

Um sistema para melhoria de mamogramas utilizando técnicas de processamento de imagens regionais

Felipe Aguiar, Felipe Jaber Castro, Aura Conci e Diego Dacal

Instituto de Computação, UFF

24210-240, Niterói, RJ

{fcastro, ddacal, faguiar, aconci}@ic.uff.br

Resumo: *Este trabalho mostra que todas as técnicas originalmente propostas para aplicação em uma imagem inteira podem ser também usadas para melhoria de mamogramas regionalmente com mais vantagens. Essas regiões podem ser definidas por simples subdivisão da imagem em janelas de tamanhos constantes como também podem ser determinadas de maneira adaptativa pela observação de algum parâmetro significativo como as intensidades médias ou a variância das regiões da imagem. O sistema aqui apresentado foi todo desenvolvido a partir de trabalhos do curso de Análise de Imagens da UFF, sem o uso de qualquer ferramenta adicional à linguagem Pascal, sendo de utilização simples e intuitiva; seu código fonte é aberto e disponibilizável pelos autores. A estrutura das rotinas baseadas em regiões é, em termos de processamento de imagens, idêntica às convencionais. Apresentamos trechos de códigos em Delphi das diversas técnicas para melhor ilustrar aspectos da implementação. Os testes realizados em imagens de mamografia indicam que com a utilização adequada dos parâmetros na técnica de unsharp masking propiciou os melhores resultados.*

Palavras-chave: processamento de imagens, mamografia, *windowing*.

Introdução

O nódulo mamário e a própria mama têm uma estrutura muito complexa, o que torna a análise dos mamogramas uma tarefa muito difícil e especializada. Podem existir nódulos com alterações de tecido quase imperceptíveis ao olho humano, e outras muito nítidas. A variação do tecido, a superposição destas estruturas na imagem e, às vezes, o não desenvolvimento aparente da forma do nódulo, dificulta o seu diagnóstico. De um modo geral, é possível caracterizar um nódulo como maligno ou benigno através da análise da sua textura ou rugosidade. Se no nódulo existir textura do tipo pipoca, laminar concêntrica, difusa ou central,

provavelmente será benigno. Entretanto, se o nódulo não tiver forma definida, possuir áreas de necrose, e for formado por ramificações invasoras aos tecidos adjacentes provavelmente será maligno [2]. A geometria é outra maneira de se caracterizar um nódulo como maligno ou benigno. Se o nódulo for arredondado ou de contorno bem definido, provavelmente é benigno. Se for espicular ou de contorno mal definido, provavelmente é maligno [3]. O diagnóstico precoce de nódulos possibilita uma maior chance de cura e uma sobrevivência ao paciente. Porém, algumas vezes, o diagnóstico pode ser muito difícil, o que pode levar a procedimentos desnecessários, e a perda de tempo causando até a morte do paciente. Para amenizar estes casos têm sido preconizado nas últimas décadas a realização de exames frequentes e o uso de ferramentas computadorizada que segmentem automaticamente o nódulo em exame e auxiliem na sua visualização para caracterizar suas estruturas internas.

Há no mercado diversos produtos que oferecem um sistema para identificação de tumores e melhorias de mamogramas digitais, desde a fase de aquisição, entretanto estes são geralmente complexos e associados aos equipamentos de exames. Isso os inviabiliza para serem usados nos consultórios médicos e abranger uma fatia de usuários maior, que precisam de sistemas com abordagem mais simples e possíveis de serem usados em microcomputadores comuns de consultório [5].

Este trabalho apresenta um sistema produzido com o intuito de auxiliar médicos na detecção de tumores de mama através de uma interface de uso simples, não dispendiosa e eficiente. As imagens mamográficas são processadas para uma melhor visualização do médico e uma melhor análise do próprio mamograma, para depois definição, caso o nódulo exista, se é benigno ou maligno.

O objetivo deste trabalho é apresentar um conjunto de técnicas que foram desenvolvidas, testadas com sucesso na melhoria de imagens radiológicas e que foram agrupadas em um programa único. O código deste é todo aberto e estruturado de maneira simples, podendo ser entendido e estendido por qualquer programador com uma experiência mínima e com conhecimento de alguns conceitos de análise de imagens. O sistema foi todo desenvolvido a partir de trabalhos do curso de Análise de Imagens da UFF (<http://www.ic.uff.br/~aconci/AI.html>)

sem o uso de qualquer ferramenta adicional à linguagem Pascal, sendo de utilização simples e intuitiva.

