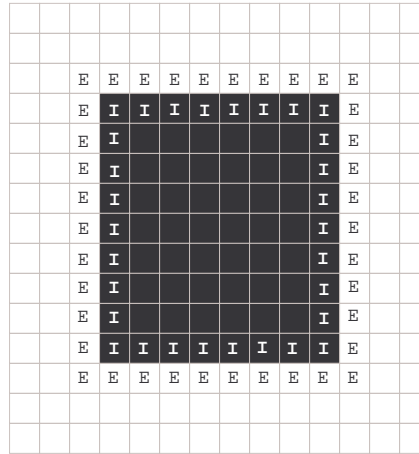


Figura 3: Exemplo de contornos interno (I) e externo (E) de uma forma discreta.

seus oito-vizinhos e define-se como próximo ponto do contorno aquele que pertencer a última transição fundo-objeto seguindo o sentido anti-horário, partindo-se do último pixel visitado.

Procedimento para extração do contorno

```
// encontrando primeiro pixel do contorno
para i <- 1 até altura da imagem, faça
  para j <- 1 até largura da imagem, faça
    se (i,j) for objeto
      saia dos laços;
    fim
  fim
fim

se (i,j) está fora da imagem // não há objeto
  devolva o conjunto vazio;
senão
  adicione (i,j) ao contorno;
```

```

// encontrando o segundo pixel do contorno
nCandidatos <- 0;
para todo candidato em (8-vizinhos de (i,j) de oeste a
sudeste em sentido anti-horário), faça
    se o candidato é fundo e o próximo candidato é objeto
        se nCandidatos é zero
            prox = candidato;
        fim
        nCandidatos++;
    fim
fim

seja B um valor booleano; // diz se há mais de
                        // um caminho a percorrer
se nCandidatos > 1, então B <- verdadeiro;
senão, B <- falso;

seja Dcn a direção de (i,j) a prox; // Dcn = Direction from
                        // Current to Next
enquanto prox != (i,j) ou B, faça
    se (i,j) for igual a prox e B
        B <- falso;
    fim
    adicione prox ao contorno;
    Dpc <- Dcn; // Dpc = Direction from Previous to Current
    Dcn <- encontre_próximo(prox, Dpc);
fim
fim
fim

// aqui as direções são tratadas como inteiros de zero a oito
Sub-procedimento encontre_próximo(prox, Dpc)
    Dcp <- inverte_direção(Dpc);
    para r <- 0 até 6, faça // varre as sete direções possíveis
        De <- (Dcp + r) mod 8; // "mod" é a operação
        Di <- (Dcp + r + 1) % 8; // módulo ou resto da divisão
        Pe <- partindo do ponto prox, ande um pixel na direção De;

```

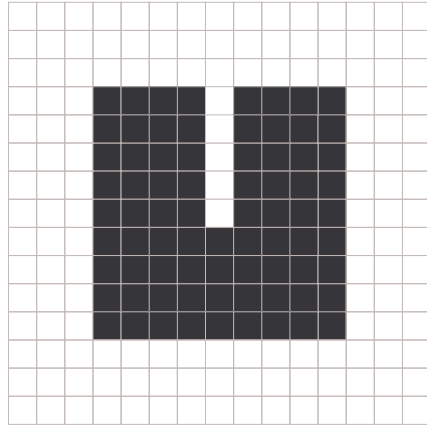



Figura 4: Exemplo de imagem binária com reentrância de um *pixel* de largura onde o trajeto do contorno deve entrar na reentrância e sair dela.

```

    Pi <- partindo do ponto prox, ande um pixel na direção Di;
    se Pe for fundo e Pi for objeto
        prox <- Pe;
        Dcn <- De;
    fim
fim
devolva Dcn;
fim

```

Dessa forma, segue-se o contorno no sentido anti-horário.

Note que o último pixel visitado também é candidato a próximo pixel do contorno. Isso ocorre para que possamos extrair o contorno externo de uma forma como a da figura 3, onde há uma reentrância de largura um pixel. Nela, o algoritmo consegue sair da reentrância uma vez tendo entrado nela.

