

CENTRO UNIVERSITÁRIO FEEVALE

JOÃO BATISTA MOSSMANN

ESTUDO DE TÉCNICAS DE PROCESSAMENTO DE IMAGENS
APLICADAS AO APOIO DO DIAGNÓSTICO DE DERRAMES
Serosos de etiologia benigna e maligna

Novo Hamburgo, junho de 2007.

JOÃO BATISTA MOSSMANN

ESTUDO DE TÉCNICAS DE PROCESSAMENTO DE IMAGENS
APLICADAS AO APOIO DO DIAGNÓSTICO DE DERRAMES
Serosos de etiologia benigna e maligna

Centro Universitário Feevale
Instituto de Ciências Exatas e Tecnológicas
Curso de Ciência da Computação
Trabalho de Conclusão de Curso

Professor Orientador: Marta Rosecler Bez el Boukhari
Professor Co-Orientador: Miriam Alice Frantz

Novo Hamburgo, junho de 2007.

Agradecimentos

...primeiramente a algo extraordinário que não sei explicar, apenas aprendi a chamar de DEUS.

Ao amor incondicional dos meus pais a quem devo carinho e amor eterno, a todos os amigos e familiares que me apoiaram nas diferentes etapas da vida.

A Feevale e ao CSI por me proporcionar crescimento profissional, acadêmico e pessoal.

Aos Mestres que fizeram parte desta caminhada, em especial a Professora Marta que sempre de forma incansável conduziu a orientação deste trabalho, não apenas como professora e orientadora, mas como amiga, tal como a Professora Miriam que ajudou a iluminar o mundo biomédico.

Muito Obrigado!

RESUMO

O uso de imagens médicas como fonte de dados pode melhorar o processo de diagnóstico. Através do seu processamento e análise pode-se aprimorar a etapa de caracterização patológica. É sabido que a baixa qualidade e a sobreposição das estruturas nas imagens que são artefatos finais de exames dificultam à etapa de diagnóstico. Além destes problemas técnicos, a distração, a avaliação subjetiva do profissional e cansaço visual são fatores determinantes que justificam o uso de Sistemas de Apoio ao Diagnóstico (SAD) para auxiliar o profissional da área da saúde. Sua finalidade é processar os dados relevantes e disponibilizar as informações necessárias de forma clara e precisa, formalizando assim uma segunda opinião para melhorar o processo de identificação patológica e minimizar os erros durante a etapa de diagnóstico. Auxiliar ao diagnóstico não é o único propósito das imagens médicas. Estas também são empregadas no planejamento e simulações cirúrgicas assim como em navegação intra-operativa. Este fato pode ser observado em alguns trabalhos publicados na área. O presente projeto descreve algumas técnicas aplicadas na análise de imagens médicas, tendo como principal foco de atuação o estudo dos princípios de realce e segmentação, verificando também alguns métodos aplicados na quantificação de estruturas, ou seja, da análise morfométrica de núcleos de derrames serosos de etiologia benigna e maligna corados por May-Grünwald Giemsa – MGG.

Palavras-chave: Imagens Médicas; Processamento de Imagens; Visão Científica

ABSTRACT

The use of medical images as a data source can improve the diagnostic process. Through the process and analysis of medical images it can accurate the pathologic characterization stage. It is known that the low quality, the overlapping of the structures in the images that are final analysis devices make it difficult to the diagnostic stage, beyond the technique problems, distraction, subjective evaluation of the physician and the visual fatigue are determining factors that justify the computer aided design employment to help health professionals. The system's purpose is to process important data and turn available the necessary information in a clear and objective way, acting as a second opinion to improve the pathologic identification process and to minimize the mistakes during the diagnostic. To help the diagnostic is not the only challenge of the medical images. These are also used in the planning and to simulate surgeries and in the computer-integrated surgery. This fact can be observed in some researches published in this area. The present project describes some techniques that usually are applied to analyze medical images, having as the main actuation the study of the enhancement principles, segmentation, moreover, to verify some methods applied to quantify the structures applied in analyzed nuclear by morphological measured.

Key words: Medical Images; Image Processing; Computer Vision

LISTA DE FIGURAS

Figura 2.1- Equalização do Histograma - (a) Imagem original; (b) Histograma da imagem (a); (c) Histograma cumulativo da imagem (a) _____	20
Figura 2.2 - Equalização do Histograma - (a) Imagem com o histograma equalizado; (b) Histograma da imagem (a); (d) Histograma cumulativo da imagem (a) _____	21
Figura 2.3 - Máscaras de vários tamanhos (a) Máscara 3x3; (b) Máscara 5x5; (c) Máscara 7x7 _____	22
Figura 2.4 –Filtro da Média - (a) Imagem original; (b) Imagem Original com adição de ruído; (c) Resultado do filtro da média com uma máscara 3x3 (d) Resultado do filtro da média com uma máscara 9x9 _____	23
Figura 2.5 - Operação de realce com o filtro da mediana _____	24
Figura 2.6 – Subtração de Imagens - (a) Máscara para realizar a técnica de subtração de imagem; (b) Imagem após injeção de corante na corrente sanguínea e com subtração da máscara _____	26
Figura 2.7 - Exemplo de redução de ruído através de médias _____	27
Figura 2.8 - Exemplo de um histograma propício à aplicação da técnica de limiarização ____	29
Figura 2.9 - Exemplo de segmentação de células com a técnica de limiar _____	29
Figura 2.10 - Típicas máscaras do operador de Sobel _____	30
Figura 2.11 - Máscaras para o operador Laplaciano _____	31
Figura 2.12 - Segmentação pelo algoritmo de <i>hill-climbing</i> _____	33
Figura 2.13 - Exemplo de segmentação com o algoritmo de watershed _____	34
Figura 2.14 - Exemplo de uma mamografia _____	37
Figura 2.15 - Cálculo da média do volume do ventrículo esquerdo _____	38
Figura 3.1 - Resultado do ensaio da equalização do histograma _____	42
Figura 3.2 - Histogramas do ensaio _____	42
Figura 3.3 - Ensaio da Média _____	43

Figura 3.4 - Imagem original e resultados obtidos com a técnica de <i>Connected Threshold</i>	44
Figura 3.5 - Imagem original e resultados obtidos com a técnica de <i>Neighborhood Connected</i>	46
Figura 3.6 - Imagem segmentada com <i>Neighborhood Connected</i> para testar a seleção de pequenas estruturas presentes no artefato original	47
Figura 3.7 - Ensaio do algoritmo de <i>watersheds</i>	47
Figura 3.8 - Ensaio do modelo de detecção de bordas	48
Figura 3.9 - Imagem original	49
Figura 3.10 - Ensaio de segmentação de imagens	50
Figura 3.11 - Segmentação por contornos ativos; (a) Processo de segmentação; (b) Término da segmentação	51
Figura 3.12 -Ensaio com a técnica de <i>Geodesic Active Contours</i>	52
Figura 4.1 – Imagem citopatológica que apresenta as várias estruturas presentes na placa histológica	59
Figura 5.1 - Fluxo do sistema de análise de imagens	63
Figura 5.2 - Seleção da área de interesse	63
Figura 5.3 - Imagem 277448-P-01 utilizada como base para testes	64
Figura 5.4 - Imagem resultante da operação <i>Curvature Anisotropic Diffusion</i> ; (a) Operação com 25 iterações; (b) Operação com 50 iterações	64
Figura 5.5 - Imagem resultante da operação <i>Gradient Anisotropic Diffusion</i> ; (a) Operação com 25 iterações; (b) Operação com 50 iterações; (c) Operação com 70 iterações	65
Figura 5.6 - Canal vermelho; Canal azul; Canal verde	65
Figura 5.7 - A esquerda a imagem no canal verde e a direita com o ajuste do brilho e contraste	67
Figura 5.8 - Sucesso na detecção de bordas, primeiros três para os Casos Negativos de câncer e as três últimas para Casos Positivos	68
Figura 5.9 - Casos negativos com ajustes manuais; (a) Detecção da borda; (b) Local dos ajustes manuais; (c) Borda ajustada; (d) Segmentação	69
Figura 5.10 - Casos positivos com ajustes manuais; (a) Detecção da borda; (b) Local dos ajustes manuais; (c) Borda ajustada; (d) Segmentação	69
Figura 5.11 - Testes de união de bordas	70
Figura 5.12 - Tela do protótipo computacional desenvolvido	74
Figura 5.13 - Menu Pré-Processamento	74
Figura 5.14 - Menu Segmentação	75

Figura 5.15 - Menu de Quantificação _____	76
Figura 5.16 - Imagens de teste para o sistema de avaliação _____	78
Figura 5.17 - Sistema de avaliação de segmentação de imagens _____	79
Figura 5.18 - (a) Imagem segmentada manualmente pelo patologista; (b) Máscara binária baseada na segmentação manual _____	80
Figura 5.19 - Caso negativo para discussão _____	83
Figura 5.20 - Imagem de referência _____	84
Figura 5.21 - Comparação entre a imagem segmentada manualmente e o ROI _____	84
Figura 5.22 - Caso 291418-P-06 _____	84
Figura 5.23 - Comparação entre a imagem segmentada manualmente e o ROI _____	85

LISTA DE TABELAS

Tabela 3.1 - Descrição dos dados de entrada para o algoritmo de <i>Connected Threshold</i> ____	44
Tabela 3.2 - Descrição dos dados de entrada para o algoritmo de <i>Neighborhood Connected</i>	45
Tabela 5.1 - Amostra estudada _____	62
Tabela 5.2 - Resultados do RUMA em casos negativos _____	80
Tabela 5.3 - Resultados do RUMA em casos positivos _____	81

LISTA DE ABREVIATURAS E SIGLAS

SAD	Sistema de Apoio ao Diagnóstico
CT	Tomografia Computadorizada
2D	Bidimensional
MRI	Magnetic Resonance Imaging
PET	Tomografia por Emissão de Pósitron
SPECT	Tomografia por Emissão de Fóton
3D	Tridimensional
ITK	Insight Toolkit
MATLAB	Matrix Laboratory
UNC	University of North Carolina
UT	University of Tennessee
UPenn	University of Pennsylvania
MGG	May-Grünwald Giemsa
TICAS	Taxonomic Intra-Cellular Analytic Systems for cell identification
TIFF	Tagged Image File Format
ROI	Região de Interesse
RIUL	Robust Image Understanding Laboratory
UBC.ca	University of British Columbia
RUMA	Relative Ultimate Measurement Accuracy

SUMÁRIO

INTRODUÇÃO	13
1.1 Motivação e Objetivos	13
1.2 Estrutura do Trabalho	14
2 CONCEITOS BÁSICOS EM IMAGENS MÉDICAS	16
2.1 Imagens Digitais	16
2.1.1 Aquisição de Imagens Médicas	16
2.2 Técnicas de Realce	17
2.2.1 Equalização do Histograma	18
2.2.2 Filtro de Suavização	21
2.2.3 Subtração de Imagens	24
2.2.4 Média de Imagens	26
2.3 Técnicas de Segmentação	27
2.3.1 Limiarização – <i>Thresholding</i>	28
2.3.2 Métodos baseados em detecção de bordas	30
2.3.3 Métodos baseados no crescimento de regiões	31
2.3.4 Algoritmo <i>Watershed</i>	33
2.3.5 Modelos Deformáveis	34
2.4 Quantificação	36
3 APLICAÇÃO PRÁTICA DE CONCEITOS BÁSICOS EM IMAGENS MÉDICAS	40
3.1 Ferramentas	40
3.2 Ensaios	41
3.2.1 Realce	41
3.2.2 Segmentação	43
3.2.3 Outros Testes	49
4 ESTUDO BIOMÉDICO DA ANÁLISE MORFOMÉTRICA (ASSINATURA NUCLEAR) DE NÚCLEOS DE DERRAMES SEROSOS DE ETIOLOGIA BENIGNA E MALIGNA CORADOS POR MAY-GRÜNWARD GIEMSA - MGG	53
4.1 O Estudo Biomédico	53
4.1.1 Objetivos	54
4.1.2 Características Morfométricas	54
4.2 Protótipo computacional aplicado ao estudo biomédico	57
4.2.1 Objetivos	57
4.2.2 Metodologia	58
5 SEGMENTAÇÃO DE IMAGENS E ANÁLISE MORFOMÉTRICA DE NÚCLEOS DE DERRAMES SEROSOS DE ETIOLOGIA BENIGNA E MALIGNA	61

5.1 Delineamento do estudo	61
5.2 População estudada	62
5.3 Processo	62
5.3.1 Selecionar ROI	63
5.3.2 Passar para o canal verde	63
5.3.3 Ajuste de brilho e contraste	66
5.3.4 Detecção de bordas	67
5.3.5 <i>Linking Edge</i>	70
5.3.6 Seleção da área	70
5.3.7 Negativo da Imagem	71
5.3.8 Soma com o ROI	71
5.4 Análise morfométrica	72
5.4.1 Cálculo da Área	72
5.4.2 Cálculo do Perímetro	72
5.4.3 Cálculo da Circularidade	72
5.5 Protótipo Desenvolvido	73
5.6 Avaliação da segmentação	76
5.6.1 Índice de Pratt	76
5.6.2 Índice de Similaridade	76
5.6.3 RUMA	77
5.7 Discussões	82
CONSIDERAÇÕES FINAIS	86
REFERÊNCIAS BIBLIOGRÁFICAS	88
ANEXOS	93

INTRODUÇÃO

O presente capítulo relaciona os objetivos e os eventos que motivam a realização deste trabalho, bem como a forma em que está organizado este volume.

1.1 Motivação e Objetivos

Conforme destacado por Ambrósio (2002, p.11) é crescente a utilização de computadores como ferramenta de auxílio nas diversas áreas do conhecimento, onde se destaca a Medicina como sendo uma das principais. Dentre as inúmeras aplicações para sistemas computadorizados na área médica, os sistemas de apoio ao diagnóstico são os de aplicação mais expressiva. Sistemas desse tipo trabalham com processos que envolvem um grande número de dados e informações que necessitam ser processadas rapidamente e têm como principal objetivo auxiliar profissionais da área na tomada de decisões.

Os estudos de imagens médicas podem auxiliar no diagnóstico, já que possibilitam uma melhor visualização e inspeção das estruturas anatômicas. Além disso, já vem sendo usadas no planejamento e simulações cirúrgicas, navegação intra-operativa e detecção de patologias. (SILVA, 2005). Quanto à segmentação de imagens médicas, estas estão sendo habitualmente aplicadas em várias situações, tais como quantificação de tecido, volumes, diagnósticos, localização de patologia, estudo da estrutura anatômica, planejamento do tratamento e cirurgia auxiliada por computador. (XU, 2000).

A falta de trabalhos referente a imagens médicas, envolvendo o curso de Ciência da Computação da instituição, justifica e motiva a realização do referido projeto, a fim de proporcionar uma visão geral da área e sua aplicação. Pode-se comprovar a necessidade de tal estudo, visto que se encontram aplicação em futuras e presentes pesquisas realizadas no Instituto de Ciências da Saúde.

O objetivo geral do presente trabalho é um levantamento teórico das técnicas de análise de imagens, que quando aplicadas permitem solucionar dificuldades clássicas na área

tema do projeto, aqui evidenciada pela possível baixa qualidade nos artefatos, assim como a necessidade de segmentação de estruturas de relevância clínica e científica.

O presente trabalho tem como objetivo o estudo das técnicas de análise de imagens médicas, bem como sua aplicação em sistemas que as utilizam como sua principal fonte de dados. No decorrer da disciplina de Trabalho de Conclusão I tinha-se como meta estudar a Correlação da Infecção por *Helicobacter pylori* com a ocorrência de processo aterosclerótico em camundongos C57BL/6 submetidos à dieta hiperlipidêmica. Em função de problemas no procedimento biológico que seria aplicado às cobaias, o projeto se tornou inviável, uma vez que os camundongos não inocularam a bactéria. A professora biomédica que acompanhava o estudo propôs então um novo projeto, onde as mesmas técnicas computacionais poderiam ser aplicadas, qual seja “Análise Morfométrica (Assinatura Nuclear) de núcleos de derrames serosos de etiologia benigna e maligna corados por May-Grünwald Giemsa – MGG”, produzido pelo Instituto de Ciências da Saúde no Curso de Biomedicina desta instituição.