Nas próximas seções, após esta breve introdução a cada técnica, que foi incluída no sistema, é mostrado a solução implementada relativa a mesma e apresentados os resultados experimentais obtidos. Na última seção, possibilidades de melhorias são consideradas.

Modificação em pixels das imagens

Tradicionalmente, as transformações nas imagens digitais podem ser feitas localmente pixel a pixel, ou por regiões da imagem através da modificação de determinada vizinhança de um pixel. Mas ambas são aplicadas em toda a imagem em consideração. No entanto, essas análises além de consumirem muito tempo de processamento nunca apresentam bons resultados quando um parâmetro único (por exemplo o limiar de binarização, ou o tipo de filtro) deve ser utilizado. Logo, sistemas regionais de transformação por divisões da imagens em partes menores com maior chance de características em comuns é uma ótima forma de melhorar sempre qualquer técnica. Essa alternativa têm grande relevância mas está sendo cada vez menos utilizada em detrimento de desenvolvidos muito mais dispendiosos e complexos [7].

Geralmente, o sistema automático mais usado de melhoria da imagem localmente pixel a pixel se baseia na observação do Histograma de imagem, e na mortificação dos tons dos pixels para aumentar o contraste geral da imagem. O Histograma é uma ferramenta que tem como utilidade mostrar os níveis (quantos pixels há) de cada cor na área em análise. O eixo horizontal representa a intensidade de cor (variando de 0 à 255), e o eixo vertical representa a quantidade de pixels que tem aquele valor de intensidade de cor. A figura 1 mostra um exemplo dos histogramas de cada canal R G B de uma imagem colorida, na forma como é apresentado pelo sistema desenvolvido.

Se as imagens forem monocromáticas cada canal contribui da mesma maneira na formação dos tons de cinza e apresentam histograma de mesma forma. Na implementação desenvolvida o histograma é calculado e apresentado na subrotina **Unthistograma**. Esta basicamente verifica se o usuário selecionou o **radio button Vermelho**, **Verde** ou **Azul** e a partir daí percorre cada pixel da imagem contando quantas vezes apareceu cada intensidade. O código em Delphi que faz essa verificação de maneira mais eficiente é o mostrado na tabela 1 no Apêndice no fim deste trabalho.

No início do desenvolvimento, foi utilizada a forma usual da linguagem de acesso aos valores dos pixels, mas essa solução é muito dispendiosa.

