Uma ferramenta morfométrica
MAC 499 - Trabalho de formatura supervisionado

aluno: Heitor Lellis Nicoliello 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 . . . 5
2.2 Vizinhança entre pontos . . . 5
2.3 Segmentação . . . 5
2.3.1 *Thresholding* . . . 6
2.4 Área . . . 7
2.5 Contorno . . . 7
2.6 Perímetro e maior diâmetro . . . 10
2.7 Centróide . . . 12
2.8 Raio médio . . . 12
2.9 Assimetria . . . 12
2.10 Proporção de escala . . . 13
2.11 Série de Fourier . . . 13
2.12 Transformada de Fourier . . . 14
2.12.1 FFT . . . 16
2.13 Curvatura . . . 16
2.13.1 Curvograma . . . 16
2.14 Energia de dobramento . . . 17

3 Atividades realizadas **18**

4 Resultados - A ferramenta **18**
4.1 Detalhes de implementação 20

5 Conclusão **20**

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, 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 proposto pelo mestrando Fabio Machado 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 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 ser tipicamente divididas em três grandes etapas: pré-processamento da forma, transformações de forma e classificação da forma, como mostra a figura 1. O foco 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

Trata-se da etapa na qual se insere este trabalho, ou seja, o momento em que é feita a extração de características da forma, quando se obtém informações abstratas sobre a imagem.

1.5 Classificação

Trata-se da separação das formas em classes, baseando-se nas descrições obtidas na etapa de transformações. Neste processo, existem 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.