Dentre os objetivos específicos pretende-se a aplicar e estudar de técnicas computacionais capazes de identificar e segmentar os núcleos celulares e quantificar morfometricamente estas estruturas segmentadas, afim de que estas sejam utilizadas no apoio ao diagnóstico de benignidade e malignidade. As imagens utilizadas no estudo aqui apresentado são cedidas pelo estudo biomédico do curso de Biomedicina do Centro Universitário Feevale.

1.2 Estrutura do Trabalho

O presente capítulo apresenta os temas estudados, destacando os objetivos propostos, motivação para realização do trabalho, assim como relata a estrutura da monografia, demonstrando de forma sucinta os assuntos abordados nas seções seguintes deste trabalho.

No capítulo dois são descritas estratégias para o realce e segmentação de imagens médicas, assim como técnicas de quantificação de estruturas. As técnicas descritas nesse capítulo são oriundas de um levantamento teórico básico que visa apresentar primeiramente a contextualização e definições sobre imagens, em seguida modelos de realce, segmentação e finalmente, quantificação de estruturas utilizadas em projetos correlatos a este trabalho.

No capítulo três são realizados alguns testes, ensaios práticos, com algoritmos de segmentação e realce os quais são executados com ferramentas já utilizadas para os mesmos propósitos. Também é apresentado estudo biomédico no capítulo quarto. Este descreve o

delineamento de um protótipo empregado na segmentação de imagens e da análise morfométrica de células adquiridas junto a uma pesquisa biomédica.

O capítulo cinco demonstra o protótipo desenvolvido assim como as técnicas para quantificação morfométrica, pré-processamento, segmentação, além da análise qualitativa das segmentações efetuadas durante o desenvolvimento do sistema apresentado no capítulo.

O capítulo seis contém as considerações finais em relação ao trabalho desenvolvido até o momento e as perspectivas para trabalhos futuros decorrentes deste estudo, possibilitado a continuidade da pesquisa.

2 CONCEITOS BÁSICOS EM IMAGENS MÉDICAS

O objetivo deste capítulo é apresentar os métodos básicos de realce, segmentação e quantificação de estruturas em imagens médicas. Embora tais técnicas se apliquem nas diversas áreas do processamento e análise de imagens, pretende-se aqui ilustrar a aplicação de tais princípios em imagens médicas.

O intuito dessa revisão bibliográfica é proporcionar uma visão geral dos modelos aqui descritos, objetivando elucidar os resultados destas e assim, ajudar no processo de seleção das técnicas para construção do protótipo proposto no trabalho.

2.1 Imagens Digitais

Conforme Falcão (1993, p.7) define-se por imagem digital uma matriz $m \times n$ onde n é o índice de linha e m identifica a coluna, representando assim um ponto qualquer da imagem. Em cada ponto da imagem é associado um valor que representa uma grandeza física qualquer medida naquele determinado ponto, onde a grandeza física pode variar conforme o equipamento utilizado para a aquisição da imagem.

Cada elemento da matriz que representa a imagem é chamado de *pixel*, uma abreviação para *picture element* (FALCÃO, 1993, p.7; QIU, 2005).

Uma imagem monocromática pode ser expressa através de uma função bi-dimensional $f(x, y)$ onde (x, y) são as coordenadas espaciais e o valor de f representa o nível de cinza de um determinado ponto (FALCÃO, 1993, p. 7). Frequentemente, a literatura da ciência da computação refere-se a $f(x, y)$ como *image intensity function* (QIU, 2005, p. 2).

2.1.1 Aquisição de Imagens Médicas

O objetivo de obterem-se imagens médicas pode ser dividido em três áreas: 1 – operação de filtro para remoção de ruído e realce, 2 – detecção de condições e eventos clínicos e 3 – quantificação e análise das estruturas. (YOO, 2004)

Existem várias formas de aquisição de imagens médicas, dentre os quais pode-se citar:

- A Tomografia Computadorizada (CT) que foi desenvolvida no princípio dos anos 70. É capaz de obter uma imagem girando um emissor de raios-x em torno do paciente medindo a intensidade de raios transmitidos. O resultado é uma coleção 2D de seções do corpo que fornecem informações anatômicas. (YOO, 2004)

- *Magnetic Resonance Imaging* (MRI): uma das vantagens do MRI é o fato de não utilizar radiação (FALCÃO, 1993). Um sistema simples de MRI é formado por um cilindro, onde o paciente deve ser acomodado e um ímã é utilizado para produzir um campo magnético (YUAN, GALLAGHER, CROZIER, 2005). As imagens obtidas através dos processos de CT e MRI são de alta resolução – normalmente 512 x 512 *pixels*. (FALCÃO, 1993)

- No caso de aquisição de imagens biológicas, as estruturas podem ser obtidas através de um microscópio com uma câmera acoplada. Em estruturas opacas, a câmera obtém imagens de cortes previamente realizados sobre a estrutura, quando transparentes, as imagens são obtidas ajustando o foco do microscópio (FALCÃO, 1993, p.10).

Além dos métodos anteriormente destacados, existem outros capazes de registrar informações de estruturas, em formato de imagem, como é o caso dos equipamentos de ultrassom, tomografia por emissão de pósitron (PET), tomografia por emissão de fóton (SPECT) que fornecem imagens de baixa resolução, normalmente 64 x 64 *pixels*. (FALCÃO, 1993).

2.2 Técnicas de Realce

As técnicas para realce de imagens são procedimentos computacionais que objetivam melhorar o visual da imagem a fim de ajudar a análise humana da mesma. (GONZALES, 2002). Qualquer tipo de melhoria, adequada, na qualidade visual da imagem, ajuda na interpretação por um médico especialista. (RANGAYYAN, 2000)

As imagens médicas podem ser de baixa qualidade devido às várias fontes de ruído promovidas por diversos fenômenos inerentes ao processo de aquisição da imagem, além disso, os procedimentos envolvidos na aquisição de imagens provocam a diminuição do

contraste e, por conseguinte, inviabilizam a visualização de detalhes. Exemplo disso são os ruídos gerados pela emissão de fótons na medicina nuclear (RANGAYYAN, 2000), ou ainda, as mamografias obtidas através de raios-x, que possuem baixo contraste e a sobreposição natural de tecidos, tornado assim dificultosa a análise direta da imagem tendo em vista que a sobreposição dificulta a distinção dos tecidos anormais dos normais. (RANGAYYAN, 2000; MARQUES, 2001).

O realce é obtido através de técnicas matemáticas processadas computacionalmente e promovem a melhoria da qualidade da imagem dada, além de permitirem a melhor visualização de algumas características próprias previamente selecionadas. Essas técnicas podem ser utilizadas por um observador humano ou ainda outro algoritmo subsequente com propósito definido. As técnicas de realce devem ser utilizadas de maneira correta, visto que se aplicadas de forma incorreta podem aumentar o ruído ou ainda eliminar detalhes presentes no artefato original. (RANGAYYAN, 2000)

2.2.1 Equalização do Histograma

O histograma de uma imagem é uma das ferramentas mais utilizadas nos processos que fazem uso de imagens digitais. Normalmente representa-se um histograma através de um gráfico de barras horizontais onde a altura dessas representa o número de *pixels* que possuem o mesmo nível de cinza. (DAVIS, 1992). Conforme Paranjape (2000, p. 4), histograma é um vetor, iniciando em zero até $P-1$, que armazena o número de *pixels* em todos os níveis de cinza e pode ser representado matematicamente por:

$$h(i) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sigma(f(m,n) - i), \quad i = 0, 1, \dots, P-1$$

(2.1)

onde

$$\sigma(w) = \begin{cases} 1 & w = 0, \\ 0 & \text{qualque outro caso.} \end{cases}$$

Sendo, $f(m,n)$ onde f é a intensidade do *pixel* e m e n a posição do mesmo no plano, logo M o número de linhas, N o número de colunas e P o nível de intensidade, variando de zero até $P-1$. (PARANJAPE, 2000).

Com o auxílio dos dados apresentados no histograma pode-se inferir uma série de informações qualitativas importantes sobre a imagem. Uma imagem em tons de cinza que

possui grande concentração de *pixels* próximas a 255 é uma imagem clara, já aquelas que apresentam uma aglomeração perto do zero são consideradas imagens escuras. É possível melhorar o contraste das imagens ajustando a distribuição dos níveis de cinza, utilizando técnicas de equalização de histogramas. (DAVIS, 1992; PARANJAPE, 2000).

Considerando-se uma imagem em tons de cinza, é possível melhorar o realce da mesma a partir da distribuição uniforme dos níveis de cinza sobre o comprimento total da imagem, movendo os *pixels* com uma intensidade dada para outra, resultando assim em um incremento na intensidade para onde o *pixel* foi movido, assim como o decremento de onde o mesmo foi removido. (GONZALES, 2002; PARANJAPE, 2000).

Uma forma de melhorar a distribuição é dada por $(M * N) / P$, visto que como definido anteriormente, o histograma, $h(i)$, possui a contagem dos níveis de cinza de cada *pixel* da imagem, de forma que cada posição do vetor deve conter uma contagem $(M * N) / P$ de *pixels* (PARANJAPE, 2000).

Uma das técnicas para realizar a redistribuição dos *pixels* em uma imagem é a baseada na normalização do histograma cumulativo e pode ser definido como:

$$H(j) = \frac{1}{M \cdot N} \sum_{i=0}^j h(i) \quad j = 0, 1, \dots, P-1 \quad (2.2)$$

A técnica anteriormente descrita pode ser usada para verificar os níveis de cinza entre a imagem original e os novos valores necessários para o realce da imagem. Uma imagem realçada, $g(m, n)$, terá um histograma uniforme se o mesmo for definido como:

$$g(m, n) = (P - 1) \cdot H(f(m, n)).$$

As duas figuras a seguir apresentam o histograma e o histograma acumulativo das imagens em destaque.

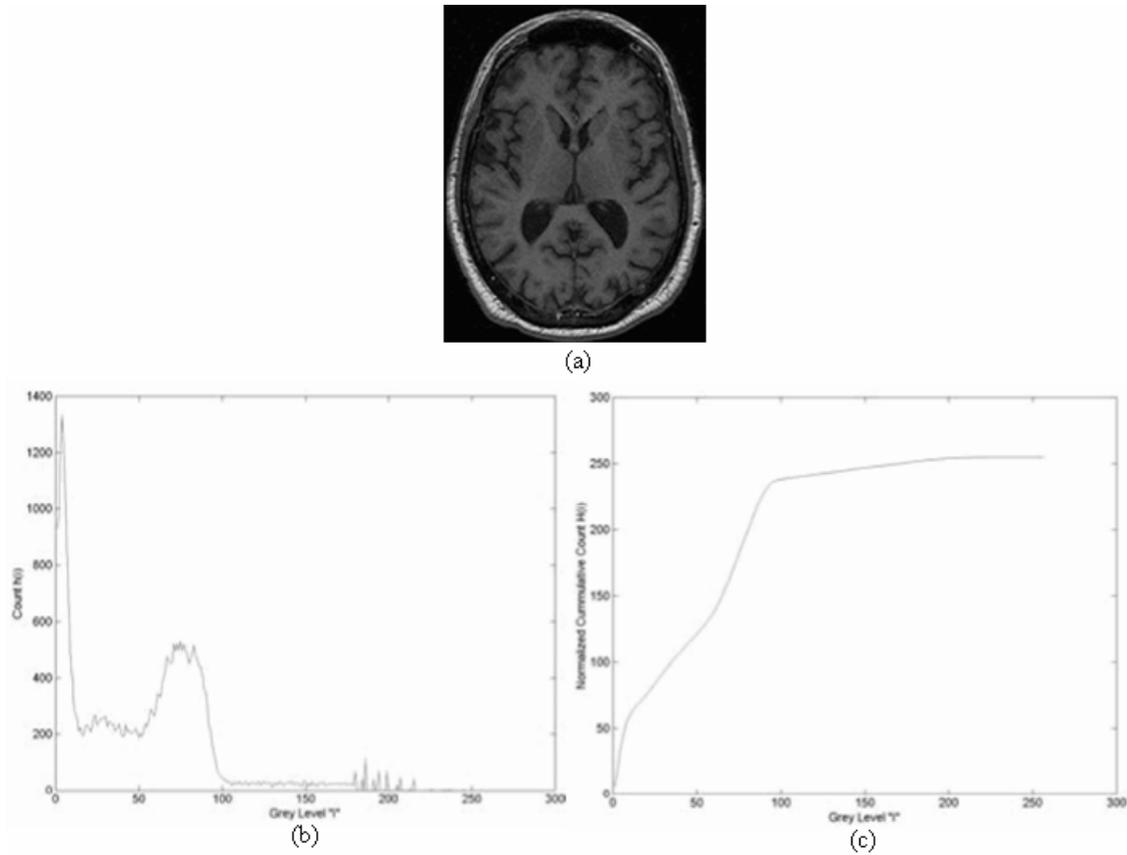


Figura 2.1- Equalização do Histograma - (a) Imagem original; (b) Histograma da imagem (a); (c) Histograma cumulativo da imagem (a)

Fonte: (PARANJAPE, 2000, p. 9)

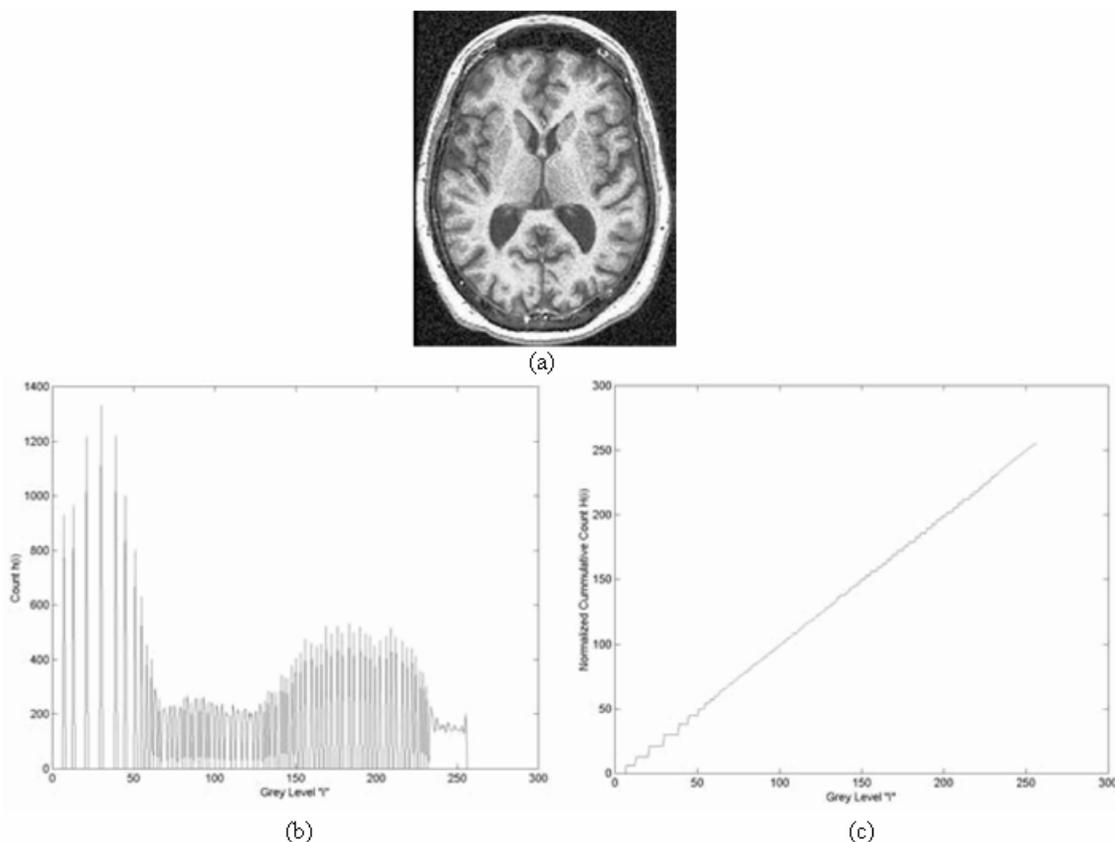


Figura 2.2 - Equalização do Histograma - (a) Imagem com o histograma equalizado; (b) Histograma da imagem (a); (d) Histograma cumulativo da imagem (a)

Fonte: (PARANJAPE, 2000, p 10)

A Figura 2.1 (a) apresenta uma imagem MRI do cérebro onde os níveis de cinza não estão distribuídos uniformemente. Observa-se que muitos detalhes não são visíveis na imagem original, já a Figura 2.1 (b) e (c) demonstram o histograma e o histograma cumulativo respectivamente. A Figura 2.2 (a) é a representação da imagem após a normalização do histograma cumulativo, onde são observados vários detalhes não presentes na Figura 2.1 (a). A partir da equalização do histograma é possível uma melhor detecção das estruturas patológicas. A Figura 2.2 (b) é o histograma, já a Figura 2.2 (c) é o histograma cumulativo correspondente a Figura 2.2 (a). O ideal é que a representação gráfica da Figura 2.2 (c) tenha a forma de uma linha reta, iniciando no ponto $(0,0)$ até $(P-1, P-1)$. O gráfico atual não possui a forma ideal dada à distribuição inicial dos níveis de cinza. (PARANJAPE, 2000)

2.2.2 Filtro de Suavização

Os principais intuítos do filtro de suavização são: 1 - supressão de ruídos e 2 - borramento, sendo este utilizado para conexão de pequenas discontinuidades, assim como

O principal problema relacionado ao filtro da média da vizinhança é o fato de que o mesmo borra as bordas, assim como outros detalhes importantes da imagem. (GONZALES, 2000).