Essa escolha do último pixel pertencente a uma transição e não do primeiro é necessária para um caso específico, ilustrado na figura 4, onde há mais de um caminho a seguir. É fácil ver que, seguindo o algoritmo proposto acima, o contorno segue primeiramente pela parte “interna” no sentido horário e depois pela parte “externa” no sentido anti-horário. Se a escolha não fosse desta forma, percorreria-se apenas a parte “externa” do objeto.

O procedimento para encontrar o contorno interno é análogo. Basta considerar que os pontos do contorno devem pertencer ao objeto e não ao fundo.

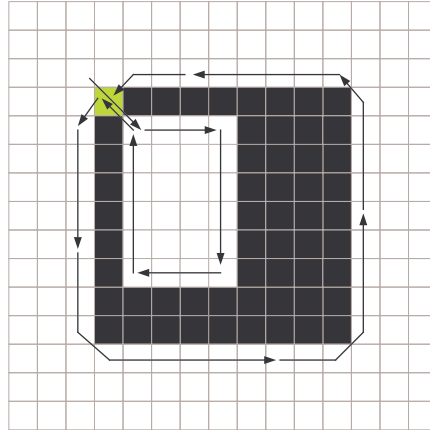


Figura 5: Exemplo de imagem binária em que o contorno deve seguir também pela parte “interna” da forma. O pixel destacado é o primeiro pixel do contorno.

2.6 Perímetro e maior diâmetro

O perímetro é definido como o número de pixels do contorno externo. O maior diâmetro é o maior segmento de reta inteiramente contido no objeto de interesse. É fácil deduzir que os extremos deste segmento pertencerão ao contorno interno. Para encontrá-lo, é necessário fazer uma busca exaustiva, calculando a distância (norma euclidiana) entre cada par de pixels do contorno. A intuição pode sugerir que precisemos buscar apenas entre os pontos “diametralmente” opostos, mas a figura 5 ilustra um exemplo em que o maior diâmetro não passa pelo seu centróide ou baricentro (definido a seguir), já que este não está contido no objeto de interesse neste caso.

2.7 Centróide

O centróide é o ponto da figura cujas coordenadas são as médias de todas as coordenadas dos pixels do objeto de interesse. Tem a propriedade de ser o centro de massa (baricentro) do objeto. Nem sempre pertence ao objeto, como é mostrado na figura 5.

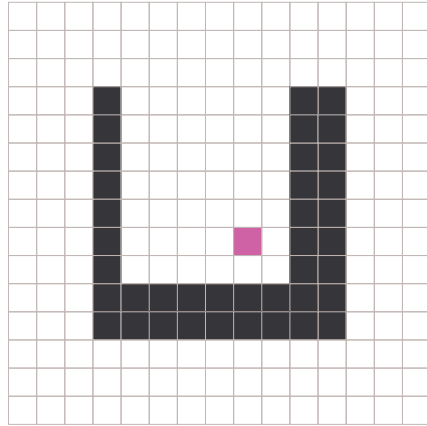


Figura 6: Uma imagem binária com o maior diâmetro destacado em amarelo e seu centróide marcado em verde.

2.8 Raio médio

Para inferir o raio médio de uma figura, dados o centróide da imagem e o contorno do objeto, calcula-se a distância média entre cada ponto do contorno e o centróide. Teremos a quantidade média de pixels que formam o raio médio da figura.

2.9 Proporção de escala

Uma vez que as imagens não são extraídas todas com mesma distância entre a lente fotográfica e o objeto, para que possamos comparar as medidas entre diferentes imagens, é preciso extrair as medidas em unidades métricas. Para isso, é preciso saber a relação de escala entre pixels e milímetros. Se dispomos de uma referência de escala circular com raio conhecido, basta encontrar o centróide desta referência, calcular seu raio médio e calcular a proporção (raio médio/raio real da referência). Assim teremos a relação pixels/milímetros.