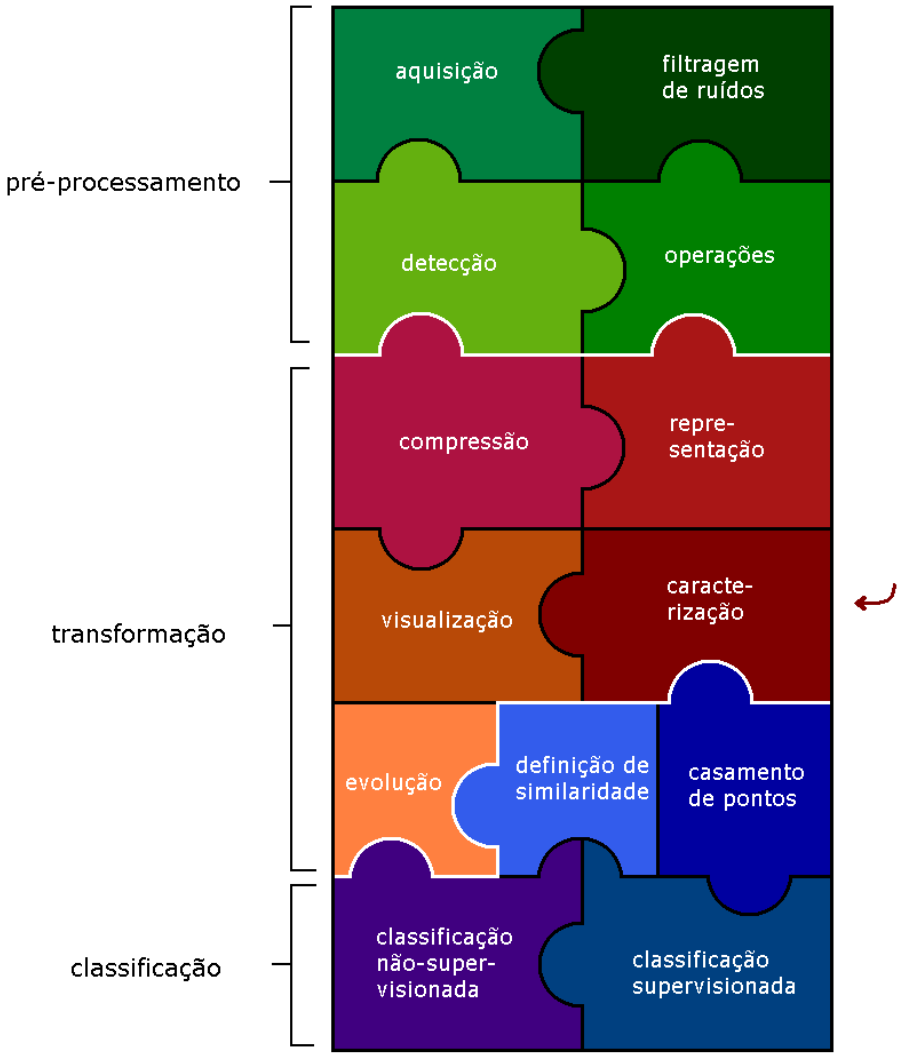

Figura 1: Tarefas típicas do processo de análise de imagens. Figura extraída de [1]

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 para *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 representar uma imagem digital é por meio de níveis de cinza. Uma matriz de inteiros variando entre zero e 255 representa os tons de cinza da imagem: quanto mais baixo for o valor de um *pixel*, mais escura é sua tonalidade.

Um terceiro tipo de imagem digital é a binária, que só possui valores zero ou um (usalmente 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.

2.2 Vizinhança entre pontos

Existe mais de uma maneira de decidir se dois pontos são vizinhos ou não. Duas delas são definidas as seguir. O resultado pode ser visualizado na figura 2.

- 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 separação entre o objeto de interesse e o fundo na imagem. É um procedimento que recebe uma imagem colorida ou em níveis de cinza e

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

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 como limiarização, crescimento de regiões, redes neurais, métodos estatísticos, transformada de Fourier, transformada *wavelet*, transformação *Watershed* e 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 freqüência de intensidades da imagem; ou pode ser pré-definido caso as imagens a serem processadas sigam um padrão conhecido. Esta técnica de binarização se baseia na hipótese de que os níveis de intesidade do objeto e do fundo são diferentes e, portanto, existe um valor capaz de separá-los.

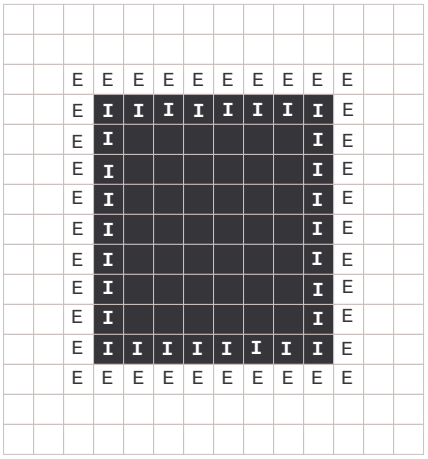

Figura 3: Exemplo de contornos interno (I) e externo (E) de uma forma discreta.

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 varia a relação entre *pixel* e milímetros. Só pode ser utilizada como 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 3 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 seus 8-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.

Apresentamos abaixo um procedimento para extração do contorno de uma forma em sentido anti-horário.

```
// 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 = cadidato;
         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;
      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
```

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 4, onde há uma reentrância de largura um pixel.

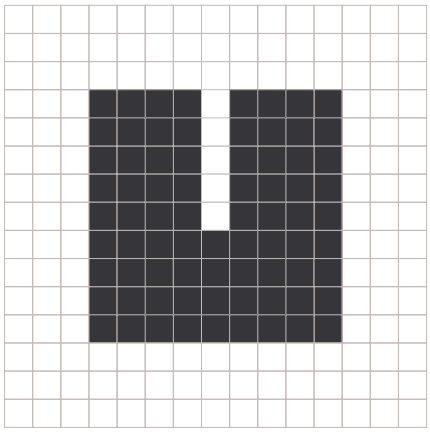

Figura 4: Exemplo de imagem binária com reentrância de um *pixel* de largura onde o trajeto do contorno deve entrar e sair dela.

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 5, onde há mais de um caminho a seguir. Podemos 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.