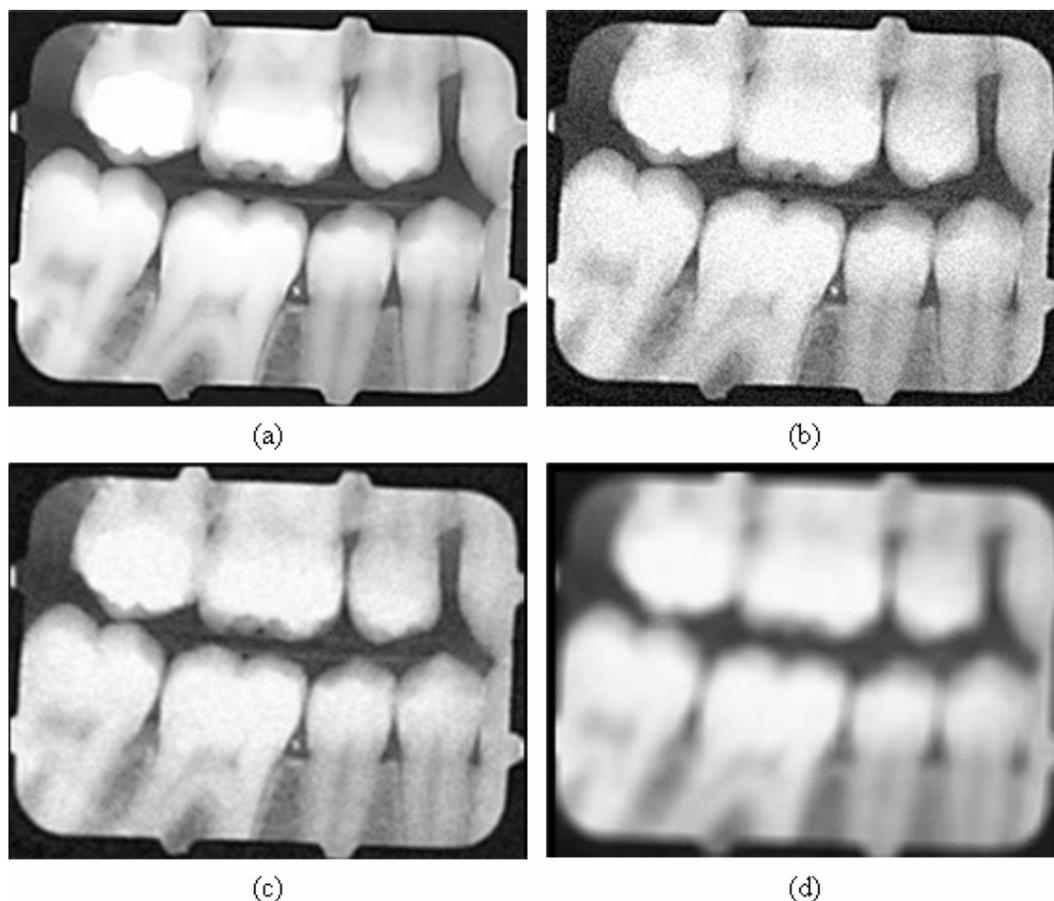


Figura 2.4 –Filtro da Média - (a) Imagem original; (b) Imagem Original com adição de ruído; (c) Resultado do filtro da média com uma máscara 3x3 (d) Resultado do filtro da média com uma máscara 9x9

Fonte: (PARANJAPE, 2000, p. 11)

Uma alternativa ao filtro da média da vizinhança é o filtro da mediana. Nesse tipo de filtro cada *pixel* é substituído pela mediana da vizinhança deste *pixel*, ao invés da média como visto anteriormente.

O filtro mediano é um método não linear para eliminação de ruído mantendo as características significativas da imagem em oposição ao filtro da média que elimina, borra partes significativas da imagem, como é o caso das bordas que representam as fronteiras das estruturas presentes na imagem. (PARANJAPE, 2000; GONZALES, 2000)

Para determinar a mediana e atribuir esse número a um determinado *pixel*, primeiro deve-se determinar o valor do *pixel* e de seus vizinhos. Assim em uma vizinhança 3 x 3, a

mediana é o quinto maior valor, logo em uma vizinhança 5×5 a mediana é o 13º maior termo. A fim de calcular a mediana deve-se ordenar e agrupar os valores do *pixel* e de seus vizinhos. (GONZALES, 2000)

A Figura 2.5 apresenta o resultado com a operação do filtro da mediana. A imagem original apresentada é (a), já em (b) tem-se a adição de ruído sobre a aquisição original e em (c) tem-se o resultado produzido do filtro da mediana com uma vizinhança de tamanho 3×3 .

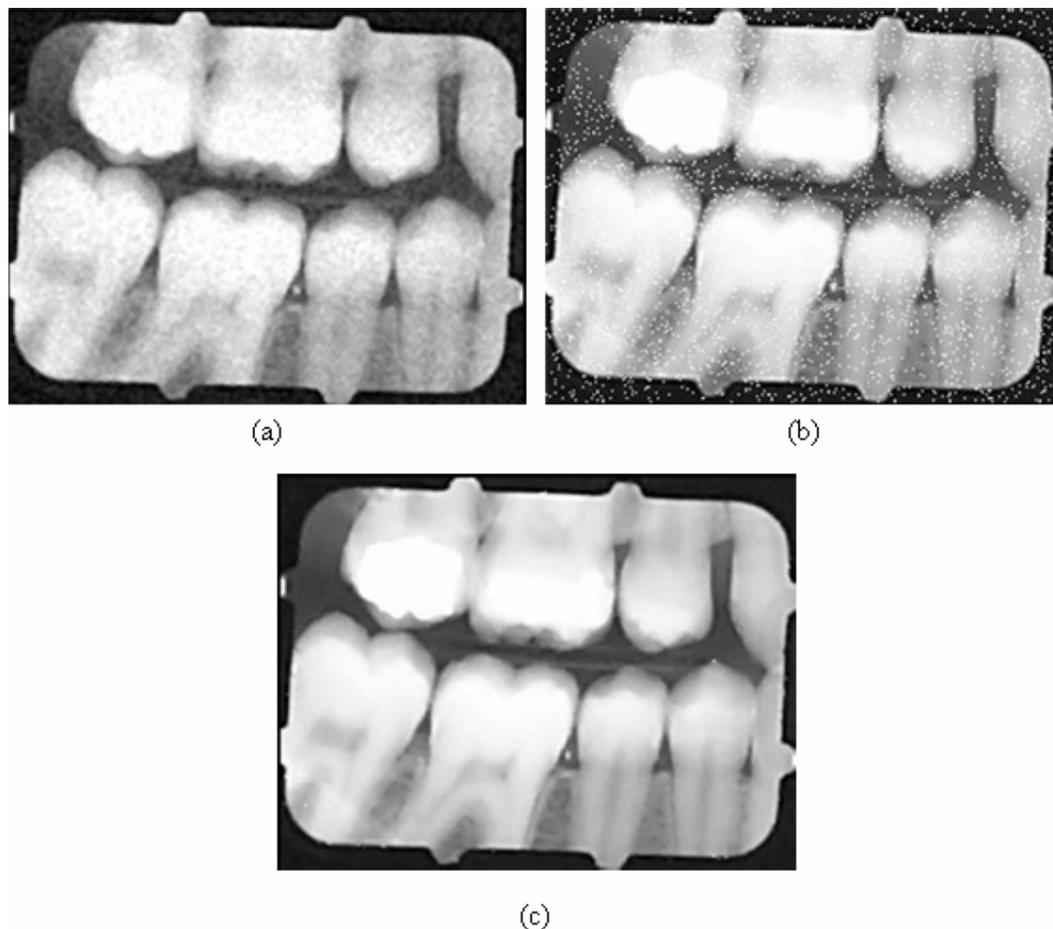


Figura 2.5 - Operação de realce com o filtro da mediana
Fonte: (PARANJAPE, 2000, p. 12)

Observa-se que o filtro da mediana preserva as características da imagem em oposição a algumas formas de realce. Na imagem produzida e apresentada em (c) o ruído foi significativamente removido, preservando as estruturas necessárias da imagem para a realização correta da etapa de diagnóstico. (PARANJAPE, 2000)

2.2.3 Subtração de Imagens

A subtração de imagens é realizada a partir de duas imagens que possuem certa similaridade. O principal objetivo desta é realçar as diferenças entre as imagens. (PARANJAPE, 2000)

Conforme Gonzales (2000, p.132), a diferença entre duas imagens $f(m,n)$ e $h(m,n)$, pode ser expressa como: $g(m,n) = f(m,n) - h(m,n)$ onde a subtração ocorre a partir do cômputo da diferença entre todos os pares de *pixels* de f e h .

Apresentado por Gonzales (2000, p.133), uma aplicação clássica desta técnica em imagens médicas ocorre quando uma máscara, $h(m,n)$, é obtida através de uma imagem de raio-x de uma parte do corpo do paciente com a utilização de um intensificador e câmera de TV, ao invés de filme de raio-x tradicional. A imagem $f(m,n)$ é uma dentre uma série de imagens obtidas da mesma região anatômica, após a injeção de corante na corrente sanguínea. O resultado de cada operação $g(m,n) = f(m,n) - h(m,n)$ em $g(m,n)$ é uma imagem com os detalhes realçados. Caso as imagens sejam capturadas em taxa de vídeo, esse mesmo procedimento pode fornecer um filme mostrando a propagação do corante nas artérias.

O exemplo expresso na Figura 2.6 **(a)** demonstra uma imagem de raio-x do topo da cabeça de um paciente antes de uma injeção de corante de iodo na rede sanguínea. Afim de adquirir esse artefato uma câmera foi posicionada acima da cabeça do paciente voltada para baixo. A mancha brilhante na terça parte inferior da imagem é o centro da coluna vertebral. Já a figura em **(b)** é o resultado da operação de subtração de imagens entre **(a)** que foi utilizada como filtro e outra imagem tirada algum tempo depois da que o corante foi introduzido. Como pode ser observado na Figura 2.6 **(b)**, os caminhos arteriais estão realçados, claras, graças ao efeito da subtração, uma vez que eles não fazem parte da máscara. (GONZALES, 2000)

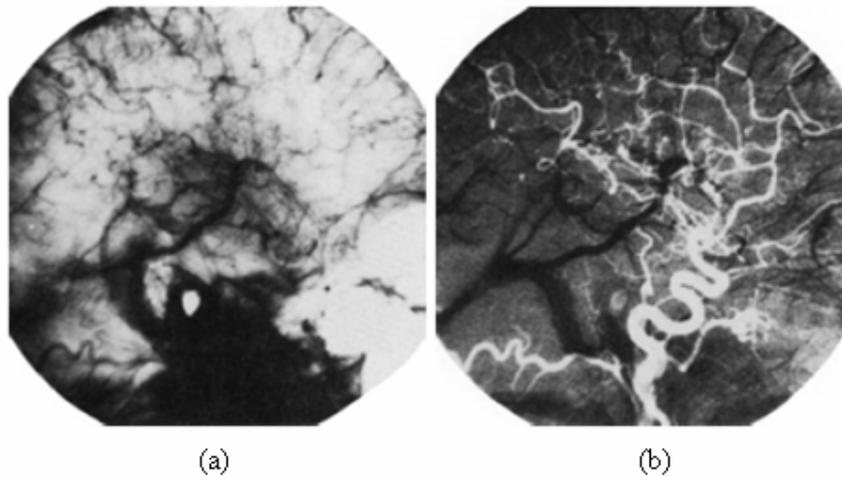


Figura 2.6 – Subtração de Imagens - (a) Máscara para realizar a técnica de subtração de imagem; (b) Imagem após injeção de corante na corrente sanguínea e com subtração da máscara

Fonte: (GONZALES, 2000, p. 133)

2.2.4 Média de Imagens

Assim como na seção anterior, esse método de realce requer mais de uma imagem da mesma cena para realizar a operação esperada.

Conforme apresentado por Paranjape (2000, p. 15), pode-se utilizar a supressão de ruído através do cálculo da média de imagens a partir de três suposições básicas: 1- quando pode-se contar com um número relativamente grande de imagens para realizar a operação; 2 – todas as imagens devem ser afetadas pelo mesmo tipo de ruído; 3 – o ruído possui média zero, ocorrência randômica. Quando tais suposições estiverem atendidas pode ser vantajosa à aquisição de várias imagens com o intuito de realizar o cálculo da média das imagens.

$$g(m,n) = \frac{1}{Q} \sum_{i=1}^Q a_i(m,n) \quad (2.3)$$

Tem-se assim que

$$E\{g(m,n)\} = f(m,n) \quad (2.4)$$

e

$$\sigma_g = \frac{\sigma_d}{\sqrt{Q}} \quad (2.5)$$

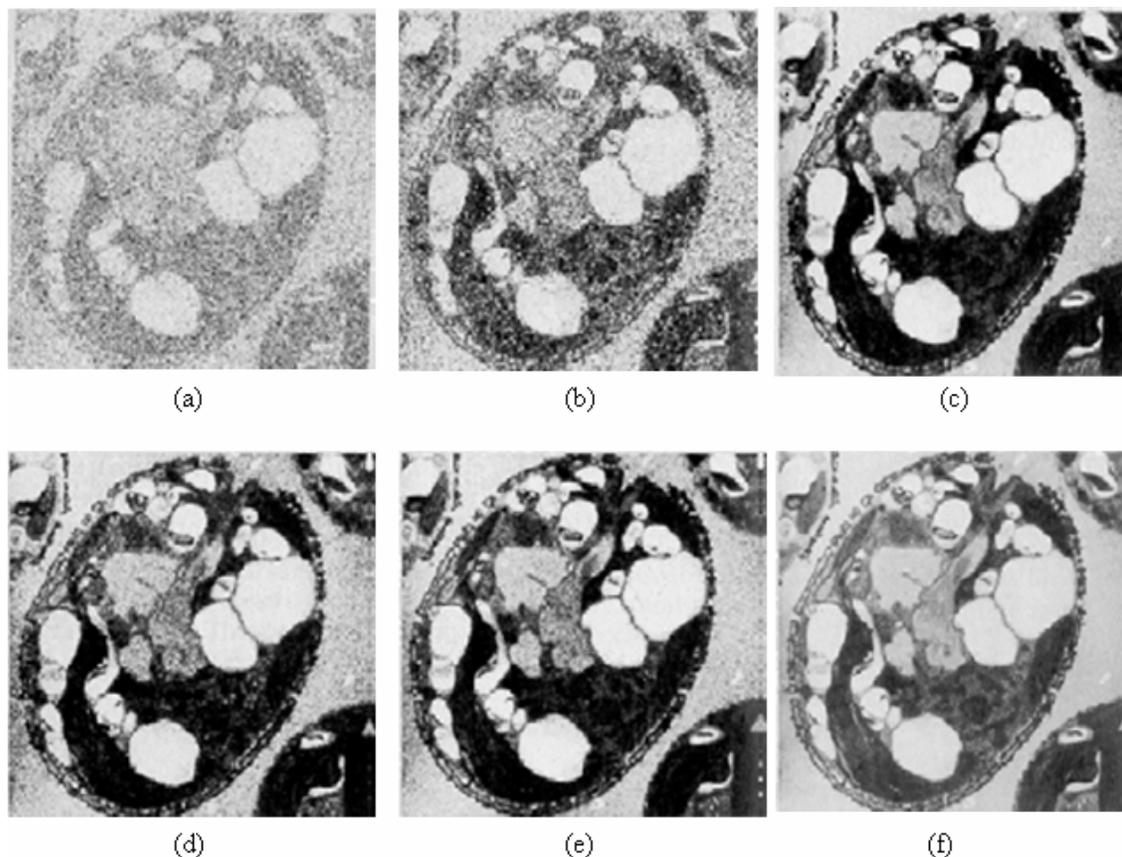


Figura 2.7 - Exemplo de redução de ruído através de médias

Fonte: (GONZALES, 2000, p. 135)

