

Processamento de imagem biomédica

Estudo das dimensões da cabeça de um feto em
Matlab



Autores: Ricardo Carvas @14812;
Luís Pereira @14868;

Orientador Científico: Prof. José Henrique Brito



Índice

Introdução.....	3
Manual de utilizador	4
Ecrã inicial	4
Recorte da imagem	4
Segmentação da imagem	5
Cálculo do perímetro.....	6
Scripts.....	7
Conclusão.....	10
Referências.....	10



Introdução

A área de estudo do processamento de imagem tem como objetivo a decomposição da imagem em componentes fundamentais com o propósito de extrair informações significativas. Algumas das tarefas incluem: encontrar formas, detectar bordas, remover ruídos, contar objetos e calcular estatísticas para análise de textura ou qualidade de imagem.

Neste trabalho, como objeto de avaliação à UC, foi realizada uma solução no Matlab para calcular o perímetro da cabeça de um feto através da sua ecografia

O objetivo do projeto divide-se em três partes:

- Segmentação da cabeça;
- Cálculo do seu perímetro;
- Exibição da segmentação e perímetro.

O perímetro será então calculado através da obtenção do número de pixels na linha de contorno da cabeça com a escala 25pixel/cm.



Figura 1- Perímetro craniano, retrieved from: institutopensi.org.br



Manual de utilizador

Ecrã inicial

Ao iniciar a aplicação, será apresentado ao utilizador uma mensagem para seleccionar uma das imagens para ser processada. O ecrã inicial apresenta também cinco botões: Seleccionar imagem, Recortar, Calcular Contorno, Calcular Perímetro e Reset. Para prevenir erros de utilização, os botões desativados vão ficando disponíveis à medida que o utilizador interage com a aplicação.



Figura 2-Ecrã inicial

Recorte da imagem

Após ter seleccionado uma imagem, o utilizador deverá fazer uso do clique do rato para desenhar um polígono ponto a ponto em volta da área que pretende analisar para proceder ao seu recorte. Quando o polígono de recorte estiver pronto, o utilizador deve clicar no botão “Recortar”.

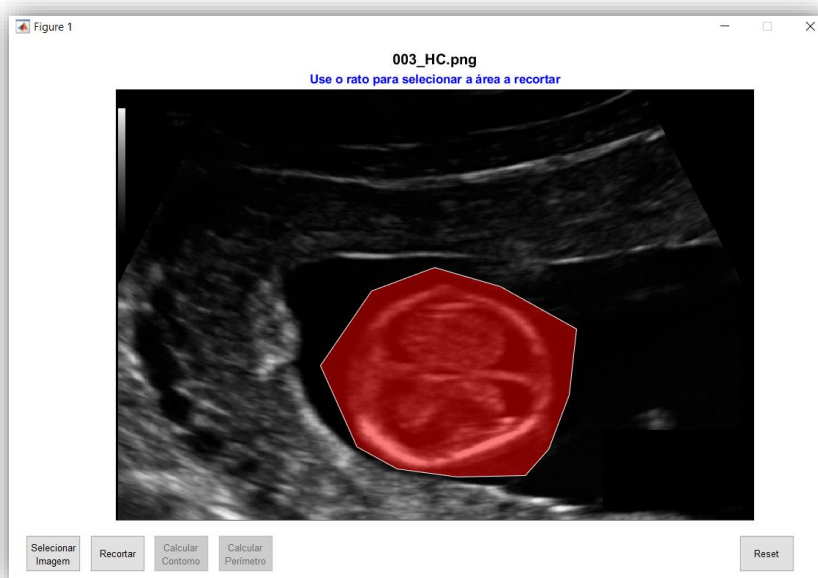


Figura 3-Recorte da imagem



Segmentação da imagem

Ao clicar no botão “Calcular contorno”, a aplicação irá exibir o resultado de todas as etapas de processamento que a aplicação está a realizar em tempo real. (Theresholding, Filling, Activecontour e Outline final).