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 6 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.

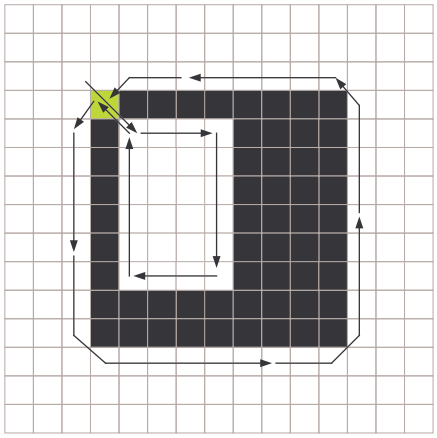

Figura 5: Exemplo de imagem binária em que o contorno deve seguir também pela parte "interna" da forma. O *pixel* destacado é o primeiro *pixel* do contorno.

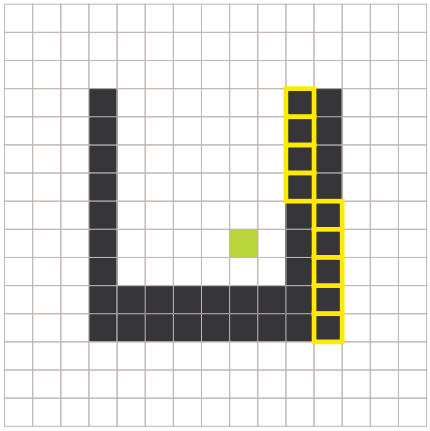

Figura 6: Uma imagem binária com o maior diâmetro destacado em amarelo e seu centróide marcado em verde.

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 6.

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 Assimetria

Para calcular alguma medida de assimetria de uma forma, é necessário antes definir o eixo de assimetria. Uma possibilidade é usar um dos eixos principais da forma, diretamente relacionados aos vetores próprios (ou autovetores) de uma matriz associada à imagem binária. Para calculá-los, devemos guardar as coordenadas dos pontos da imagem em dois vetores x e y. Em seguida, calculamos a matriz de covariância desses dois vetores, encarando cada ponto como um ensaio de duas variáveis aleatórias. Sendo X e Y variáveis aleatórias associadas, respectivamente, aos valores de x e de y, a matriz de covariância é definida da seguinte forma:

$$matriz\,de\,covariância\,(X,Y) = \begin{bmatrix} var(X) & cov(X,Y) \\ cov(X,Y) & var(Y) \end{bmatrix}$$

Onde:

$$var(X) = E(X^2) - E^2(X)$$
$$cov(X,Y) = E(X.Y) - E(X).E(Y)$$
$$E(X) = \sum_{i=1}^{n} x_i.P(X_i = x_i)$$

Tendo esta matriz calculada, basta calcular seus autovetores e têm-se os vetores diretores dos eixos principais. Um ponto que pertence a ambos os

eixos é o centróide. Com isso podemos definir as retas às quais pertencem os eixos principais da forma.

Uma medida de assimetria pode ser a diferença entre as áreas de cada uma das partes da figura separadas por um dos eixos.

2.10 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 tanto, 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 prorpoção (raio médio/raio real da referência). Assim teremos a relação *pixels*/milímetros.

2.11 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:

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

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:

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

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

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

2.12 Transformada de Fourier

A transformada de Fourier sintetiza uma função por uma ponderação de senos e cossenos. Normalmente, em análise de formas, 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 7 ilustra esse fenômeno.