A imagem anterior exemplifica a aplicação do filtro da média. A figura (a) é uma típica imagem ruidosa, já (b) até (f) são os resultados dos filtros com 2, 8, 16, 32 e 128 imagens ruidosas. Observa-se que com um número maior de imagens melhor será o resultado do filtro da média. (GONZALES, 2000, 134-135)

2.3 Técnicas de Segmentação

A segmentação é a separação das estruturas de interesse do fundo, sendo esse processo essencial para a análise e processamento de imagens. Inúmeros algoritmos foram desenvolvidos com esse propósito. A segmentação de um determinado objeto é conseguida através da identificação dos *pixels* ou *voxels* que pertencem ao objeto ou encontrando aqueles que representam o limite da estrutura. (BANKMAN, 2000)

O objetivo principal do processo de segmentação é dividir uma imagem em regiões, também chamadas classes, ou subconjuntos, (ROGOWSKA, 2000). Essas regiões devem ser homogêneas respeitando uma ou mais características. (GONZALES, 2000)

A segmentação é uma importante ferramenta para as imagens médicas, sendo aplicada em diversas situações tais como detecção das bordas coronárias em angiogramas, quantificação das lesões em esclerose múltipla, simulações cirúrgicas, medição do volume de tumores, planejamento cirúrgico, verificação da resposta a uma dada terapia, detecção de tumores, entre outros. (ROGOWSKA, 2000)

A extração das características de imagens é importante dado o intuito posterior de extrair medidas das estruturas segmentadas. Em algumas aplicações a etapa de segmentação pode classificar as regiões de forma anatômica tais como os ossos e músculos, já em outras, em regiões patológicas como o câncer. Em alguns estudos o objetivo é dividir a imagem inteira em sub-regiões tais como a matéria branca e a matéria cinzenta no caso do cérebro. Existem várias utilidades para o uso dos métodos de segmentação. (ROGOWSKA, 2000)

As técnicas de segmentação podem ser manuais, quando um operador é responsável por fornecer os dados de entrada, ou ainda automáticas, quando os dados são informados de forma autônoma. Esse capítulo apresenta os modelos ditos clássicos (ROGOWSKA, 2000), (limiar, métodos baseados em bordas e crescimento de região) assim como o algoritmo de *Watershed* e uma introdução aos modelos deformáveis.

2.3.1 Limiarização – *Thresholding*

Conforme Gonzales (2000, p. 316) limiarização é uma das mais importantes abordagens na segmentação de imagens. A limiarização consiste em separar objetos do fundo através da seleção de um limiar T .

Diversas técnicas de limiarização foram desenvolvidas. Enquanto algumas se utilizam do histograma da imagem outras são baseadas em propriedades locais. A aproximação mais intuitiva é a limiarização global, sendo a que possui um único ponto selecionado na imagem baseado no histograma da mesma, enquanto que as baseadas em propriedades locais, tais como, valores médios locais do nível de cinza são denominados limiarização local e são dinâmicos ou adaptáveis caso sejam selecionados mais de um limiar independente para cada grupo de *pixels*. (ROGOWSKA, 2000)

Conforme dito anteriormente, limiarização global é aquela que utiliza o histograma de imagem e através deste consegue separar um objeto alvo do fundo da imagem com a seleção de um único limiar, conforme ilustrado na Figura 2.8.

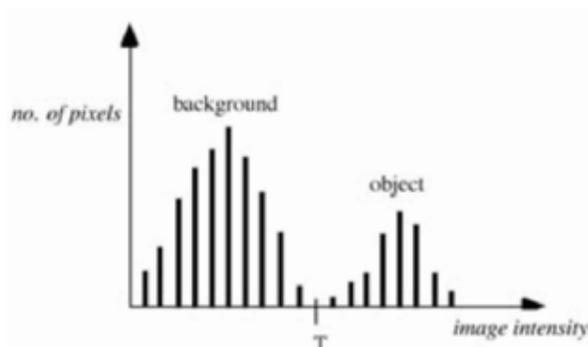


Figura 2.8 - Exemplo de um histograma propício à aplicação da técnica de limiarização
 Fonte: (ROGOWSKA, 2000, p. 70)

Na Figura 2.8 observa-se que para separar o objeto do fundo basta selecionar um limiar T adequado.

Uma imagem limiarizada $g(m,n)$ é definida como:

$$g(m,n) = \begin{cases} 1 & \text{se } f(m,n) > T \\ 0 & \text{se } f(m,n) \leq T \end{cases} \quad (2.6)$$

Os *pixels* rotulados com o valor “um” correspondem aos objetos segmentados e “zero” representa o fundo (GONZALES, 2000), logo nesse caso tem-se uma imagem binária (ROGOWSKA, 2000), contudo pode-se selecionar outros níveis de cinza convenientemente ao invés de utilizar zero e um.

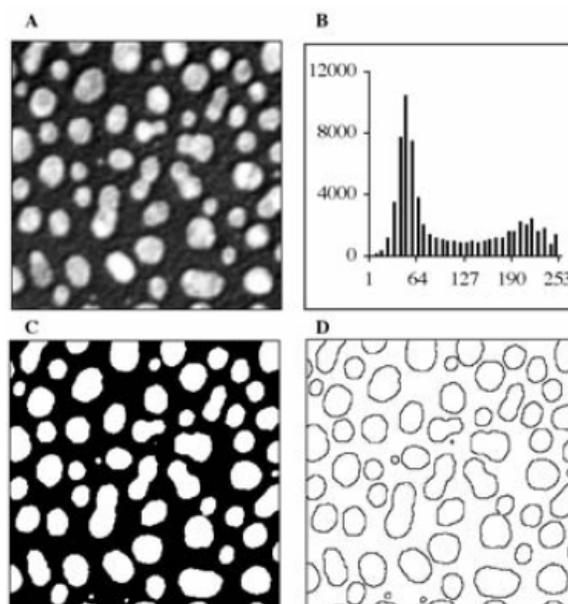


Figura 2.9 - Exemplo de segmentação de células com a técnica de limiar
 Fonte: (ROGOWSKA, 2000, p. 71)

A Figura 2.9 demonstra a aplicação da técnica de limiarização global, onde: **(A)** é a imagem original com células em branco e o fundo preto, sendo **(B)** o histograma e **(C)** a imagem depois da técnica de limiarização global com $T = 127$ e **(D)** após ser aplicado uma máscara Laplaciana 3x3 na imagem **(C)**.

Técnicas de limiar, como a limiarização global, são de complexidade simples e rápida. Imagens que possuem objetos uniformes na intensidade e fundo contrastando com o alvo a ser segmentado são candidatas a esse tipo de técnica, contudo se existir um baixo contraste entre a estrutura e o fundo, ruído, ou ainda se a intensidade do fundo varia significativamente, esse método de segmentação não produzirá resultados satisfatórios. (ROGOWSKA, 2000)

2.3.2 Métodos baseados em detecção de bordas

Os métodos baseados nessa abordagem são os mais comumente utilizados, com a finalidade de detectar discontinuidades significativas em imagens em tons de cinza.

Os modelos baseados em bordas são análogos aos filtros de aguçamento, assim denominados na literatura, aja visto que esses compartilham algoritmos e ainda objetivos tais como o realce de bordas. Esse trabalho prefere abordar tal assunto na seção de segmentação.

Define-se borda como o limite entre duas regiões com propriedades distintas de níveis de cinza. Basicamente as técnicas de detecção de bordas utilizam-se da computação de um operador local diferencial (GONZALES, 2000), por exemplo, o gradiente. (ROGOWSKA, 2000)

A maioria dos métodos que utiliza operadores de gradiente para detecção de bordas realiza as operações a partir de máscaras, sendo estas convolucionadas com a imagem original. (ROGOWSKA, 2000). Um exemplo da manipulação de gradiente é o operador de Sobel. (GONZALES, 2000)

$$\begin{array}{cc} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} & \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \\ \text{(a)} & \text{(b)} \end{array}$$

Figura 2.10 - Típicas máscaras do operador de Sobel
Fonte: (ROGOWSKA, 2000, p. 75)

A primeira máscara é usada para computar G_x já a segunda é utilizada para G_y . O cálculo do valor do gradiente é utilizado para combinar essas máscaras, conforme apresentado pela próxima equação. (ROGOWSKA, 2000)

$$|G| = \sqrt{|G_x|^2 + |G_y|^2} = \sqrt{\left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]} \quad (2.7)$$

$$D = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

Além do operador de Sobel existem vários operadores de funcionamento similar que postulam a detecção da borda em imagens, dentre esses pode-se citar: Roberts, Prewitt, Robinson, Kirsch e Frei-Chen.

$$\begin{array}{ccc} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} & \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} & \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \\ \text{(a)} & \text{(b)} & \text{(c)} \end{array}$$

Figura 2.11 - Máscaras para o operador Laplaciano

Fonte: (ROGOWSKA, 2000, p. 76)

O operador Laplaciano é aplicado de forma similar aos filtros que operam sobre o gradiente, entretanto o filtro Laplaciano é computado através da segunda derivada (ROGOWSKA, 2000) ao invés da primeira como os de gradiente. A imagem anterior apresenta algumas máscaras típicas para operações com o operador de Laplace.

Por fim, outro método amplamente utilizado foi proposto por (CANNY, 1986). Esse modelo produz um resultado similar ao modelo de Sobel (RUSS, 1998), dada à característica de processamento composto por linhas e colunas da imagem convoluídos com Gaussianas unidimensionais e suas derivadas nos eixos x e y . Portanto, assim como em Sobel, são extraídas as bordas que representam linhas e as colunas individualmente, e finalmente realizada uma composição do resultado. (CARVALHO, 2005)

2.3.3 Métodos baseados no crescimento de regiões

Esses métodos utilizam um ponto semente para cada objeto de interesse. Após a seleção da semente, que pode ser executada de forma automática ou indicada por um operador, o algoritmo examina a vizinhança do ponto semente em busca dos *pixels* que pertençam a determinado critério de uniformidade, também conhecido como critério de

homogeneidade, com a semente previamente selecionada. O processo é contínuo até que não existam mais *pixels* capazes de se unir à região através do critério de homogeneidade.

Defino por Gonzales (2000, p. 326) a formulação básica pode ser dada por:

$$(a) \bigcup_{i=1}^n R_i = R,$$

$$(b) R_i \text{ é uma região conexa, } i = 1, 2, \dots, n,$$

$$(c) R_i \cap R_j = \emptyset \text{ para todo } i \text{ e } j, i \neq j,$$

$$(d) P(R_i) = \text{VERDADEIRO para } i = 1, 2, \dots, n, e$$

$$(e) P(R_i \cup R_j) = \text{FALSO para } i \neq j,$$

Onde $P(R_i)$ é um predicado lógico sobre os pontos do conjunto R_i e \emptyset é o conjunto vazio. Logo:

(a) Cada *pixel* deve pertencer à mesma região;

(b) Requer que os *pixels* de uma mesma região sejam conexos;

(c) Indica que as regiões devem ser disjuntas;

(d) Teste de uniformidade, por exemplo, $P(R_i) = \text{VERDADEIRO}$ se todos os *pixels* em R_i possuírem a mesma intensidade;

(e) Indica que as regiões R_i e R_j são diferentes no sentido do predicado P .

Existem vários algoritmos baseados na técnica de crescimento de região. A Figura 2.12 mostra as etapas da segmentação usando o algoritmo de *hill-climbing* (ROGOWSKA, 2000). A técnica foi aplicada com sucesso em mamografias. As vantagens desta é que não necessita da seleção de um ponto semente inicial, e cresce da região das bordas para o centro, evitando um crescimento excessivo. (ROGOWSKA, 2000)

A primeira parte, representada pela Figura 2.12 (A) é a imagem original dada como entrada, (B) são os 16 pontos da borda determinados pelo algoritmo, (C) segmentação da região (D) bordas destacadas durante o processo de segmentação.

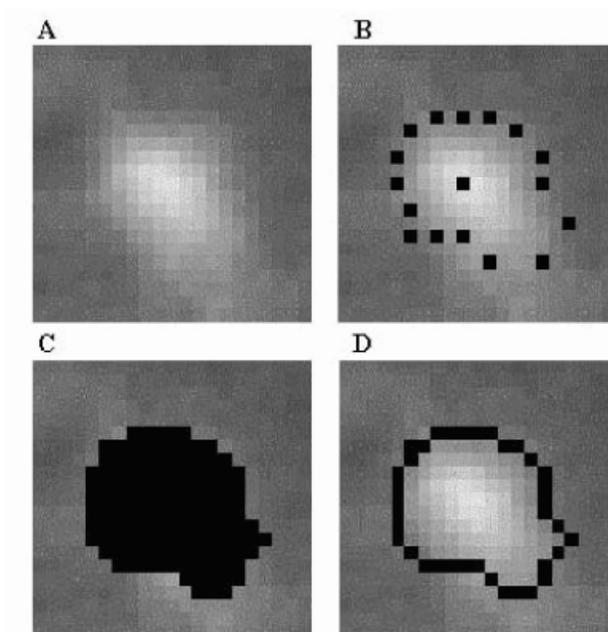


Figura 2.12 - Segmentação pelo algoritmo de *hill-climbing*
 Fonte: (ROGOWSKA, 2000, p. 74)

Várias aplicações em imagens médicas utilizam-se do modelo de crescimento da região, tais como a segmentação de importantes estruturas em imagens cardíacas, angiografias e ainda extração de regiões da superfície do cérebro. (ROGOWSKA, 2000)

2.3.4 Algoritmo *Watershed*

O *Watershed* é uma ferramenta rápida para a segmentação de imagens. Foi derivada originalmente da observação do modelo geofísico da chuva, proposta por Beucher and Lantuéjoul publicado no trabalho “*Use of watersheds in contour detection*” no *Workshop Image Processing* realizado na França em setembro de 1979. (BEARE, 2006)

O algoritmo é baseado na idéia que um pingo de chuva cai na superfície e “escorrega” em direção a parte mais íngreme até encontrar um vale que será inundado, o resultado será uma imagem que deve ser considerada terreno, dividida em bacias onde cada uma dessas conteria um objeto de interesse. (BEARE, 2006)

Outra abordagem simples para entender o funcionamento dessa técnica é pensar na imagem como uma superfície onde os *pixels* brilhantes representam montanhas e os mais escuros vales, então os vales ficam submersos lentamente por água, como em uma chuva, não sendo permitido que as águas oriundas de diferentes vales misturem-se. Conseqüentemente, existe a necessidade de criar represas nos pontos de contato, tais represas representam os limites das bacias, assim como também às dos objetos da imagem. (ROGOWSKA, 2000)

O artefato produzido pelo processo de segmentação através da técnica de *watershed* é uma imagem dividida em categorias, onde cada objeto é etiquetado. Dada uma imagem que após o processo de segmentação tenha resultado em duas regiões, os *pixels* pertencentes à primeira região devem possuir um valor de intensidade diferente dos valores da segunda região, tal exemplo fica mais evidente se observada a construção do ensaio prático disposto na seção três desse trabalho. A Figura 2.13 também ilustra a utilização desse método, onde **(a)** é a imagem original, enquanto que **(b)** é a imagem após o processo de segmentação com o auxílio do *Insight Toolkit* (ITK). (BEARE, 2006) Observe as etiquetas geradas, uma de cada cor, representando as diversas regiões segmentadas.