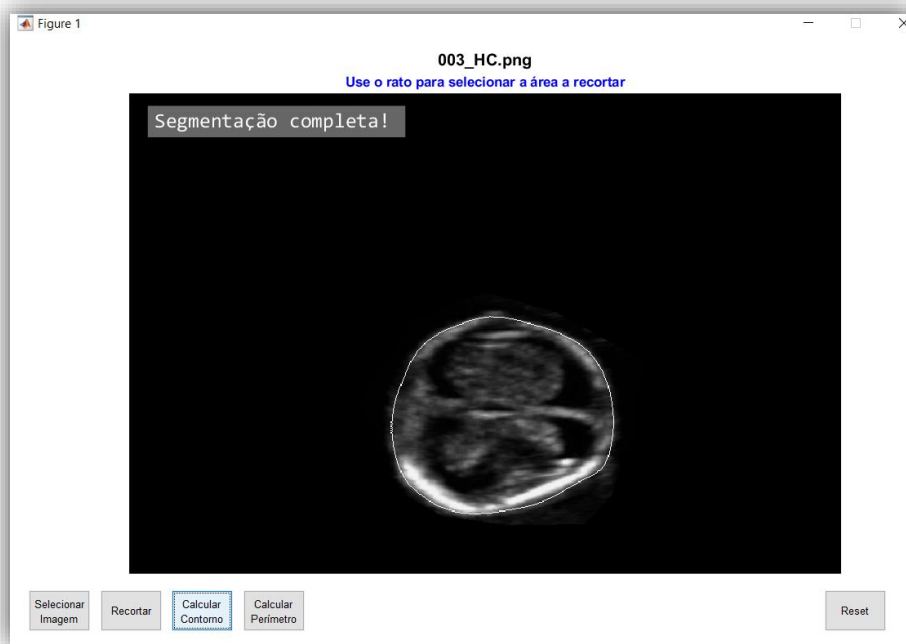
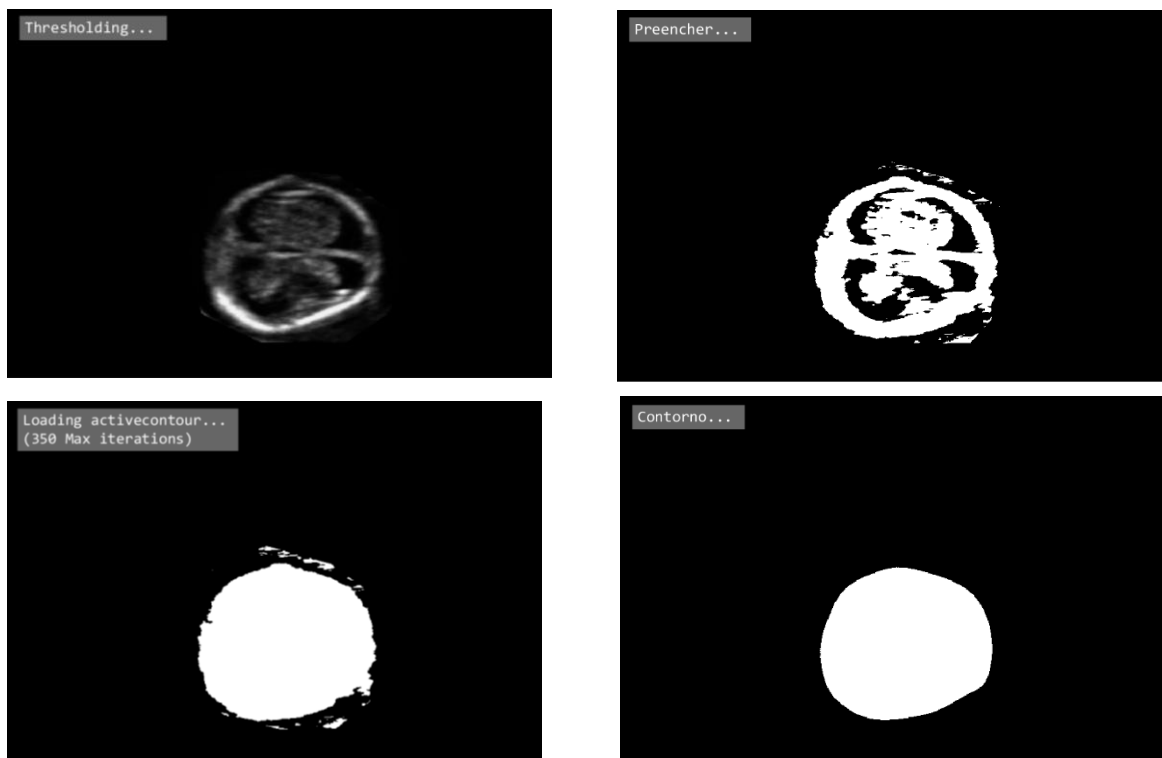