Como os componentes desta ponderação (senóides) são periódicos, cada um deles possui uma freqüência. Por esse motivo, usualmente dizemos que a função passada pela transformada de Fourier está no domínio das freqüências. 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:

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

onde:

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

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

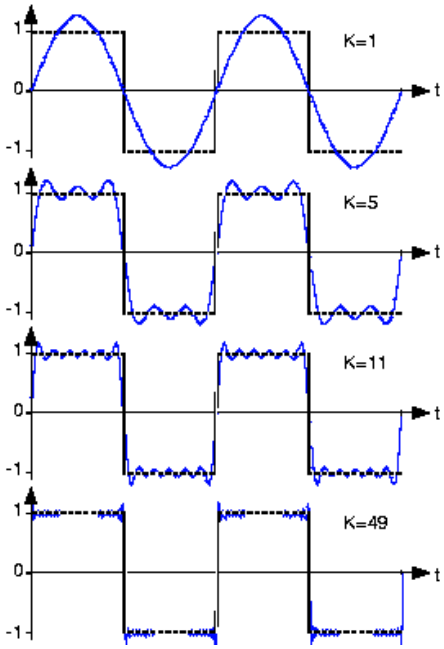

Figura 7: Aproximações de uma onda quadrada por somas de senos. No primeiro gráfico, apenas o primeiro coeficiente da transformada de Fourier é usado. Aos poucos se aumenta a quantidade de senos usada na síntese da função original e a aproximação fica cada vez melhor. Podemos notar que os primeiros coeficientes definem o tom geral das baixas freqüências da função original e os últimos coeficientes melhoram apenas o nível de detalhe. Imagem extraída de [2]

2.12.1 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*).

2.13 Curvatura

Dada a curva paramétrica $c(t) = (x(t), y(t))$, que simula o contorno da forma em função do tempo, 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 a segunda derivadas da curva paramétrica. O curvograma (*curvegram*) apresenta uma medida multi-escala da curvatura.

2.13.1 Curvograma

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

$$\begin{aligned} X'(f) &= i2\pi f\, X(f) \\ Y'(f) &= i2\pi f\, Y(f) \\ X''(f) &= -(2\pi f)^2 X(f) \\ Y''(f) &= -(2\pi f)^2 Y(f) \end{aligned}$$

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. O que se faz é, ao invés de tomar todos os coeficientes da transformada de Fourier com peso igual, multiplica-se os coeficientes por uma curva gaussiana de média zero, altura 1 e escala variável. Assim, o primeiro coeficiente (no ponto zero) é tomado com peso 100%. Os coeficientes na seqüência têm um peso menor, mas ainda sim relevantes, e os pesos vão diminuindo até que os coeficientes mais altos influenciam muito menos no

Figura 8: Uma forma com energia de dobramento mais alta

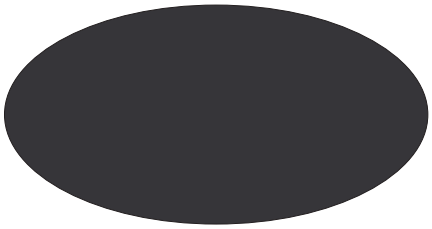

Figura 9: Uma forma com energia de dobramento mais baixa

cálculo da curvatura. Esse procedimento suaviza a curva. Como a variância da curva de Gauss não é pré-definida, introduzimos uma nova variável. Por isso este método é chamado de multi-escala.

2.14 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 8 tem mais energia de dobramento que a figura 9. Esta medida depende do material (quanto mais maleável, menor é a energia de dobramento). No caso digital, a resistência do material é ignorada.

Ela é calculada a partir da curvatura multi-escala. É a integral do quadrado da curvatura ao longo do contorno. Não é tão descritiva quanto a curvatura, porém é muito mais sintética por ser 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:

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

em que L é o comprimento do perímetro, $k(a,n)$ é a curvatura no ponto a com a escala n.

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 forma de iluminação usadas na aquisição das imagens que proporcionasse melhores segmentações. 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 - A ferramenta

O resultado deste trabalho é um *software* escrito em java que serve de ferramenta para extrair automaticamente todas as medidas aqui detalhadas. A figura 10 exibe algumas capturas de tela do programa em funcionamento. O usuário deve abrir uma imagem colorida em formato *jpeg* seguindo o protocolo que foi definido: fundo azul, ovo no centro e círculo vermelho de raio 13 milímetros ao lado. Então, o usuário usa a barra vertical para escolher o limiar para segmentação e obtém as medidas através do menu “Ferramentas”.