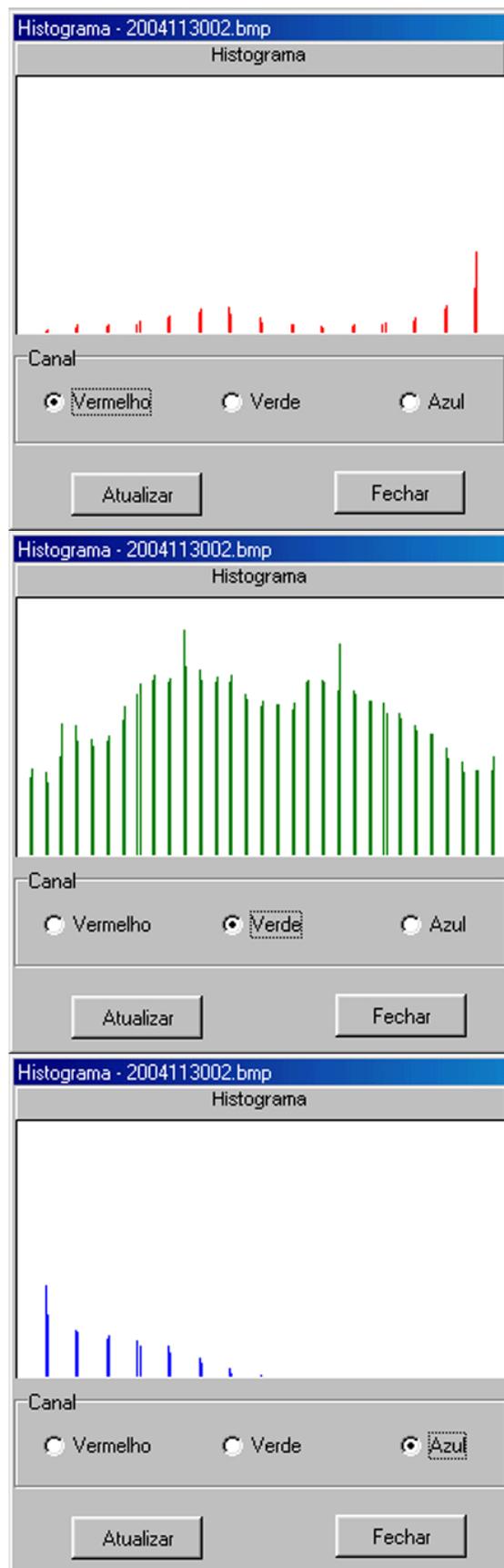


Figura 1: Exemplo de Histogramas RGB de uma Imagem colorida disponível no sistema.

Optou-se por desenvolver uma solução própria em Delphi. Explicando melhor, o maior gasto de tempo é encontrar o valor inteiro que representa cada banda de cor, já que o retorno do pixel, é na forma de um inteiro longo. Esse acesso a cor de cada pixel é feito diversas vezes nas diversas rotinas executadas pelo programa. Para superar esse problema desenvolvemos o código mostrado na tabela 2. Neste código o retorno de cada linha é do tipo inteiro, o que tornará muito mais fácil e rápida a utilização da intensidade de cada banda de cor dentro do programa.

Para um sistema de verificação automática de nódulos a primeira melhoria eficiente das imagens é recuperar os contrastes ótimos das regiões com amplitude reduzidas [2]. Alguns histogramas (como o mostrado na Figura 2) têm uma concentração de pixels muito grande em uma determinada área, isso faz com que seu contraste seja muito pequeno.

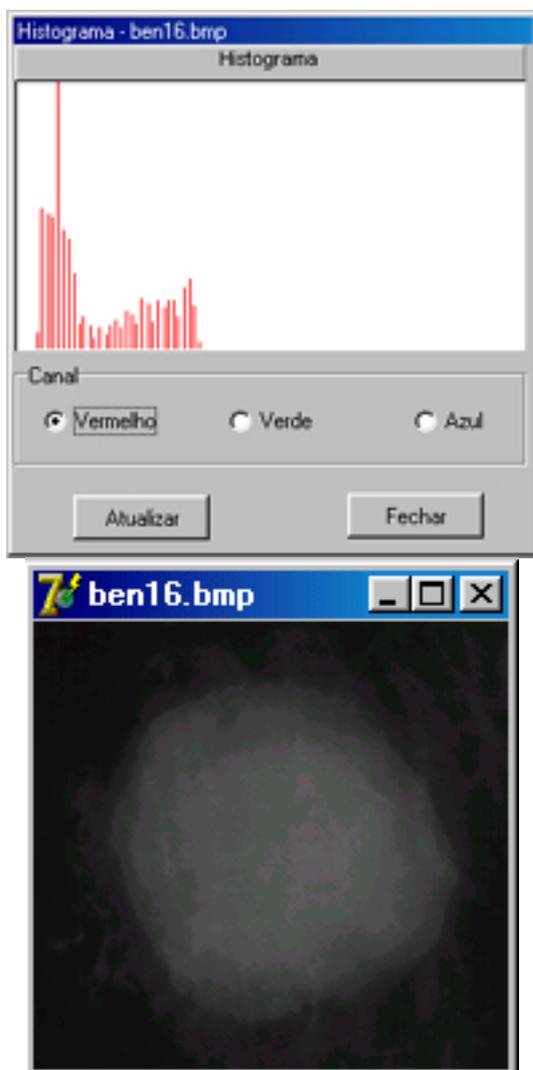


Figura 2: Histograma e nódulo com baixo contraste ainda não processados

A ampliação dos histogramas consiste em deslocar o gráfico, a partir da primeira intensidade considerável (definida através de cálculos estatísticos e denominada no sistema de **Rmin**, **Gmin** e **Bmin**), até a intensidade 0, e depois, a partir da última intensidade considerável (**Rmax**, **Gmax** e **Bmax**), “esticá-lo” até a intensidade de 255. O procedimento implementado mostrado na tabela 3 consiste em transformar os tons da imagem de modo que o contraste seja aumentado, o que resulta num histograma menos concentrado, ou seja, o gráfico passa a cobrir toda gama possível de intensidade tonal. Isso auxilia no realce dos possíveis tumores nas imagens mamográficas para um melhor processamento posterior e visualização