Figura 4-Etapas de Segmentação



Cálculo do perímetro

Para concluir o processo, o utilizador deve carregar no botão 'Calcular Perímetro' e será apresentado na tela o número de pixéis brancos encontrados e o devido perímetro da cabeça.

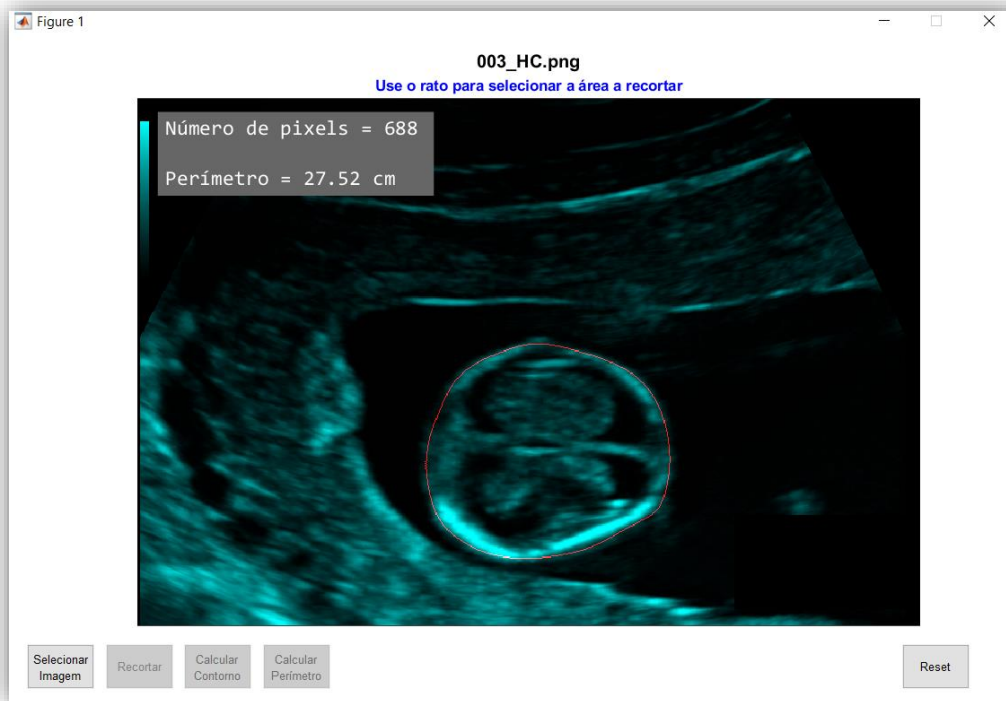


Figura 5-Apresentação do perímetro



Scripts

Main.m: Script responsável pela declaração da figura e botões que permitem o interface gráfico da aplicação. É este o script a ser corrido.

Img.m: Script responsável pela seleção de imagem abrindo uma *dialog box* através da função [*"uigetfile"*](#) que permite ao utilizador selecionar uma imagem em formato *.png* do seu disco.

mouseButton.m: Script utilizado nas aulas fornecido pelo docente que permite a criação do polígono de recorte. Este script cria um array com as coordenadas x,y onde o utilizador clicou com o rato a primeira vez e vai adicionando as coordenadas de cada novo clique do utilizador ao mesmo tempo que desenha o polígono correspondente à ligação com cada nova coordenada.