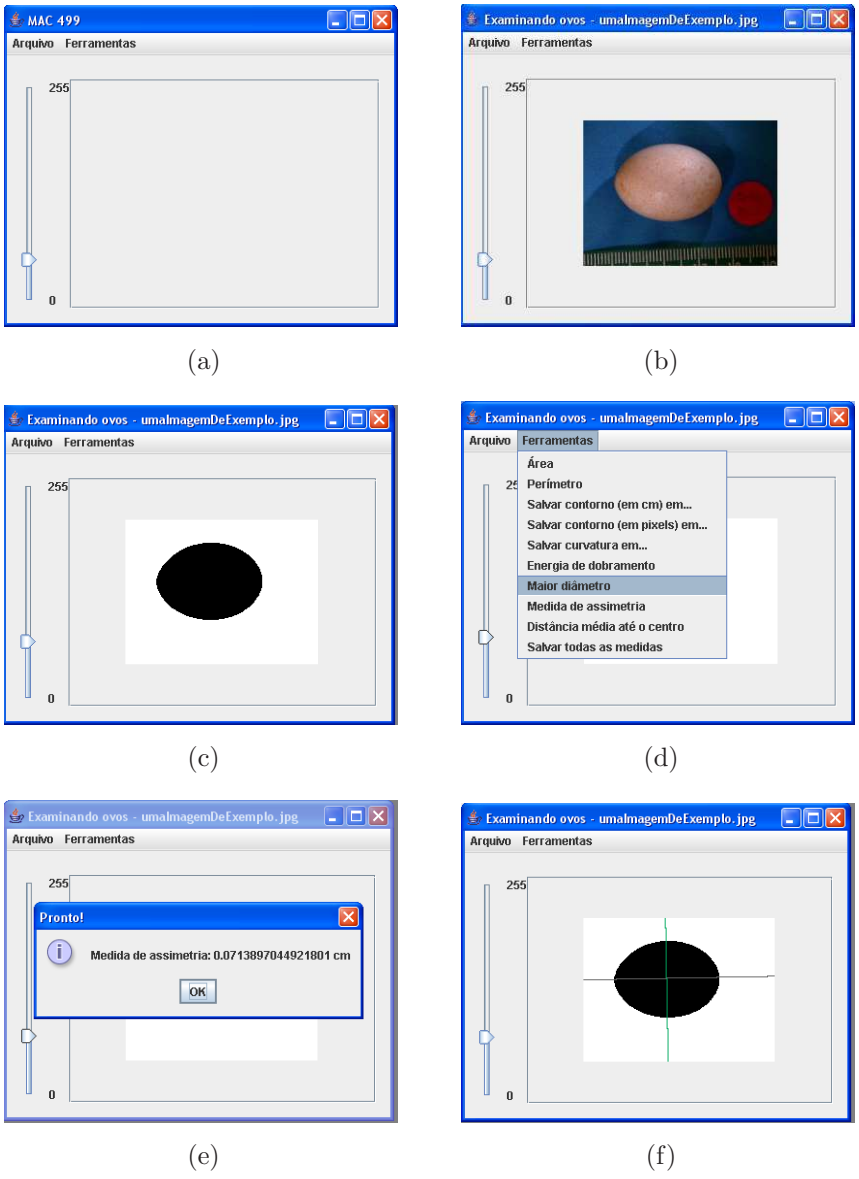

Figura 10: Capturas de tela do *software* em ordem temporal de um caso de uso: estado inicial do programa (a), imagem aberta (b), binarizada (c), pedindo para se extrair medida de assimetria (d), medida calculada e informada na tela (e) e eixos principais do objeto desenhados na tela (f).