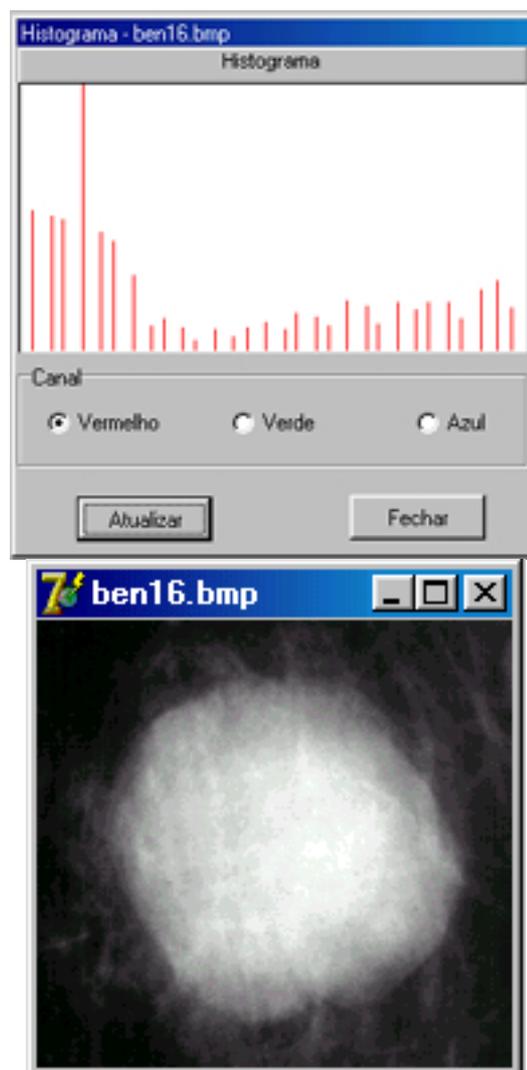


Figura 3: Histograma e nódulo depois do histograma ampliado.

Unsharp Mask

Filtros são muito utilizados em processamento de imagens. A técnica de *Unsharp Mask* trabalha com as vizinhanças de um pixel da imagem e consiste em subtrair da imagem uma versão com filtro passa-baixa homogêneo dessa mesma imagem. Essa técnica faz dois processos ao

mesmo tempo: amplia o sinal da imagem original (multiplicando esse por um fator) e retira as partes desfocadas subtraindo-as depois de multiplicadas por outro fator (redução). Assim versões da imagem são multiplicadas pelos fatores **a** e **b** respectivamente. A variação desses irá determinar qual será o resultado da imagem final. Os fatores podem ser escolhidos como constantes para toda a imagem ou variarem de região para região de acordo com as subdivisões definidas para a imagem (*windowing*). Neste trabalho, depois de diversos testes, utilizamos **a** e **b** como função de um outro parâmetro **c** que irá determinar o nível de detalhamento do resultado a partir de um diálogo mais simples com o usuário (Figura 5). O parâmetro **c** determina a modificação feita na imagem original através da seguinte equação para a aplicação do *Unsharp Mask*:

$$\frac{c}{2c-1}F - \frac{1-c}{2c-1}F_L$$

onde F se refere a imagem original e F_L a imagem processada por um filtro passa-baixa, como discutido na próxima seção. O valor de **c** varia de 0.6 a 0.85 na nossa implementação. Quando o valor de **c** tende a 0.6 o resultado tende a uma imagem mais bem definida e quando o valor de **c** tende a 0.85 o resultado tende a ficar próximo da imagem original. A versão com filtro passa-baixa é feita por uma matriz de convolução homogênea e o tamanho também pode ser escolhido pelo usuário (Figura 5).