ButtonCortar.m: Script utilizado nas aulas fornecido pelo docente que permite fazer o recorte da imagem com base no polígono desenhado com o rato.

ButtonSeg.m: Script responsável por todo o processamento da zona de interesse da imagem (cabeça do feto) recortada previamente. O objetivo deste processamento será então encontrar a linha de contorno da cabeça e obter o seu número de pixéis.

Dentro deste script foram realizadas 4 etapas para permitir o desenho do contorno da cabeça (ver figura 3):

1. **Thresholding:** Esta etapa tem como objetivo binarizar a imagem, de forma a ser possível distinguir áreas vazias e áreas com objetos. Isto permite a criação de uma imagem onde a cabeça e o background são facilmente distinguidos pela cor branco e preto, respetivamente.

Como a imagem da cabeça se trata de uma ecografia, isto permite que seja possível utilizar a função [*"imbinarize"*](#) para realizar o thresholding, visto se tratar de uma imagem em tons de cinzento, que é requisito desta função.

A função foi utilizada com a seguinte assinatura:

```
binImg = imbinarize(cutImg, 'adaptive', 'Sensitivity', 1);
```

Esta função cria uma imagem binária fazendo o thresholding com base no valor de sensibilidade passado como parâmetro na assinatura da função. Para este trabalho entendemos que o valor máximo de sensibilidade (1) fornece os melhores resultados para as 3 imagens em estudo.



Figura 7-Etapa 1 output

2. **Preenchimento:** Devido à existência de algumas partes menos densas no interior da cabeça do feto, na imagem binarizada o interior da cabeça ainda apresenta alguns conjuntos de pixéis a preto, isto devido à atenuação feita pelos ultrassons não conseguir distinguir essas partes do background, logo, ao realizar o thresholding, essas áreas serão identificadas com pixel preto.

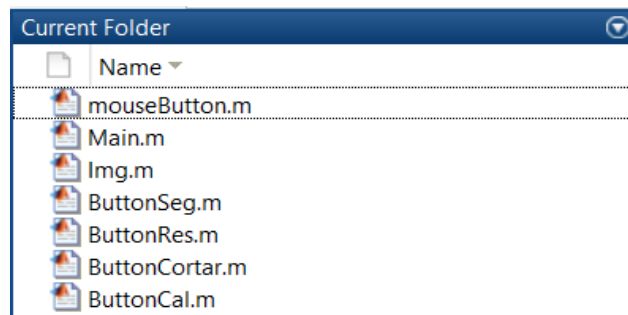


Figura 6-Scripts utilizados



Em ordem de uniformizar a cabeça para realizar a etapa seguinte, foi utilizada a função `"imfill"` com a seguinte assinatura:

```
fillImg = imfill(binImg, 'holes');
```

Deste modo, todas as manchas a preto do interior da cabeça serão passadas a branco, formando assim uma imagem da área da cabeça uniforme.



Figura 8-Etapa 2 output

3. **Activecontour:** Para eliminar o ruído à volta da cabeça e desenhar o seu contorno de uma forma regular e suave, foi utilizado a função `"activecontour"` com a seguinte assinatura:

```
it=350;  
ctrImg = activecontour(I, fillImg, it, 'edge',  
    'SmoothFactor', 15);
```

Esta função segmenta a imagem em foreground e background utilizando a técnica de *region growing active contour*.

Neste script, a função recebeu como parâmetro a imagem `I` (imagem original da ecografia) e a máscara `fillImg`, (imagem binária obtida na etapa 2). Isto indica à função o estado inicial do *active contour*. Os contornos do objeto a branco da máscara definem a posição inicial da evolução do contorno para segmentar a imagem `I`. O output será uma imagem binária com a parte da cabeça a branco e o background a preto, onde o contorno da cabeça foi atenuado consoante o número de iterações da função e o valor de *SmoothFactor*, valor esse que representa a suavidade da linha de contorno.

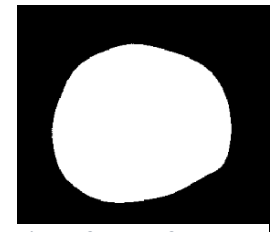


Figura 9-Etapa 3 output

Para este projeto, entendemos que 350 seria um bom número de iterações máximas com resultados positivos.