2.10 Série de Fourier

A série de Fourier é uma série harmônica que serve para representar funções periódicas como soma de funções exponenciais complexas (ou senóides, equivalentemente). Sendo $2L$ o período da função, seguem as duas formas de

representação, a complexa:

$$f(t) = \sum_{n=-\infty}^{\infty} \frac{1}{2L} \int_c^{c+2L} f(t) e^{\frac{-i\pi n t}{L}} dt$$

e a trigonométrica:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left[a_n \cos\left(\frac{n\pi t}{L}\right) + b_n \sin\left(\frac{n\pi t}{L}\right) \right]$$

onde:

$$\begin{aligned} a_0 &= \frac{1}{L} \int_c^{c+2L} f(t) dt \\ a_n &= \frac{1}{L} \int_c^{c+2L} f(t) \cos\left(\frac{n\pi t}{L}\right) dt \\ b_n &= \frac{1}{L} \int_c^{c+2L} f(t) \sin\left(\frac{n\pi t}{L}\right) dt \end{aligned}$$

A equivalência entre ambas as formas acima se dá pela equação de Euler:

$$e^{ix} = \cos(x) + i \sin(x) \quad (1)$$

2.11 Transformada de Fourier

A transformada de Fourier sintetiza uma função por uma ponderação de senos e cossenos. Os primeiros termos da transformada são os mais significativos. Isso significa que podemos aproximar uma função tomando apenas os primeiros termos com um erro menor do que um dado épsilon positivo. A figura 6 ilustra esse fenômeno.

Como os componentes desta ponderação (senóides) são periódicos, cada um deles possui uma frequência. Por esse motivo, usalmente dizemos que a função passada pela transformada de Fourier está no domínio das frequências. Em contrapartida, a função original está no domínio do tempo (pois a princípio está descrita em função do tempo).

A versão discreta da transformada de Fourier é dada pela fórmula:

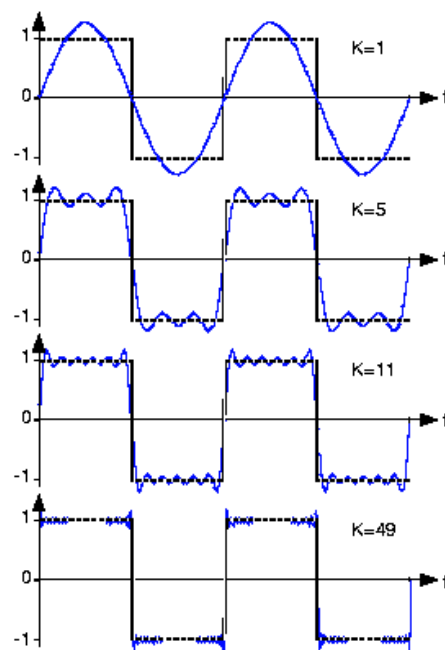


Figura 7: Aproximações de uma onda quadrada por somas de senos. No primeiro gráfico, apenas o primeiro coeficiente da transformada de Fourier é usado. Aos poucos se aumenta a quantidade de senos usada na síntese da função original e a aproximação fica cada vez melhor. Podemos notar que os últimos coeficientes melhoram apenas o nível de detalhe. Imagem extraída de <http://cnx.org/content/m12929/latest>

$$x_k = \frac{1}{n} \sum_{j=0}^{n-1} f_j e^{\frac{2\pi i}{n} jk} \quad k = 0, \dots, n-1.$$

$$f_j = \sum_{k=0}^{n-1} x_k e^{-\frac{2\pi i}{n} jk} \quad j = 0, \dots, n-1$$

Para realizar este cálculo, novamente é preciso utilizar a relação da fórmula 1 de Euler.

2.11.1 FFT

A implementação da transformada de Fourier como foi descrita gera um algoritmo quadrático. Ela pode ser implementada usando um algoritmo de complexidade $O(n \log n)$, conhecido como Transformada Rápida de Fourier (FFT - *Fast Fourier Transform*).

2.12 Curvatura

Dada a curva paramétrica $c(t) : \mathbb{N} \rightarrow \mathbb{N}^2$, $c(t) = (x(t), y(t))$, que simula o contorno da forma, uma maneira de calcular sua curvatura $k(t)$ é:

$$k(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{(x'(t)^2 + y'(t)^2)^{\frac{3}{2}}}$$

Usa-se a primeira e segunda derivadas da curva paramétrica. Mas não é simples de descobrir a curva paramétrica de um contorno. Uma outra forma de inferir a curvatura é através do curvograma (*curvegram*).