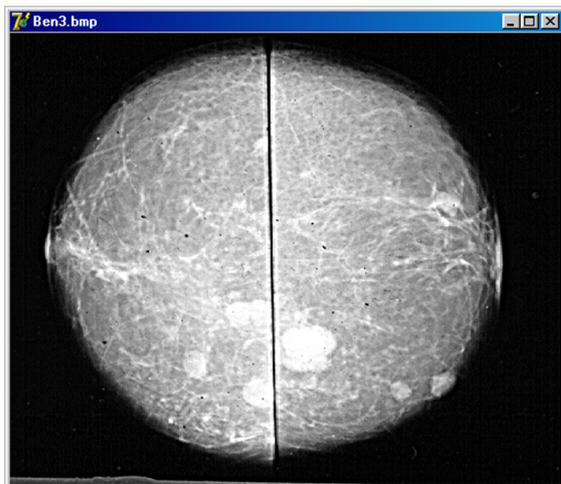


Figura 4: Imagem original e depois de processada pela técnica de Unsharp Mask

Na implementação desenvolvida, esse método é feito pela subrotina **UntUnsharp** que chama quatro outras funções: **ImgToMat**, **MatToImg**, **Convolucao** e **Subtracao** e executa o código mostrado na tabela 4. As duas primeiras funções, **ImgToMat** e **MatToImg**, fazem a transformação de uma *scanline* para a forma matricial da imagem. As

funções seguintes consideram a execução do processo de filtragem por convolução (discutido a seguir) e subtração das imagens processadas.



Figura 5: Diálogo de definição do parâmetro **c** e da área da imagem considerada como vizinhança para filtragem.

Filtragem por convolução

As técnicas de filtragem por convolução, da mesma forma que as manipulações de contraste, são transformações nos tons dos pixels na imagem. Entretanto, a modificação na imagem filtrada neste caso, não depende apenas do nível de cinza de um determinado pixel da imagem original, mas também do valor dos níveis de cinza dos pixels vizinhos àquele. A vizinhança considerada depende do tamanho da máscara do filtro, por isso, a filtragem espacial é uma transformação que depende do contexto em que se insere um dado pixel.

Na prática, o conceito de filtragem por convolução a denominação de passa alta ou passa baixa é aplicada no domínio de frequência, usando-se os conceitos da Transformada de Fourier. Os filtros geralmente se agrupam em dois tipos: os que ampliam as variações na imagem (passa-alta) ou os que reduzem essas variações (passa-baixa). Os filtros passa-alta eliminam as baixas frequências espaciais na imagem, deixando apenas as altas frequências, normalmente expressas por bordas ou limites entre áreas de diferentes valores de níveis de cinza. A maioria dos filtros de detecção de contorno atuam diferenciando a imagem e salientando as características que representam as variações sendo filtro passa-alta. Esses filtros aumentam os contrastes numa imagem, mas podem produzir, muitas vezes, bordas artificiais, que podem confundir o usuário do sistema.

No processamento de imagens digitais e nas técnicas de filtragem são largamente usadas a Transformada de Fourier, nas suas formas Discreta (DFT - *Discrete Fourier Transform*) e Rápida (FFT - *Fast Fourier Transform*) [7]. A Transformada de Fourier faz com que todo o conteúdo da imagem seja pelo somatório de senos e cossenos, em diversas frequências e esse forma de descrição da imagem é conhecida como descrição no domínio da frequência. A função original é dita descrita no domínio espacial.

Supondo que $F(u,v)$ e $H(u,v)$ denotem as transformadas de Fourier de $f(x,y)$ e $h(x,y)$, respectivamente, o teorema da convolução diz que $f(x,y) * h(x,y)$ e $F(u,v)H(u,v)$ constituem um par de transformadas de Fourier. Isto é formalmente escrito como:

$$f(x,y) * h(x,y) \Leftrightarrow F(u,v)H(u,v)$$

o que mostra que a multiplicação em um domínio equívale a convolução no outro e é realizada mais rapidamente.