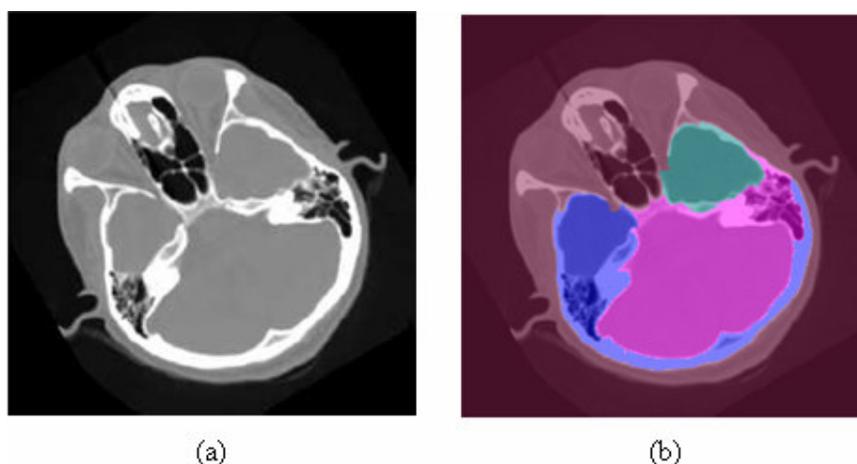


Figura 2.13 - Exemplo de segmentação com o algoritmo de watershed
Fonte: (BEARE, 2006, p. 4)

É muito raro conseguir uma segmentação útil aos propósitos funcionais usando um único procedimento, mesmo sob circunstâncias favoráveis. Normalmente os processos de segmentação bem sucedidos fazem uso de uma combinação de diversas técnicas para selecionar as estruturas alvo em uma imagem médica. Uma combinação de técnicas que pode ser aplicada nesse caso é a utilização de detector de borda, com o intuito de produzir limites de objetos brilhantes, possibilitando assim uma melhor atuação do *watershed*. (BEARE, 2006)

A segmentação por *watershed* é uma poderosa ferramenta para aplicações em imagens 2D. Um exemplo é a segmentação de imagens microscópicas da retina humana. Existem pesquisas que se propõem a aplicar extensões desse tipo de segmentação em imagens tridimensionais como é o caso da afeição de volumes em imagens cardíacas. (ROGOWSKA, 2000)

2.3.5 Modelos Deformáveis

Nas últimas quatro décadas a segmentação com auxílio computacional tornou-se indispensável e muito importante para selecionar as estruturas em imagens médicas. (XU, 2000)

Para diminuir as dificuldades encontradas com o uso de alguns modelos de segmentação, os modelos deformáveis são estudados e aplicados intensamente na segmentação de imagens médicas, tendo resultados satisfatórios. (XU, 2000)

Modelos deformáveis são curvas ou superfícies definidas dentro de um domínio da imagem que se movem sob a influência de forças internas definidas na própria curva e das forças externas que são computadas a partir de dados provenientes da imagem. As forças internas são projetadas para manter o modelo liso durante a deformação, enquanto que as forças externas são definidas para mover o modelo até o limite do objeto a ser segmentado. (XU, 2000)

A popularidade dos modelos deformáveis deve-se a Kass, Witkin, e Terzopoulos pelo trabalho (KASS, 1987). A partir desta publicação os modelos deformáveis destacaram-se como sendo uma das áreas mais ativas e bem sucedidas da segmentação de imagens. (XU, 2000)

Existem basicamente dois tipos de modelos deformáveis: os paramétricos e geométricos. Os modelos paramétricos permitem uma interação direta, o que leva a uma representação compacta em uma execução de tempo real rápida, com parâmetros que representam curva e superfícies de forma explícita. Neste caso, a adaptação topológica desse modelo, sobre estruturas que se fundem ou separam-se é de difícil aplicação. Já os modelos geométricos adaptam-se naturalmente as mudanças topológicas, sendo baseados na teoria da evolução da curva e do *level set method* que representa curvas e superfícies de forma implícita. Apesar da diferença fundamental, os princípios subjacentes de ambos os métodos são muito similares. (XU, 2000)

O estudo publicado “*A Summary of Geometric Level-Set Analogues for a General Class of Parametric Active Contour and Surface Models*” (XU, 2001) estabelece relacionamento matemáticos entre os dois mundos dos modelos deformáveis, geométricos e paramétricos. No referido artigo são apresentados os relacionamentos matemáticos precisos, tendo como resultado uma formulação geométrica geral.

Os modelos deformáveis oferecem robustez à existência de ruído e ressaltam os limites dos objetos de forma confiável e consistente. (XU, 2000)

2.4 Quantificação

A análise de imagens médicas beneficia significativamente áreas onde se faz necessária medidas precisas, rápidas e repetitivas através do auxílio de recursos computacionais. As medidas quantitativas contribuem para a correta análise de estruturas possibilitando a detecção e a classificação de casos anormais, assim como os normais, vários dados são utilizados para realizar a correta medição quantitativa, tais como forma do tecido, tamanho, textura, densidade, entre outros. Diversas técnicas computacionais são aplicadas para fornecer medidas quantitativas com a finalidade de auxiliar na interpretação e contribuir para a realização dos exames necessários.

O *Compactness* é uma técnica para quantificar estruturas presentes, por exemplo, em mamografias a fim de verificar se os objetos são nódulos benignos ou malignos. Esse modelo matemático verifica se a medida calculada é similar a um círculo. (BANKMAN, SPISZ e PAVLOPOULOS, 2000)

Uma forma comum de se obter o Compactness é através da computação do perímetro P e da área A . Assim:

$$C = \frac{P^2}{A} \quad (2.8)$$

Quando C é 4π significa que a medida representa um círculo perfeito, quando essa técnica é aplicada na prática, valores mais altos podem ser considerados tais como $C = 13,6$. (BANKMAN, SPISZ e PAVLOPOULOS, 2000)

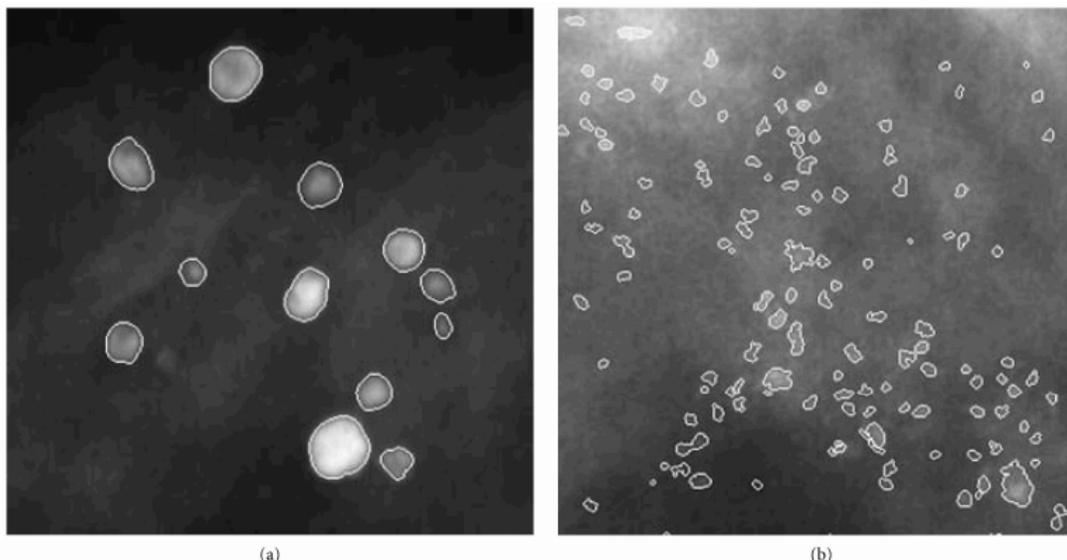


Figura 2.14 - Exemplo de uma mamografia
 Fonte: (BANKMAN, SPISZ e PAVLOPOULOS, 2000, p. 216)

Outro exemplo é normalizar na seguinte forma: $C' = 1 - 4\pi / C$ variando entre zero e um. O uso dessa técnica é demonstrado na imagem anterior, onde a Figura 2.14 (a) é benigna já (b) ilustra microcalcificações malignas apresentadas na mamografia. (BANKMAN, SPISZ e PAVLOPOULOS, 2000)

Sensores capazes de criar visualizações 3D estão se proliferando principalmente na área médica. Tais sensores operam no plano criando múltiplas fatias que podem ser tratadas como dados de volume, sendo que algumas permitem a reconstrução tridimensional. A habilidade limitada da visão humana não permite que os dados sejam investigados com segurança, já a análise computadorizada permite que os dados sejam processados sem as limitações humanas. (KOVALEV, 2000)

Inúmeras informações podem ser calculadas tendo como entrada imagens com estruturas 3D. Várias características das imagens médicas são estudadas objetivando a criação de indicadores que possibilitem diagnósticos e prognósticos eficientes com valores clínicos. Dentre essas características citam-se a análise de texturas das imagens, distribuição uniforme do volume, entre outros. (KOVALEV, 2000)

No caso específico do coração, como este é um órgão com uma função mecânica os principais dados quantitativos extraídos das imagens médicas para a investigação clínica deste focam-se na verificação do seu desempenho hemodinâmico, dada à função de “empurrar” mecanicamente volumes de sangue através do sistema vascular. (RATIB, 2000)

A verificação do volume do ventrículo esquerdo do coração é uma das medidas mais comuns aplicadas em imagens médicas digitais do coração. Graças ao avanço das técnicas computacionais a medição do volume a partir de imagens obtidas em exames é de fundamental relevância clínica. O cálculo geométrico do volume ventricular ocorre a partir da identificação apropriada das bordas e das cavidades ventriculares nas imagens. Diversas técnicas com este intuito foram propostas, tanto automáticas como semi-automáticas. Modelos de análise quantitativas das funções cardíacas podem ser vistas em (RATIB, 2000) outro estudo que postula sobre técnicas de processamento e análise de imagens médicas aplicadas para quantificar dados de *Tagged Cardiac* em MRI pode ser obtido em (KERWIN, 2000).

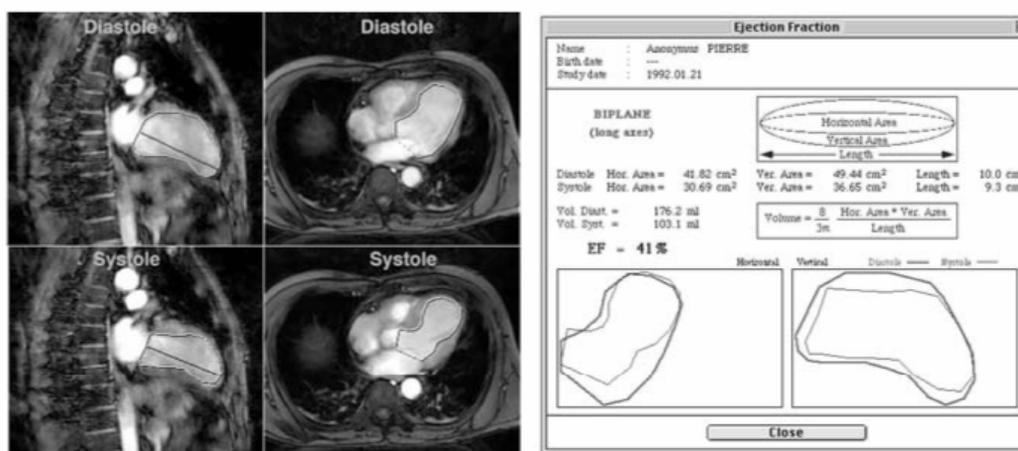


Figura 2.15 - Cálculo da média do volume do ventrículo esquerdo

Fonte: (RATIB, 2000, p. 362)

A Figura 2.15 demonstra o cálculo de importantes indicadores que possibilitam a correta investigação durante a etapa de diagnóstico. Neste caso, depois que os limites de interesse são encontrados, as bordas, os volumes *end-diastolic* e *end-systolic* podem ser obtidos e o volume pode ser calculado. Um dos métodos extensamente utilizado nesse caso é baseado no comprimento da área, de forma que o ventrículo esquerdo seja modelado como um elipsóide. (RATIB, 2000)

Essa sessão com o breve estudo direcionado a quantificação de estruturas presentes em imagens médicas demonstra que existem inúmeros métodos para computar dados de importância para a área médica. A técnica aplicada varia conforme o contexto de aplicações, contudo nas técnicas estudadas relacionam cálculos que inferem sobre a área, volume e distribuição dos alvos nas imagens médicas. Tais propriedades são obtidas através de artefatos

segmentados em regiões de interesse, assim como nas características presentes nessas como à textura entre outras.

Na busca de criar uma representação prática dos elementos teóricos abordados nessa seção, o próximo capítulo apresenta uma série de ensaios práticos realizados com imagens médicas.

3 APLICAÇÃO PRÁTICA DE CONCEITOS BÁSICOS EM IMAGENS MÉDICAS

Esse capítulo apresenta testes realizados com algoritmos de segmentação e realce. Esses testes são executados com ferramentas já utilizadas para os mesmos propósitos, os ensaios aqui representando são construções e reproduções baseadas em estudos de áreas correlatas a deste trabalho. Além disso, são demonstradas simulações com imagens próximas a que será adquirida junto ao estudo biomédico apresentado e delineada no capítulo quatro.

3.1 Ferramentas

Com a finalidade de realizar os testes e ensaios apresentados nessa seção utilizaram-se duas ferramentas difundidas no meio científico no qual este estudo se enquadra. Estas são: o *Insight Toolkit* (ITK) (IBÁÑEZ, 2005) e o MATLAB (MATLAB, 2007).

O MATLAB é um sistema interativo de fácil manipulação que permite formular soluções para diversos problemas, especialmente aqueles que envolvem matriz. Com ele pode-se desenvolver soluções específicas de maneira rápida se comparada ao desenvolvimento dessas mesmas em linguagens como o *C* ou *Fortran*. O nome MATLAB é oriundo de *matrix laboratory*. Foi concebido originalmente para realizar cálculos com matrizes. Hoje o MATLAB possui diversas bibliotecas para atuar em diferentes áreas da computação científica. Em ambientes universitários, o MATLAB é usado com frequência para pesquisas e desenvolvimento. Existe uma versão deste produto para estudante que pode ser adquirida com descontos significativos quando comparado a outros modelos de licenciamento do mesmo produto. (GONZALES, 2002)

Em 1999 o *National Library of Medicine* do *National Institutes of Health* contratou o desenvolvimento de uma biblioteca *open-source* tendo como objetivo a segmentação e o registro de imagens. Hoje essa biblioteca é conhecida como *Insight Toolkit* (ITK). Inicialmente participaram do consórcio de desenvolvimento seis instituições das quais três são

universidades e as demais empresas comerciais. As universidades participantes são: *University of North Carolina (UNC)*, *University of Tennessee (UT)* e *University of Pennsylvania (UPenn)*. As empresas comerciais que compõem o consórcio são: *GE Corporate R&D*, *Kitware Inc.* e *MathSoft* (agora conhecida como *Insightful*). No ano de 2002 ocorreu a primeira liberação pública e oficial do ITK. (IBÁÑEZ, 2005) Os algoritmos utilizados nos ensaios práticos realizados com o ITK são baseados nos exemplos apresentados em (IBÁÑEZ, 2005) o qual reproduz o funcionamento dos diversos filtros e funcionalidades presentes no ITK.

3.2 Ensaios

Nas próximas seções são apresentados alguns modelos de segmentação, diversas vezes técnicas de realce e segmentação são combinadas, conforme a natureza do teste realizado, objetivando a extração ou o realce eficiente das estruturas alvo.

3.2.1 Realce

Para estudar a prática dos modelos de realce de imagens médicas realizaram-se testes com as técnicas de equalização do histograma e média.

Objetivando a equalização do histograma da imagem foi construído um algoritmo no MATLAB conforme segue:

```
clear all;  
  
I = imread('C:\EnsaioTC\realce\original.png');  
  
I1 = histeq(I);  
  
imshow([I, I1]);
```

O resultado é demonstrado na Figura 3.1 e pode-se observar que pontos antes escuros foram atenuados.