4.1 Detalhes de implementação

Procuramos desenhar um esquema em que fosse fácil realizar manutenção; inserir a extração de novas medidas; e modularizar os algoritmos de processamento de imagens em relação às rotinas de interface. A classe abstrata Medida contribui com estes objetivos quando se quer inserir a extração de uma nova medida e força a manter a organização existente. Além disso, absolutamente todos os algoritmos de processamento de imagens se encontram nas classes PreProcessador e ExtratorDeMedidas, totalmente independentes da implementação do programa, podendo ser exportadas como biblioteca para ser usada em outros sistemas. Juntamente com este documento, segue o código-fonte do *software*.

5 Conclusão

A ferramenta criada possibilita a extração de características de imagens de ovos para estudos morfométricos, o que seria impossível sem o uso de imagens digitais e do computador e provavelmente inviável sem uma ferramenta customizada para este fim. Com a funcionalidade de salvar todas as medidas de uma só vez em um arquivo, é possível utilizar tais medidas em pacotes estatísticos para aplicação de algoritmos de agrupamento, escolha de características mais representativas, etc. Futuramente, gostaríamos de estender a ferramenta para que atendesse a uma gama maior de estudos morfométricos, possibilitando que o usuário informe a cor, a forma e o tamanho do objeto de referência de escala.

Parte subjetiva

Acredito que todas as disciplinas do primeiro ano foram importantes para este trabalho, pois nelas foi que aprendi a pensar abstratamente e a programar, o que é fundamental para este tipo de trabalho.

Um dos desafios que senti logo que entrei no curso foi aprender a programar. Eu nunca havia tomado contato com programação e tinha a impressão de que a maioria dos meus colegas já havia o feito. Porém, hoje penso que não é ruim ingressar neste curso sem saber programar, pois, assim, há menos chances de aprender a programar mal. Vale citar que fui monitor de disciplinas quatro vezes e em todas aprendi muito. As disciplinas de estatística

também foram importantes para, especificamente, o cálculo de medida de assimetria.

Outra disciplina importante foi análise de algoritmos, pois as imagens com que trabalhamos são muito grandes (com as quais é fácil estourar o *heap space* do Java) e há algoritmos cuja primeira idéia nem sempre é a mais adequada por questão de espaço ou de tempo. Saber analisar os algoritmos é fundamental para otimizá-los.

Engenharia de *software* e Laboratório de Programação Extrema forneceram experiências com desenvolvimento de projetos e Programação Orientada a Objetos ajudou a elaborar o modelo do sistema. Tive muita dificuldade ao trabalhar com interface gráfica por não ter quase nenhuma experiência prévia no assunto. Iniciar um sistema sozinho forçou-me a desenhar o código orientado a objetos por conta própria. Essa foi uma ótima experiência já que é freqüente usarmos as tecnologias e arquiteturas passadas pelo professor ou, às vezes, o trabalho não é complicado ou duradouro o suficiente a ponto de exigir tal elaboração.

Por fim, a disciplina mais diretamente relacionada a este trabalho que cursei foi Análise e Reconhecimento de Formas: Teoria e Prática, cujo professor foi meu orientador, ao qual sou muito grato pela orientação.

Infelizmente não me dediquei desde o início a este trabalho tanto quanto ele merecia, principalmente por estar estagiando. A experiência do estágio foi enriquecedora, porém ela podia esperar. Por isso, se eu pudesse aconselhar os calouros, diria que só trabalhassem se fosse necessário. Senão, que se dedicassem ao curso de ciência da computação.

Finalmente, este trabalho serviu de grande aprendizado para mim. Tive a oportunidade de conhecer um pouco mais sobre como funciona a pesquisa acadêmica no IME e desde então me agrada a idéia de seguir carreira acadêmica.

Referências

[1] Costa, Luciano da F. e Roberto M. Cesar Jr.
"Shape analysis and classification: theory and practice",
CRC Press, 2001.

[2] http://cnx.org/content/m12929/latest

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

Todas as páginas acessadas em 19/02/2008.