Pela Identidade de Euler, pode-se ao invés de senos e cosenos utilizar a descrição da transformada de Fourier em uma forma mais elegante. Assim a DFT bidimensional [7], é escrita como

$$F(u,v) = \frac{\sum_{n=0}^{M-1} \sum_{m=0}^{N-1} f(n,m) e^{-2\pi j(un/M + vm/N)}}{NM}$$

onde (u, v) são as variáveis de freqüência, na transformada bidimensional, (n, m) são as variáveis espaciais das imagens digitais e N, M são o número de *pixels* nas duas direções da imagem. A transformada inversa é descrita na forma digital por:

$$f(n,m) = \frac{\sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{2\pi j(un/M + vm/N)}}{NM}$$

A operação de aplicar uma máscara *pixel a pixel* sobre uma imagem e calcular a quantidade predefinida em cada *pixel*, é o fundamento da convolução. Formalmente, a convolução discreta unitária de duas funções $f(n,m)$ e $h(n,m)$ de tamanho $M \times N$ é denotado por $f(n,m) * h(n,m)$ e definido por:

$$f(n,m) * h(n,m) = \frac{1}{MN} \sum_{n'=0}^{M-1} \sum_{m'=0}^{N-1} f(n',m') h(n-n',m-m')$$

Essa equação é implementação como o deslizamento da máscara de valores (n,m) ; sobre uma área da imagem e o calculo da soma dos produtos sobre todos os valores de m e n , para cada deslocamento. Os deslocamentos (n,m) são incrementos inteiros que param quando as funções não mais se sobrepõem á qualquer área da imagem [7]. A variação de n' e m' é restrita a poucos valores que correspondem à vizinhança do filtro. A forma digital desta filtragem (impulsiva) é denominada máscara.

Como regra geral, um filtro passa-alta normalmente vai realçar feições de dimensões

menores do que a dimensão da máscara usada. Os filtros passa-alta testado neste trabalho empregam máscaras de 3 x 3 a 9 x 9 de acordo com a escolha feita pelo usuário (Figura 6). A intensidade luminosa dos pixels processados não se altera pois a soma dos valores dos pesos dos filtros usados é igual a zero



Figura 6: Exemplo de matriz 3x3 de um filtro dos de detecção de lados disponíveis

Passando-se uma janela, representada por uma matriz quadrada definida pelo usuário (de ordem 3, 5, 7 ou 9), multiplicamos cada célula da janela pela intensidade de cor do pixel no qual essa célula está sobreposta, depois fazemos o somatório de todos esses valores e o valor resultante é posto no pixel que coincide com a célula central da janela. Devemos percorrer toda a imagem a fim de realizar esses cálculos para cada pixel da imagem.

O Laplaciano é equivalente em 2D da segunda derivada da imagem. A fórmula da função do Laplaciano de uma função de duas variáveis, $f(n, m)$ é:

$$\nabla^2 f = \frac{\partial^2 f}{\partial n^2} + \frac{\partial^2 f}{\partial m^2}$$

As derivadas de $f(n,m)$ são aproximadas usando equações de diferença em relação ao pixel central (i,j) :

$$\frac{\partial^2 f}{\partial n^2} = f[i, j+1] - 2f[i, j] + f[i, j-1],$$

$$\frac{\partial^2 f}{\partial m^2} = f[i+1, j] - 2f[i, j] + f[i-1, j]$$

Por combinação, temos as duas equações em um único filtro, a máscara por aproximação de Laplaciano pode ser dado por:

0	1	0
1	-4	1
0	1	0

Outros filtros podem ser gerados usando diferenças finitas diferentes. No trabalho diversos outras máscaras de derivada segunda estão disponíveis. Bem como outros filtros passa alta podem ser selecionados como os Sobel

direcionais. Outros filtros também podem ser escolhidos do menu disponibilizado neste trabalhos como os baseados em Gaussianas 2D (filtros passa baixa). Filtros também podem ser fornecidos pelo usuário e incluídos no programa para uso posterior.

Utilizando-se máscaras para passar o filtro sobre a imagem, os pixels da borda da imagem não são tratados. Se opção "Borda: tratar Borda" estiver marcada pelo usuário no menu de definição do filtro de convolução mostrado na Figura 6, o programa utiliza um método para tratar os pixels da borda. O sistema faz uma cópia das $n+1$ linhas e colunas da borda, no lado de fora da imagem, para dessa maneira o filtro poder ser aplicado à borda. Todos os filtros (com ou sem tratamento de borda) são implementados de maneira única apenas variando seu tamanho e os valores de sua máscara como mostrado na tabela 5.