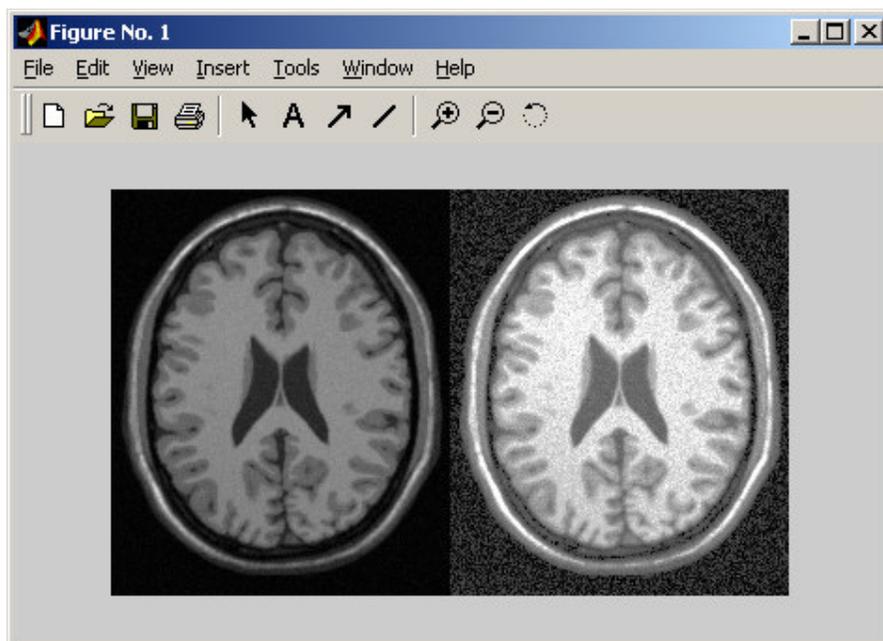


Figura 3.1 - Resultado do ensaio da equalização do histograma

A atenuação nos níveis de intensidade do *pixel* pode ser comprovada com a análise do histograma da imagem, conforme observado na Figura 3.2.

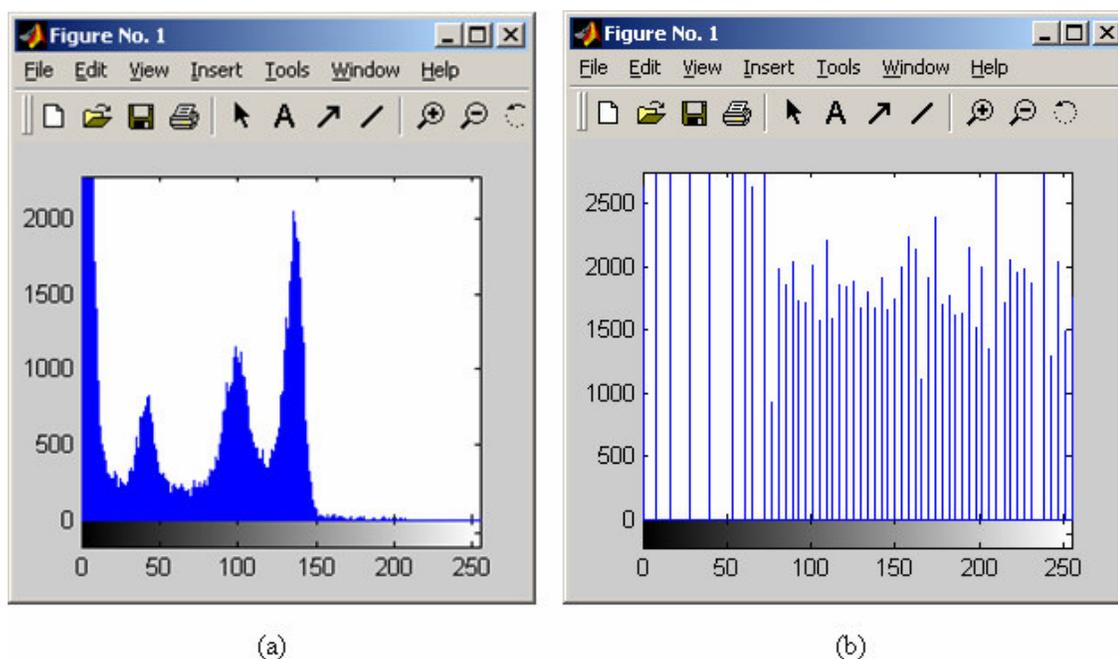


Figura 3.2 - Histogramas do ensaio

Observa-se que o primeiro histograma representado na imagem anterior em **(a)** não apresenta uma distribuição uniforme como à apresentada em **(b)**.

Para testar o algoritmo do filtro da média a ferramenta selecionada foi o ITK e os resultados estão apresentados nas próximas figuras.

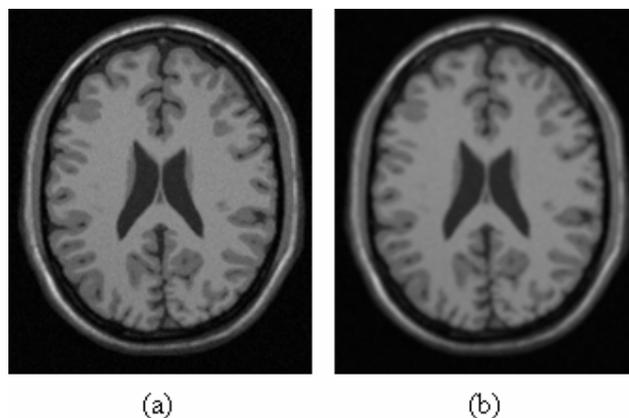


Figura 3.3 - Ensaio da Média

Conforme apresentado no capítulo de realce, o algoritmo da média é usado para suavização e objetiva borrar as estruturas da imagem para realçar ou até mesmo unir pequenas descontinuidades, essas funcionalidades podem ser visualizadas na figura anterior. O algoritmo da média encontra-se no Anexo A.

3.2.2 Segmentação

Com o objetivo de estudar o funcionamento prático dos algoritmos de segmentação, essa seção demonstra os testes executados com os mesmos, aqui algumas estruturas consideradas importantes foram selecionadas através de técnicas de segmentação.

3.2.2.1 Algoritmo *Connected Threshold*

Este algoritmo pertence ao modelo de crescimento de regiões. O critério usado para verificar se um *pixel* pertence ou não a uma determinada região encontra-se baseado em um intervalo dos valores que representam a maior e a menor intensidade, onde tais valores são fornecidos pelo usuário. Aqueles *pixels* cuja intensidade está dentro do intervalo são considerados parte do objeto a ser segmentado. (IBÁÑEZ, 2005)

Eventuais ruídos na imagem podem dificultar a segmentação das estruturas alvo, portanto é conveniente realizar etapas de pré-processamento com a finalidade de realçar os dados da imagem. (IBÁÑEZ, 2005)

A próxima tabela expõe os dados que foram utilizados como entrada para segmentar as imagens demonstradas a seguir.

A iniciação do algoritmo requer do usuário um ponto semente, que conveniente deve estar colocado na região da estrutura anatômica a ser segmentada, bem como o valor de

máximo e mínimo do limiar, que também devem ser fornecidos como entrada ao algoritmo de segmentação.

Tabela 3.1 - Descrição dos dados de entrada para o algoritmo de *Connected Threshold*

Estrutura	Ponto Semente	Limiar	
		Mínimo	Máximo
Camada Branca (<i>white matter</i>)	60,116	150	180
Ventrículo (<i>ventricle</i>)	81,112	210	250
Camada Cinzenta (<i>gray matter</i>)	48,67	180	210

Fonte: Adaptado de (IBÁÑEZ, 2005, p. 496)

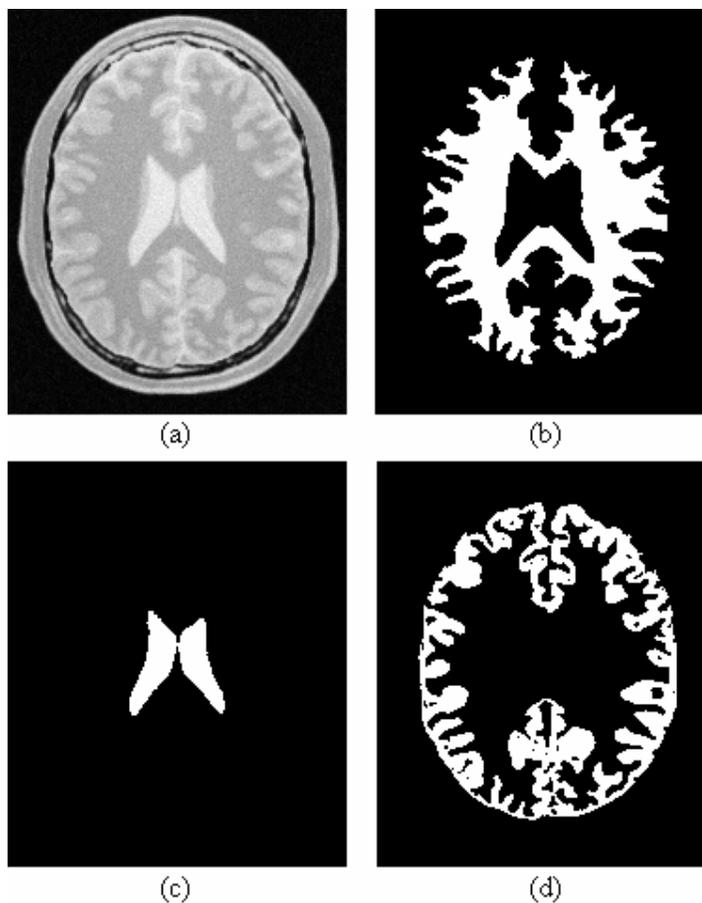


Figura 3.4 - Imagem original e resultados obtidos com a técnica de *Connected Threshold*

Para reproduzir os resultados dispostos em (IBÁÑEZ, 2005) foram adequados o ponto semente da camada cinzenta com o intuito de encontrar resultados próximos aos encontrados na referida literatura.

A Figura 3.4 demonstra os resultados dos testes onde **(a)** é a imagem original obtida a partir de material digital junto a (IBÁÑEZ, 2005), **(b)**, **(c)** e **(d)** são os resultados obtidos depois da implementação do algoritmo de segmentação apresentado no Anexo B, que respectivamente representam: camada branca, ventrículo, camada cinzenta.

O sucesso da segmentação com esse algoritmo está diretamente relacionado à eficiente distribuição homogênea da intensidade dos *pixels*. Um futuro ensaio a ser realizado é o pré-processamento através da equalização histograma e posterior segmentação.

3.2.2.2 Algoritmo *Neighborhood Connected*

Esse é uma variante do algoritmo *Connected Threshold*. Em ambos o usuário fornece como entrada um ponto semente e os valores máximo e mínimo do limiar, e o que difere o *Neighborhood Connected* é o fato do mesmo requerer o tamanho de uma vizinhança. (IBÁÑEZ, 2005)

Com o objetivo de verificar se um determinado ponto pertence ou não a uma região a ser segmentada, esse método investiga se o *pixel* possui a intensidade entre os pontos máximo e mínimo fornecidos pelo usuário. Além do *pixel* atual, os da vizinhança, de tamanho previamente definido, devem possuir seus limiares entre os valores estipulados. (IBÁÑEZ, 2005)

Tabela 3.2 - Descrição dos dados de entrada para o algoritmo de *Neighborhood Connected*

Estrutura	Ponto Semente	Limiar	
		Mínimo	Máximo
Camada Branca (<i>white matter</i>)	60,116	150	180
Ventrículo (<i>ventricle</i>)	81,112	210	250

Fonte: Adaptado de (IBÁÑEZ, 2005, p. 503)

O tamanho da vizinhança é uma informação primordial para o sucesso dessa técnica, visto que quanto maior for à vizinhança menor será o efeito do ruído, contudo grandes vizinhanças requerem mais poder computacional. Esse algoritmo promove ganho sobre seu antecessor no momento em que a imagem de entrada possui estruturas atômicas pequenas, visto que com o ajuste do tamanho da vizinhança essas podem ser computadas de forma

adequada pelo algoritmo, produzindo uma segmentação de melhor qualidade. (IBÁÑEZ, 2005) O resultado é apresentado na Figura 3.5.

A Figura 3.5 (a) é a imagem original, as demais apresentadas em (b) e (c) são resultados da segmentação da camada branca e do ventrículo. A máscara usada para vizinhança foi de 5 x 5 em ambos os testes executados com essa técnica.

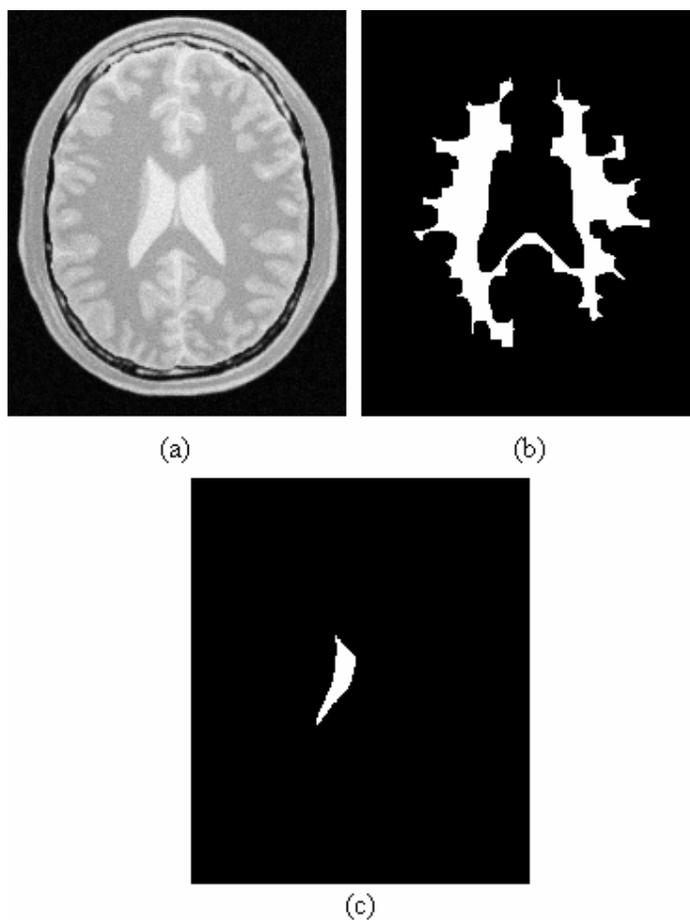


Figura 3.5 - Imagem original e resultados obtidos com a técnica de *Neighborhood Connected*

A fim de observar a qualidade do presente algoritmo em selecionar pequenos alvos nos artefatos de entrada, foi executado um teste objetivando selecionar somente a segunda parte do ventrículo, para tanto o limites de limiar são: 210 e 250. Já o ponto semente utilizado foi (101,112). Esse ensaio pode ser observado na Figura 3.6. O algoritmo de *Neighborhood Connected* usado para obter esse resultado assim como o observado na Figura 3.5 é apresentado no Anexo C.

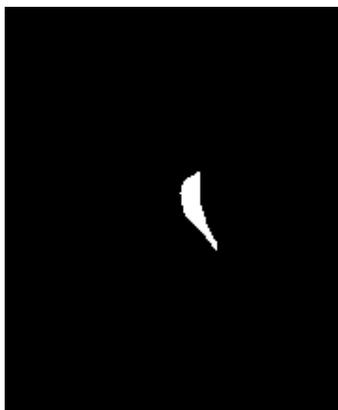


Figura 3.6 - Imagem segmentada com Neighborhood Connected para testar a seleção de pequenas estruturas presentes no artefato original

3.2.2.3 Algoritmo de *Watershed*

Como saída *Watershed* produz uma imagem com etiquetas que denotam um agrupamento de *pixel* formando uma região segmentada em particular.

A escolha correta dos parâmetros possui grande relevância, principalmente quanto ao tempo de processamento, assim como no tamanho e no número de regiões destacadas na imagem segmentada. Em inúmeras aplicações a seleção correta dos parâmetros é inferida por um processo de experimentação, tentativa e erro, até que a imagem resultante possua os corretos detalhes segmentados. (IBÁÑEZ, 2005)

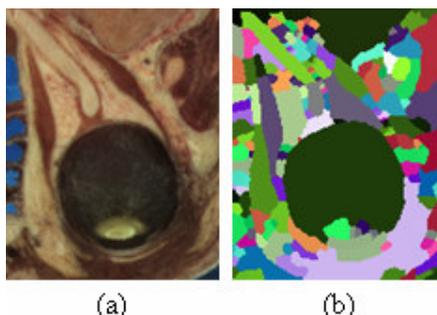


Figura 3.7 - Ensaio do algoritmo de watersheds

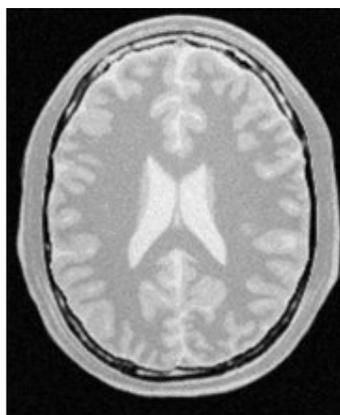
A imagem anterior é integrante do projeto *Visible Human*. É um olho direito feminino. A Figura 3.7 (a) demonstra a imagem original, logo (b) é o resultado da segmentação obtido seguindo as instruções em (IBÁÑEZ, 2005), objetivando a seleção do nervo ótico, assim como os músculos presentes na imagem.