2.12.1 Curvograma

Existe uma fórmula fechada de derivação no domínio das frequências. Sendo $X(f)$ e $Y(f)$ as transformadas de Fourier de $x(t)$ e $y(t)$ respectivamente, então:

$$\begin{array}{ll}
X'(f) = i 2\pi f X(f) & Raw \\
Y'(f) = i 2\pi f Y(f) & Raw \\
X''(f) = -(2\pi f)^2 X(f) & Raw \\
Y''(f) = -(2\pi f)^2 Y(f) & Raw
\end{array}$$

Sabemos também que transformar a função para o domínio das frequências, derivar usando o método acima e executar a transformada inversa preserva a derivação. Portanto, este é o método que se aplica.

Um cuidado que deve ser tomado é que a curvatura é muito susceptível a ruídos. Portanto, é interessante desprezar os últimos coeficientes da transformada de Fourier por serem os que trazem mais detalhes, que é onde se encontram ruídos. Um método que se utiliza é, ao invés de pegar todos os coeficientes da transformada de Fourier com peso igual, multiplica-se os coeficientes por uma curva gaussiana de média zero, altura 1 e variância variável. Assim, o primeiro coeficiente (no ponto zero) é pego com peso 100

2.13 Energia de dobramento

A energia de dobramento (*bending energy*) tem origem na teoria elástica da física. É interpretada como a quantidade de energia necessária para transformar a forma numa circunferência de mesmo perímetro. Por exemplo, a figura ?? tem mais energia de dobramento que a figura ?. Esta medida depende do material (quanto mais maleável, menor é a energia de dobramento). No caso digital, a resitência do material é ignorada.

Ela é calculada a partir da curvatura multiescala. É a integral do quadrado da curvatura ao longo do contorno. Não é tão descritiva quanto a curvatura, porém é muito mais sintética (é apenas um número), sendo uma boa característica para comparar formas.

Para ser independente da escala, é preciso fazer uma normalização. Por isso, a fórmula final da energia de dobramento multi-escala $B(a)$ é:

$$B(a) = \frac{L^2}{N} \sum_{n=0}^{N-1} k(a, n)^2$$

onde L é o comprimento do perímetro e o parâmetro a é cada escalada das linhas no curvograma.



Figura 8: Uma forma com energia de dobramento mais alta

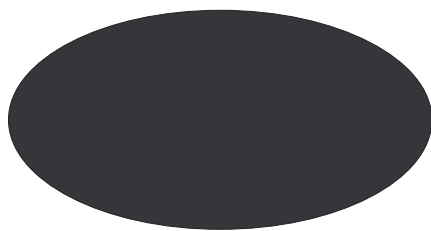


Figura 9: Uma forma com energia de dobramento mais baixa

3 Atividades realizadas

No princípio, muitos dos algoritmos já estavam prontos num *toolbox* em matlab. No final, aqueles que nos interessaram estavam implementados em java. Primeiramente fizemos apenas uma interface com um deslizador que permitisse a segmentação interativa através de limiarização. Com isso pudemos eleger a melhor dentre as formas de iluminação usadas na aquisição das imagens. Depois implementou-se a extração da área, por ser a mais simples. Ainda se usou o matlab para extrair as demais medidas. Em seguida, a extração do contorno, por servir de base para muitas outras medidas. Com isso foi possível extrair o perímetro e a referência de escala e aos poucos o *toolbox* matlab foi sendo deixado de lado. Depois foi implementada a extração de curvatura e outras medidas. A ferramenta agora está totalmente auto-contida.

4 Resultados

5 Conclusão

Parte subjetiva

Lista das disciplinas cursadas no BCC mais relevantes para o trabalho:

- Disciplinas de introdução à programação
- Introdução à Probabilidade e à Estatística I
- Álgebra Linear para Computação
- Análise de Algoritmos
- Engenharia de Software
- Programação Orientada a Objetos
- Análise e Reconhecimento de Formas: Teoria e Prática
- Laboratório de Programação Extrema

Referências

Costa, Luciano da F. e Roberto M. Cesar Jr.

Shape analysis and classification: theory and practice, CRC Press, 2001.

http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/boundary-rep-desc.pdf

http://pt.wikipedia.org/wiki/S%C3%A9rie_de_Fourier

http://pt.wikipedia.org/wiki/Transformada_de_Fourier