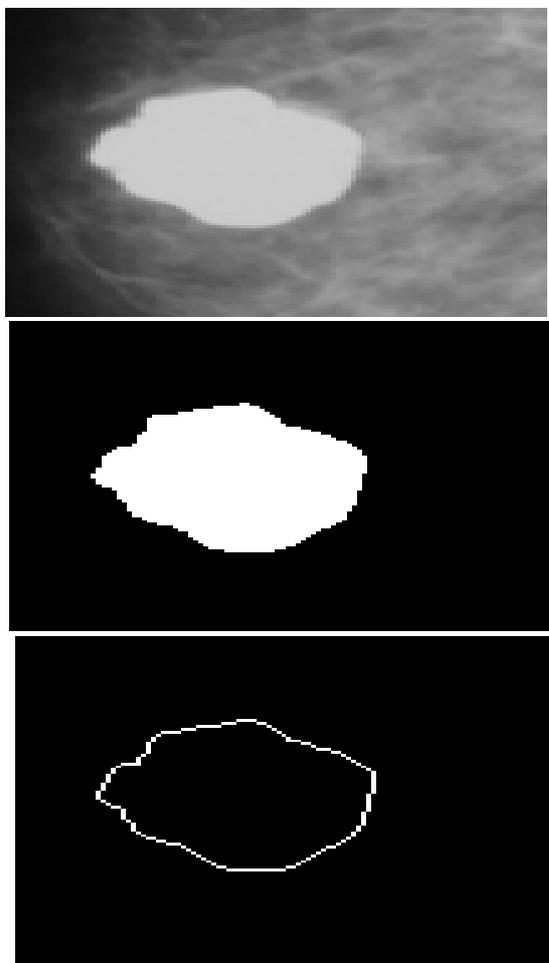


Figura 7: Imagem original, mesma imagem depois de binarizada e tratada com um Filtro de Edge Detection

Considerações finais

Outros procedimentos foram implementado no sistema como a limiarização da imagem para preto-e-branco (figura 7), sua inversão, a passagem da forma colorida para monocromática, as opções de desfazer e refazer algum processo, a equalização e outras.

Ao testar diversos filtros e técnicas foi possível perceber uma visível melhora na qualidade da mamografia, principalmente na seqüência de equalização de histograma e *unsharp mask*, que resalta o formato do tumor para melhor visualização e análise posterior [2,3,6]. O agrupamento deste sistema com técnicas de identificação automática de malignidade deve ser o passo seguinte desta linha de pesquisa.

Agradecimentos

Os autores agradecem aos demais alunos do curso de Análise de Imagens da turma de 2004 pela colaboração na implementação do sistema. O último autor agradece ao CNPq pelo apoio financeiro na participação desse congresso.

Referências

- [1] D. A. Clunie. "DICOM Structured Reporting," PixelMed Publishing, Pennsylvania, 2000.
- [2] A. Conci, L.M. Soares, A.D. Vianna, "Identification of Benign and Malignant Lesion by Feature Extraction on Mammographic Images", Applied Mechanics in Americas, Vol. 6, pp. 53-56, Proceedings of VI Pan American Congress of Applied Mechanics - PACAM Rio de Janeiro, 1999, Edited by D. Pamplona, P.B. Conçalves, I. Jasiuck C.R. Steele, H.I. Weber, L. Bevilacqua, [3] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 2.Ed. New Jersey: Prentice Hall, 2002. 793 pp.
- [3] A. Conci, L.M. Soares, "Identificação de Câncer e Tumores dos Seios pela Análise da Forma de suas Imagens", vol. 3 dos anais do IV Congresso Iberoamericano de Ingeniería Mecánica, 1999 Santiago de Chile, pp. 200-204.
- [4] P. L. Meyer, Probabilidade: aplicações à estatística. 2.ed. Rio de Janeiro: LTC - Livros Técnicos e Científicos, 1983. 426 p.
- [5] D.L., Pham, C. Xu, J.L Prince., Current Methods in Medical Image Segmentation. Annu. Ver. Biomed. Eng. 2000. 02:315-37
- [6] L.M. Soares, A. Conci, A.D. Vianna, "Automated Classifications of Masses on Mammography", SIBGRAPI'98, Rio de Janeiro - 1998, 4pp. published at <http://www.ifas.sc.usp/visao/sib98/proceedings>.
- [7] A. Watt and F. Policarpo, The Computer Image. Addison-Wesley, 1998. 751pp.