A técnica aqui demonstrada pode ser amplamente aplicada para processos de segmentação automática, entretanto, é necessário que a imagem possua estruturas com aderência a esse método. Também é importante observar que o algoritmo apresentado no

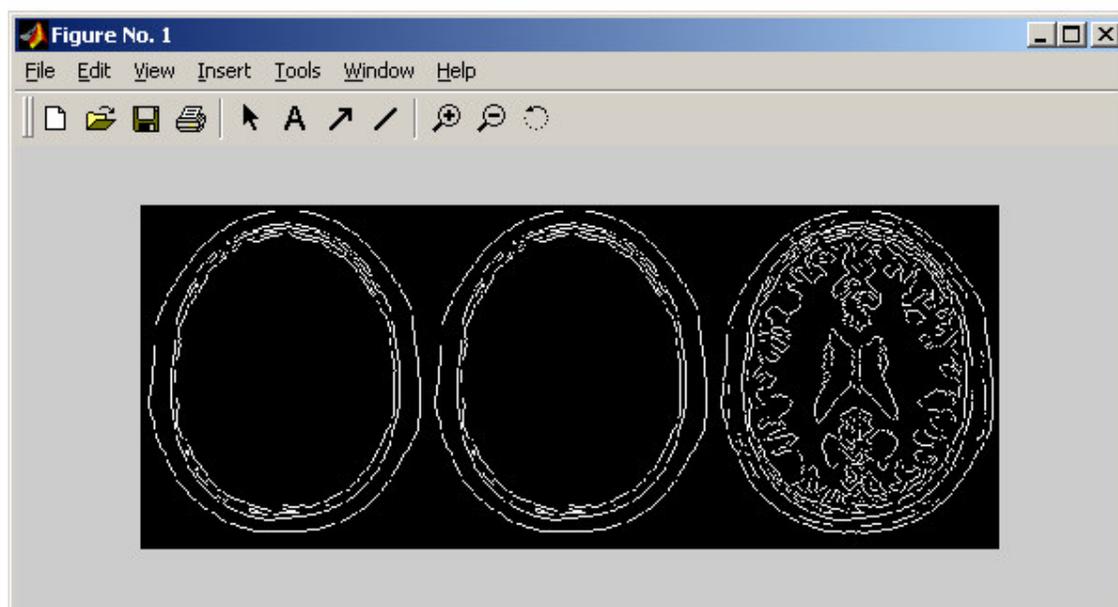
Anexo D possui um *pipeline* com filtros que manipulam, por exemplo, o gradiente da imagem a fim de proporcionar um melhor resultado.

3.2.2.4 Detecção de Bordas

A próxima figura é o resultado dos testes realizados para segmentação baseado nos modelos de detecção de bordas.



(a)



(b)

Figura 3.8 - Ensaio do modelo de detecção de bordas

Para realizar a extração das bordas utilizaram-se as funções originais dos MATLAB, conforme demonstrado a seguir:

```
I = imread('C:\EnsaioTC\bordas\original.png');
```

```
I1 = edge(I, 'sobel');  
I2 = edge(I, 'prewitt');  
I3 = edge(I, 'canny');  
imshow([I1, I2, I3]);
```

Os operadores selecionados para o teste são: *sobel*, *prewitt* e *canny* já disponíveis no MATLAB. Enquanto que a Figura 3.8 (a) é a imagem original; (b) é o produto de cada um dos operadores. Percebe-se que nesse caso de aplicação o operador de *canny* selecionou mais detalhes.

3.2.3 Outros Testes

Neste são apresentados ensaios executados a partir de imagens obtidas junto ao estudo da Correlação da Infecção por *Helicobacter pylori* com a ocorrência de processo aterosclerótico em camundongos C57BL/6 submetidos à dieta hiperlipidêmica. (ANDREOLLA F. e BONA, 2006; ANDREOLLA, 2006) A Figura 3.9 apresenta a imagem médica adquirida através da digitalização de uma lâmina histológica com o auxílio de uma câmera digital acoplada ao microscópio.

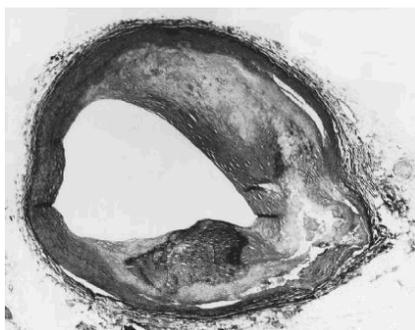


Figura 3.9 - Imagem original

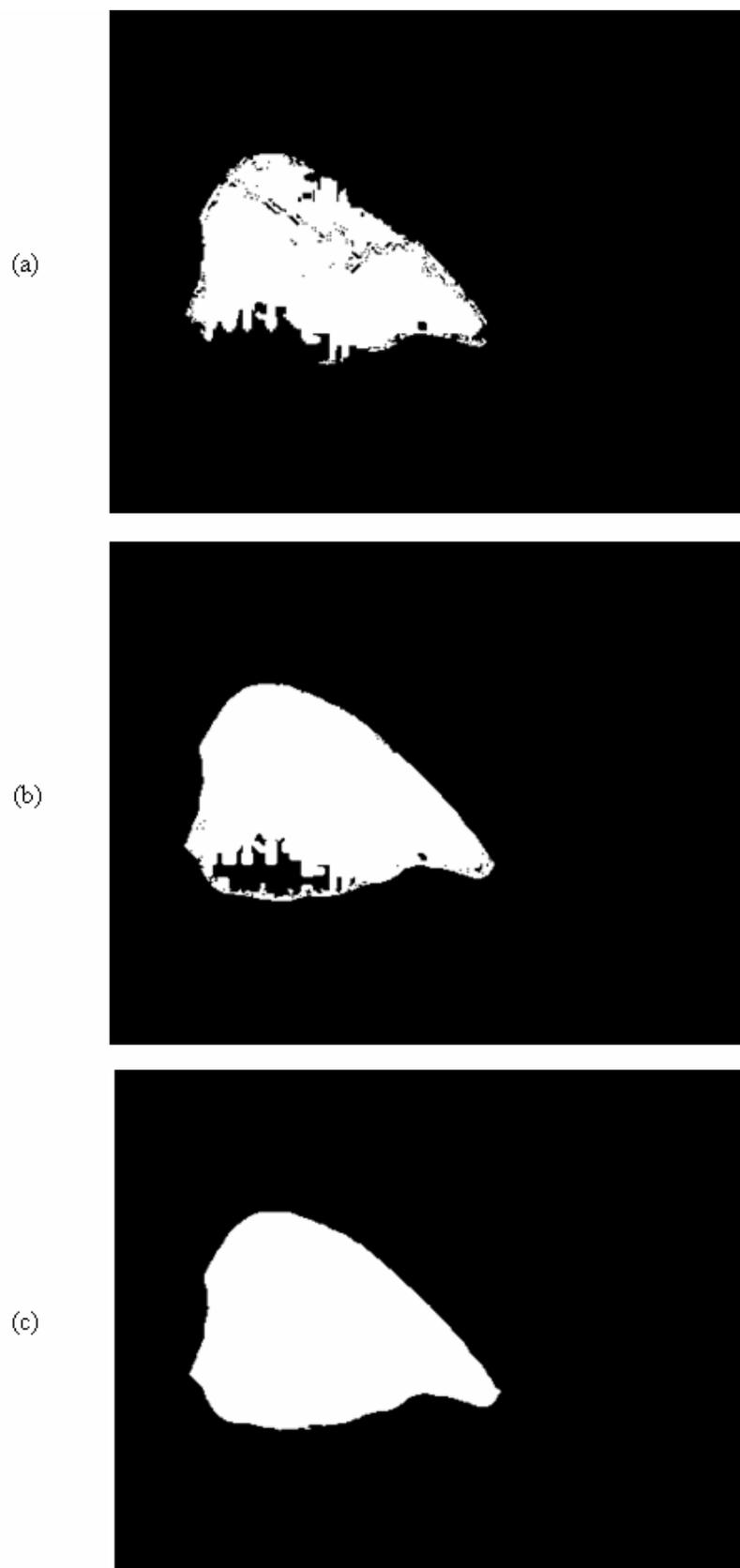


Figura 3.10 - Ensaio de segmentação de imagens

A figura anterior apresenta os resultados obtidos com a imagem original apresentada na Figura 3.9. Com o objetivo de obter os resultados mostrados em (a), (b) e (c) exibidos na Figura 3.10, como ponto semente inicial foram utilizados os pontos (124,182), variando o limiar mínimo e máximo. No caso da imagem (a) os limiares são: 221 e 227, enquanto que para (b) 210 e 230, e para (c) 200 e 250. (MOSSMANN, 2006) Esta apresentou uma boa segmentação em relação às outras, a estrutura alvo é o lúmen da artéria por onde o sangue flui. O algoritmo utilizado foi o *Connected Threshold* do Anexo B.

As próximas figuras demonstram os resultados de testes iniciais aplicando a técnica de contorno ativo, também conhecido como *snakes*, um modelo deformável paramétrico. O algoritmo utilizado, construído no MATLAB, foi obtido junto ao site da *Image Analysis and Communications Laboratory* (IACL) da *Johns Hopkins University* (XU, 2006). A perda de qualidade deve-se a conversão da imagem original para garantir a compatibilidade com o algoritmo utilizado.

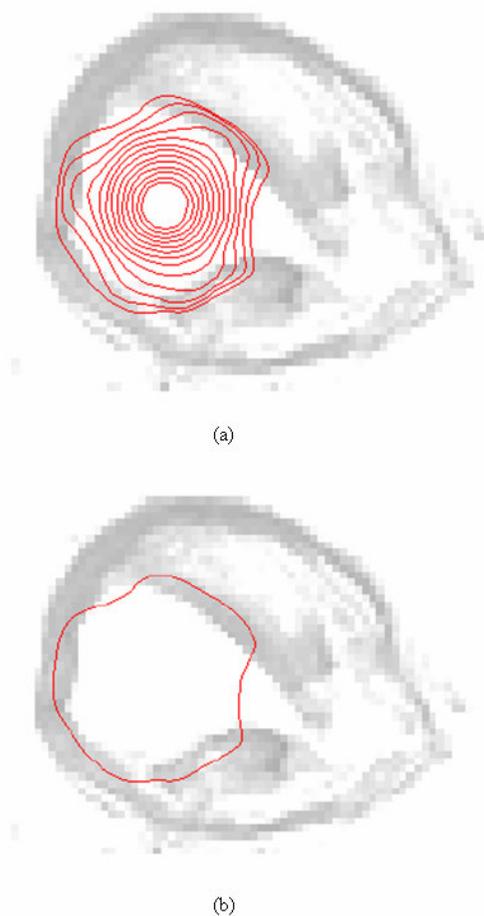


Figura 3.11 - Segmentação por contornos ativos; (a) Processo de segmentação; (b) Término da segmentação

Um dos algoritmos de modelos deformáveis disponíveis no ITK é baseado no artigo *Geodesic Active Contours* de (CASELLES, 1997) neste é apresentado um novo modelo de detecção de bordas. Na Figura 3.12 é apresentado um teste inicial realizado com esse método.



Figura 3.12 -Ensaio com a técnica de *Geodesic Active Contours*

Observa-se que através dos testes preliminares aqui apresentados o algoritmo de *Connected Threshold* obteve uma boa segmentação quando os limiares foram ajustados dentro de um intervalo de intensidade válido para a estrutura alvo. O método proposto em (CASELLES, 1997) resultou em uma segmentação efetiva do lúmen da artéria.

Esses últimos ensaios apresentados são focados em imagens cedidas por um estudo em curso no Curso de Biomedicina do Centro Universitário Feevale. O delineamento de um estudo biomédico que postula sobre derrames serosos, foco do sistema computacional desenvolvido junto a esta monografia é descrito no próximo capítulo.

4 ESTUDO BIOMÉDICO DA ANÁLISE MORFOMÉTRICA (ASSINATURA NUCLEAR) DE NÚCLEOS DE DERRAMES Serosos de etiologia benigna e maligna corados por May-Grünwald Giemsa - MGG

Este capítulo apresenta o estudo biomédico realizado no curso de Biomedicina do Centro Universitário Feevale utilizado como base para os requisitos do desenvolvimento de um Sistema de Apoio ao Diagnóstico - SAD. O software aqui proposto será aplicado neste estudo, para comprovação da técnica e posteriormente poderá ser aplicado a estudos similares.

O estudo biomédico intitulado “Análise Morfométrica (Assinatura Nuclear) de núcleos de derrames serosos de etiologia benigna e maligna corados por May-Grünwald Giemsa – MGG” é apresentado na seção 4.1 deste trabalho. Doravante neste trabalho será denominado apenas de estudo biomédico ou estudo de análise morfométrica.

4.1 O Estudo Biomédico

Os especialistas em patologia, na realização de exames citopatológicos, utilizam aspectos morfológicos visíveis para realizar os diagnósticos de lesões e assim verificar a malignidade da mesma (ROBBINS, 1996; MCKEE, 1997; TAO, 1996; BEDROSSIAN, 1991). As variações encontradas nos critérios morfológicos estabelecem dificuldades para o diagnóstico preciso a partir das células estudadas. Esta variação é mais evidente em células mesoteliais (MCKEE, 1997; BEDROSSIAN, 1991), onde alterações em células reativas podem mimetizar uma lesão maligna, uma vez que estas podem sobrepor aquelas que são encontradas em lesões malignas, gerando resultados falso-positivos e falso-negativos na etapa de diagnóstico. (MCKEE, 1997; TAO, 1996; BEDROSSIAN, 1991)

A dificuldade em definir a patologia de células fomentou inúmeras soluções, muitas das quais baseadas em estratégias multimodais, com o uso de microscopia eletrônica. Além disso, colorações especiais, imunocitoquímica, citometria de fluxo são operações biomédicas

utilizáveis (BEDROSSIAN, 1991). A disponibilidade destas técnicas auxiliares ainda é restrita a alguns centros de diagnóstico especializados e, mesmo com o uso destas, existem casos difíceis de diagnosticar.

Estudos na área verificam que é possível minimizar os problemas na etapa de diagnóstico com o uso de tecnologias, minimizando a variabilidade inter-observador presente principalmente em derrames pleurais (BODDINGTON, 1989; SILVA, 1997; SILVA, 1999).

A assinatura nuclear demonstra de forma numérica padrões e tendências, informações estas que podem auxiliar a etapa de diagnóstico e muitas vezes substituir produtos de reações citoquímicas. As informações numéricas obtidas nos núcleos permitem expressar uma série de dados que não podem ser percebidos pela visão do patologista. (SILVA e MONTIRONI, 1999; BARTELS, 1998)

O processamento de imagens médicas é aplicado com a finalidade de obter parâmetros para auxiliar a caracterização de benignidade e malignidade, isto podendo ser demonstrado em diferentes campos tais como câncer retal (BODDINGTON, 1989; ROSITO, 2003), câncer de esôfago (SILVA, 2001), adenocarcinoma pancreático (BERSCH, 2003), câncer de próstata (BARTELS, 1998) e adenocarcinoma de pleura (FRANTZ, 2004).

4.1.1 Objetivos

O objetivo deste projeto é caracterizar morfometricamente os núcleos celulares de derrames serosos benignos e malignos, definindo parâmetros de benignidade e malignidade, através da análise de imagem digital utilizando como base a coloração de May-Grünwald Giemsa (MGG). (BANKS, 1994)

4.1.2 Características Morfométricas

O sistema TICAS - *Taxonomic Intra-Cellular Analytic Systems (TICAS) for cell identification* (WIED, 1968) faz a análise de textura nuclear de mais de quinhentas características. Neste trabalho serão apresentadas algumas destas, com a numeração e nomenclatura original do TICAS, pois, em trabalhos futuros, pretende-se implementar todas que se fizerem necessárias ou forem consideradas importantes pelos patologistas que testarem o sistema.