4. **Outline:** Para conseguirmos calcular o perímetro craniano, é necessário fazer a contagem dos pixels brancos da linha de contorno da cabeça. Para isso temos de primeiro passar o interior da cabeça para preto e deixar apenas a branco o seu contorno. Para esta tarefa utilizamos a função `"bwperim"` com a seguinte assinatura.

```
segImg = bwperim(ctrImg, 4);
```

Com esta função, ao passarmos por parâmetro a imagem binária obtida na etapa 3, (`ctrImg`), temos como output a imagem binária com apenas o contorno da cabeça a branco. O segundo parâmetro corresponde à conectividade dos pixels, onde temos duas opções possíveis: 4 ou 8.

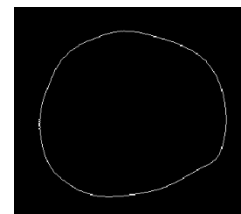
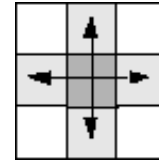


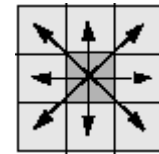
Figura 10-Etapa 4 output



4 indica que dois pixéis estão conectados apenas se as suas pontas tocarem.



8 indica que dois pixéis estão conectados se as suas pontas ou cantos tocarem.



Uma das dúvidas que surgiu na realização deste projeto foi o tipo de conectividade a usar.

Após serem testados os dois tipos, percebemos que ambos produzem outlines idênticas, porém, ao fazermos a contagem dos pixéis vemos que a contagem é significativamente maior na conectividade 8. Assim sendo, entendemos que como a linha de contorno é idêntica, os pixéis contados a mais na conectividade 8 são desprezáveis e por isso optámos por seguir pela conectividade default da função, 4.

Fica assim concluído o processo de segmentação, onde o próximo passo é então calcular o perímetro com base no número de pixéis brancos da imagem binária obtida na etapa final da segmentação.

ButtonCal.m: Script responsável pelo cálculo do perímetro craniano com base na contagem dos pixéis da outline da cabeça obtida na fase de segmentação.

Como a imagem da outline da cabeça corresponde a uma matriz binária, onde 1 corresponde a branco e 0 corresponde a preto, para fazer esta contagem foi utilizada a função “[nnz](#)”, que retorna o número de pixéis não zeros de uma dada matriz.

```
numPixels = nnz(segImg);
```

O perímetro é então calculado usando a escala 25pixel/cm.

```
Per = numPixels / 25;
```

O script exibe também no interface gráfico a imagem original da ecografia com a linha de contorno da cabeça calculada sobreposta, bem como a anotação do seu número de pixéis e perímetro em cm.

ButtonRes.m: Script que permite voltar à imagem seleccionada original e apagar todos os procedimentos feitos.



Conclusão

Ao longo deste trabalho fomos lidando com algumas dificuldades e inconvenientes onde a pesquisa e a autoaprendizagem foram chave para a sua realização bem-sucedida.

Apesar de não sermos estranhos ao Matlab, é sempre necessária uma autoavaliação dos conhecimentos e perceber onde temos de explorar e melhorar em ordem a conseguir uma boa solução para o problema apresentado.

A maior dúvida deste trabalho foi sempre entender quais os métodos e funções/algoritmos a usar, como os usar e por que ordem os usar, de forma a tornar a solução eficaz, eficiente e a produzir o melhor resultado possível.

Outra preocupação foi também elaborar um interface gráfico simples e eficiente, onde se tentou evitar os cenários de má utilização, bem como, exibir ao utilizador o resultado das etapas de processamento que estão a decorrer em tempo real.

Assim sendo e agora após a sua realização, podemos com satisfação dizer que damos os objetivos a que nos propusemos como alcançados e as dificuldades como superadas.

Referências

Image Processing ToolboxDocumentation. (2020). Retrieved from <https://www.mathworks.com/help/images/index.html>