Apêndices

Nesta seção algumas das soluções implementadas, discutidas no texto, tem seu código apresentado nas tabelas que seguem.

Tabela 1 - Cálculo do histograma

```
procedure TFrmHistogram.RBVermelhoClick(Sender: TObject);
var
  i: integer;
begin
  Histograma.SeriesList.Series[0].Clear;
  case (Sender as TRadioButton).Hint[1] of
    'R':for i:=0 to 255 do
      Histograma.SeriesList.Series[0].Add(HistR[i]," , clRed);
    'G':for i:=0 to 255 do
      Histograma.SeriesList.Series[0].Add(HistG[i]," , clGreen);
    'B':for i:=0 to 255 do
      Histograma.SeriesList.Series[0].Add(HistB[i]," , clBlue);
  end;
end;
```

Tabela 2- Acesso eficiente ao valor da cor

```
Intensidade : inteiro longo
IntR, IntG, IntB : inteiro
intensidade <- valor de cor do pixel
IntR <- (Intensidade      ) E 11111111
IntG <- (Intensidade deslocar 8bits para a direita) E 11111111
IntB <- (Intensidade deslocar 16bits para a direita) E 11111111
```

Tabela 3 - Modificação das intensidades pixel a pixel.

```
//Definição dos Fatores
fatorR:=256/(Rintmax-Rintmin);
fatorG:=256/(Gintmax-Gintmin);
fatorB:=256/(Bintmax-Bintmin);
for j:=0 to Imagem.Height-1 do
begin
  line:=Imagem.Picture.Bitmap.ScanLine[j];
//Modificação dos tons de cada pixel
  for i:=0 to Imagem.Width-1 do
  begin
    Cor:=@line^[i];
    Cor[Red_] := Round(min(max(Cor[Red_] - Rintmin, 0) * fatorR, 255));
    Cor[Green_] := Round(min(max(Cor[Green_] - Gintmin, 0) * fatorG, 255));
    Cor[Blue_] := Round(min(max(Cor[Blue_] - Bintmin, 0) * fatorB, 255));
  end;
end;
```

Tabela 4 - Código de processamento da técnica de *unsharp masking*.

```

if Result=mrOk then
  with FrmUnsharp do
  begin
    Raio:=TrBRAio.Position*2 + 1;
    for j:=0 to Raio-1 do
      for i:=0 to Raio-1 do
        ConvMat[i, j]:=1/(Raio*Raio);
      c:=(25-TrackBar1.Position+60)/100;
      FatorA:=c/(2*c-1);
      FatorB:=(1-c)/(2*c-1);
      ImgToMat(Imagem, Cmat1);
      Convolucao(Imagem, ConvMat, Raio, true, Progress);
      ImgToMat(Imagem, CMat2);
      Subtracao(CMat1, CMat2, Imagem.Height, Imagem.Width, FatorA, FatorB);
      MatToImg(CMat1, Imagem);
    end;
  end;

```

Tabela 5 - Filtragem por convolução

```

function aplicaconvolucao(Img: TImage; Progress:TProgressBar):Integer;
begin
  Result:=mrCancel;
  FrmFiltro:= TFrmFiltro.Create(Application);//TColor
  if FrmFiltro.ShowModal = mrOk then
    with FrmFiltro do
      begin
        Escala:=StrToInt(EdtEscala.Text);
        Result:= mrOk;
        if Escala = 0 then
          Escala:=1;
          GridToMat(StringGrid1, Escala);
          Convolucao(Img, Mat, Tam, RBtTratar.Checked, Progress);
          Img.Refresh;
        end;
      end;
    FrmFiltro.Destroy;
  end;
end;

```