Característica 1: Extinção total ou Densidade Ótica Total (DOT ou IOD)

É a soma dos valores de *pixels* em linhas e colunas na região nuclear.

$$IOD = \left(\frac{1}{16.48378975} \times \sum_{i=0} \sum_{j=0} I[i][j] \right) \quad (4.1)$$

Para todos os *pixels* dispostos na região nuclear. O denominador é um fator de correção do sistema TICAS.

Característica 2: Área do núcleo em micrômetros quadrados

É a área em *pixels* da região nuclear dividida por um fator de correção para o sistema óptico.

$$Área = \frac{\text{área}[\text{pixels}]}{65.935159[\text{pixels} / \mu^2]} \quad (4.2)$$

Característica 3: Circularidade nuclear (*roundness*)

É a medida da semelhança do formato do núcleo em relação a forma de um círculo. Neste caso, as estruturas perfeitamente circulares tem o valor da circularidade igual a 1 e quanto menos circulares, mais baixo o resultado.

$$Circularidade = \frac{p^2}{(4 \times \pi \times Área)} \quad (4.3)$$

Característica 6: RMS – Root mean squared

A média da raiz quadrada.

$$Ext\ 2 = \sum_{i=0}^{mx} \sum_{j=0}^{my} (I[i][j])^2 \quad Ext\ Sum = \sum_{i=0}^{mx} \sum_{j=0}^{my} (I[i][j])$$

$$RMS = \sqrt{\frac{Ext\ 2}{Ext\ Sum} - \frac{Ext\ Sum}{Área} \times \frac{Ext\ Sum}{Área}} \quad (4.4)$$

Característica 312: Heterogeneidade

Usado para calcular a heterogeneidade do núcleo.

$$HET = \frac{Nb + Nw}{Nt} \quad (4.5)$$

Onde

Nb é o número de *pixels* pretos

Nw é o número de *pixels* brancos

Nt é o total de *pixels* do núcleo

Característica 313: Homogeneidade

Usado para calcular a homogeneidade do núcleo, ou seja, a diferença entre 1 e a Heterogeneidade.

$$HOM = 1 - h1 \quad (4.6)$$

Característica 314: Granularidade ou *Clumpness*

Reflete a granularidade considerando o tamanho e a distribuição das áreas da imagem. Os oito *pixels* representam menos de um micrômetro da direção horizontal. Todos os cálculos são realizados na imagem negativa.

Para cada conjunto de 8 *pixels* em uma linha deve-se computar a diferença:

$$diff = |nb - nw| \quad (4.7)$$

Onde

nb é o número de *pixels* pretos no conjunto de 8 *pixels*

nw é o número de *pixels* brancos no conjunto de 8 *pixels*

$$cl = \sum_{i=1}^{tm} diff \quad (4.8)$$

Onde

tm é o número total de conjuntos encontrados no núcleo

$$cl = \frac{cl}{nb = nw} \quad (4.9)$$

4.2 Protótipo computacional aplicado ao estudo biomédico

Os sistemas de apoio ao diagnóstico – SAD – são softwares que auxiliam o profissional da área da saúde. A finalidade dos mesmos é processar os dados relevantes e disponibilizar as informações necessárias de forma clara e precisa, intuindo auxiliar na tomada de decisão durante a etapa de diagnóstico.

Contrariando os conceitos empregados durante as décadas de sessenta e setenta, onde os estudos apontavam para a utilização de ferramentas computacionais para gerar autodiagnósticos, hoje os SAD são utilizados como uma segunda opinião para melhorar o processo de reconhecimento, bem como reduzir os erros durante a etapa de diagnóstico. (MARQUES, 2001)

Pretende-se aplicar técnicas de processamento de imagens para o cenário anteriormente apresentado. Por sua vez, o sistema de apoio ao diagnóstico deverá segmentar e quantificar características com valor de diagnóstico para identificação do fenótipo maligno nas cromatinas presentes nas imagens obtidas durante a realização do estudo de análise morfométrica. Devido ao limitado cronograma para realização deste trabalho apenas algumas características serão desenvolvidas.

4.2.1 Objetivos

Segundo Ambrósio (AMBRÓSIO, 2002), no estudo “Redes neurais artificiais no apoio ao diagnóstico diferencial de lesões intersticiais pulmonares” uma das principais funções do SAD é diminuir os erros na etapa de diagnóstico. Carvalho e Silva (SILVA, 2005) em “Sistema de Análise de Nódulo Pulmonar” demonstram que a fadiga e o estado emocional do profissional responsável pela etapa de diagnóstico influenciam a correta qualificação e

quantificação da patologia, sendo essa uma das funcionalidades mais importantes dos sistemas de apoio ao diagnóstico.

O objetivo geral é criar um protótipo para auxiliar a prática laboratorial onde as imagens obtidas através de microscopia eletrônica devem ser submetidas a técnicas de processamento de imagens médicas. Prende-se que este software melhore a etapa do diagnóstico citopatológico.

Dentre os objetivos do sistema computacional estabelecem-se como específicos os seguintes: 1 - elaboração de uma fase responsável pelo pré-processamento dos artefatos digitais; 2 - aplicar técnicas de segmentação das estruturas necessárias nas imagens digitais; 3 – quantificação de características nucleares para ajudar na caracterização patológica das mesmas. Sendo esses os objetivos específicos do protótipo é objetivado pelo estudo também avaliar as estruturas segmentadas a partir de modelos de referência destacados por profissionais qualificados para o mesmo.

4.2.2 Metodologia

Para a realização do estudo proposto serão utilizadas imagens integrantes do projeto de pesquisa intitulado “Análise Morfométrica (Assinatura Nuclear) de núcleos de derrames serosos de etiologia benigna e maligna corados por May-Grünwald Giemsa – MGG”.

A realização deste estudo inicia com revisão bibliográfica sobre a área de processamento de imagens médicas, além disso, realização de ensaios das técnicas estudadas, conforme apresentado nos capítulos, 2 e 3 do trabalho. A seguir são delineados os requisitos funcionais para construção de um protótipo computacional que será empregado para estudar as imagens biomédicas.

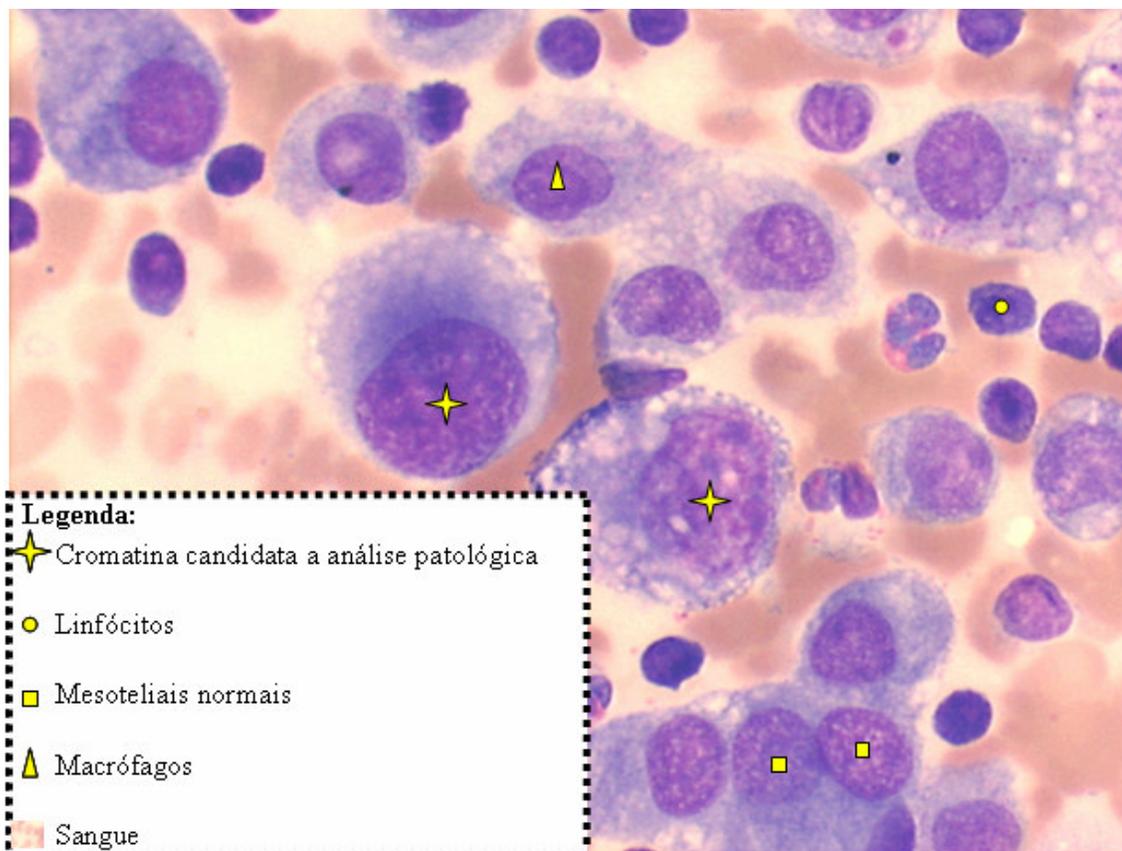


Figura 4.1 – Imagem citopatológica que apresenta as várias estruturas presentes na placa histológica

Conforme pode ser observado na Figura 4.1, um caso positivo de câncer, apresenta uma série de estruturas, tais como os linfócitos, macrófagos, entre outras. Nesta, conforme assinalado pelo citopatologista tem-se dois núcleos que podem ser utilizados para uma análise morfométrica e assim determinar através de parâmetros numéricos a malignade da patologia.

Com base na figura anterior pode-se aqui definir os dois primeiros requisitos:

1 - Permitir que o operador do software determine uma área qualquer dentro da imagem, esta área deve ser extraída da imagem original, formando assim uma segunda.

2 - A partir da saída proveniente do processo definido no primeiro requisito é necessária a localização e a extração da cromatina, ou seja, do núcleo da célula, através de um processo de segmentação, e até mesmo pré-processamento se houver necessidade.

3 - O último requisito funcional é a quantificação de características morfométricas, tendo como entrada o núcleo segmentado no processo delineado no segundo requisito. Vale ressaltar aqui que apesar das várias fórmulas morfométricas (vide capítulo 4.1.2) o protótipo

contará com três características neste momento, são elas: área, perímetro e circularidade, restrição necessária para o cumprimento do cronograma.

O requisito não funcional relevante é dado pela necessidade de verificar a qualidade das segmentações realizadas pelo software construído. Com essa finalidade é desenvolvido um sistema computacional com este propósito, este aplicará técnicas próprias estudadas e já aplicadas em estudos correlatos de processamento de imagens. Tal requisito é essencial para averiguar de forma quantitativa os núcleos extraídos, visto que o processo de segmentação neste caso é fundamental, uma vez que toda a análise morfométrica será realizada sobre o mesmo. Um objeto com segmentação errônea provocará a propagação do erro para as várias medidas métricas realizadas sobre o mesmo.

Os casos de testes são definidos por um citopatologista que definirá casos benignos e malignos para o estudo deste trabalho.

No próximo capítulo são apresentadas as técnicas aplicadas na tentativa de solucionar os requisitos funcionais e não funcionais definidos anteriormente, além de elucidar a composição da amostra de testes e apresentar os dados referentes a avaliação da segmentação.

5 SEGMENTAÇÃO DE IMAGENS E ANÁLISE MORFOMÉTRICA DE NÚCLEOS DE DERRAMES SEROSOS DE ETIOLOGIA BENIGNA E MALIGNA

Os valores de densidade óptica refletem, em sua distribuição espacial, uma estrutura de dependências que encontra suporte no estudo ultraestrutural da cromatina e matriz nuclear. Núcleos apresentam grânulos densos de cromatina e podem alterar áreas de elevada densidade óptica com outras onde a densidade é baixa. São os padrões de distribuição e a estrutura de dependência dos dados que refletem a distribuição de cromatina, também denominada textura e o estado funcional da célula. Núcleos de estados funcionais diversos são caracterizados por estruturas de dependência de densidade ópticas diferentes. (ROSITO, 2002)

5.1 Delineamento do estudo

O trabalho integrado da patologia com a informática abre novas possibilidades utilizando técnicas de processamento sobre imagens microscópicas. A partir da imagem digital é possível extrair informações celulares como o formato do núcleo, perímetro, cromatina entre outras medidas úteis para o diagnóstico médico. (WIED, 1968)

Atualmente o termo assinatura nuclear tem se disseminado, conceituando como o conjunto das características densimétricas baseadas na distribuição da cromatina do núcleo e descritas por padrões numéricos que unidas podem auxiliar em muito ao patologista.

As imagens relativas a este estudo foram adquiridas no laboratório de citopatologia do Hospital de Clínicas de Porto Alegre entre janeiro e setembro de 2004. Todas as imagens utilizaram como base a coloração de May-Grünwald Giemsa (MGG). Estas imagens possuem a resolução de 640 x 480 e estão no formato *Tagged Image File Format* (TIFF).

Para aquisição das imagens foi utilizado um microscópio da marca Olympus modelo BX 60 com uma câmera da Sony modelo DXC 970, utilizando um aumento de 50 vezes nas capturas.

5.2 População estudada

O total de imagens obtidas foi de 560 núcleos provenientes de lâminas de citopatologia de células malignas de adenocarcinoma e 360 núcleos de células mesoteliais normais e reativas de cavidades serosas de etiologia benigna.

Neste estudo foram incluídas imagens de 22 pacientes, sendo 12 casos positivos de câncer e 10 casos negativos. Para os positivos, em 10 casos foram adquiridos das imagens 3 núcleos para serem estudados e em 2 casos foram adquiridos 4 núcleos celulares. Entre os casos negativos, para os 10 casos, foram utilizados 3 núcleos de cada. A Tabela 5.1 a seguir demonstra a amostra estudada:

Tabela 5.1 - Amostra estudada

Negativos		Positivos	
Paciente	Quantidade Imagens	Paciente	Quantidade Imagens
91120	3	91416	3
91150	3	91421	3
277883	3	276691	3
277963	3	276942	3
278046	3	277206	3
278331	3	277395	3
278427	3	277448	3
278444	3	277778	3
278501	3	277867	3
278586	3	278094	3
--	--	278284	4
--	--	291418	4

Para cada núcleo foram calculadas três características descritivas de distribuição espacial, quais sejam: área, perímetro e circularidade.

5.3 Processo

O processo para a segmentação das imagens é mostrado no fluxo apresentado na Figura 5.1 a seguir e cada etapa descrita e explicada.

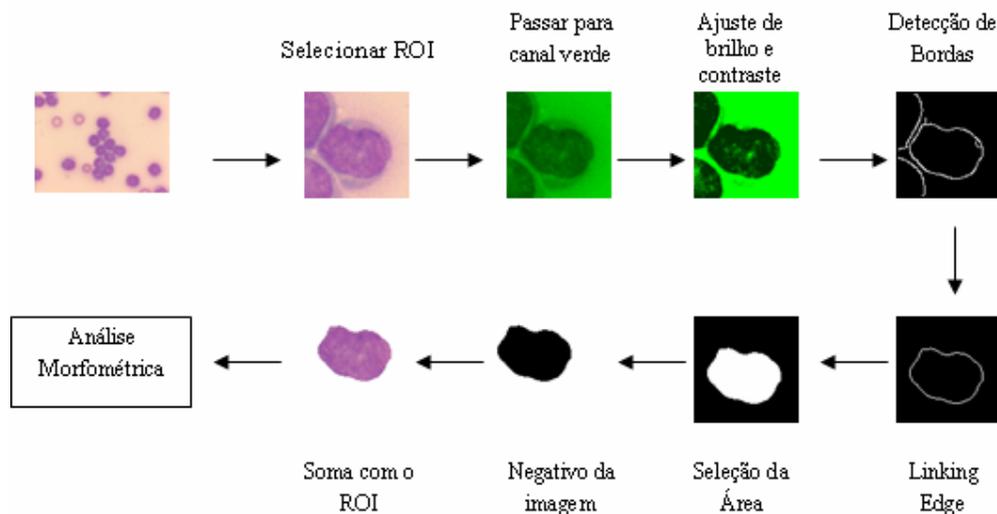


Figura 5.1 - Fluxo do sistema de análise de imagens

5.3.1 Selecionar ROI

Foi desenvolvida uma interface que apresenta a imagem e solicita a seleção do núcleo a ser segmentado. Marcando dois pontos, um no canto superior direito e outro no inferior esquerdo do núcleo, forma-se um quadrado sobre o mesmo e a imagem do núcleo é extraída e salva com um rótulo. O processo é mostrado na figura a seguir:

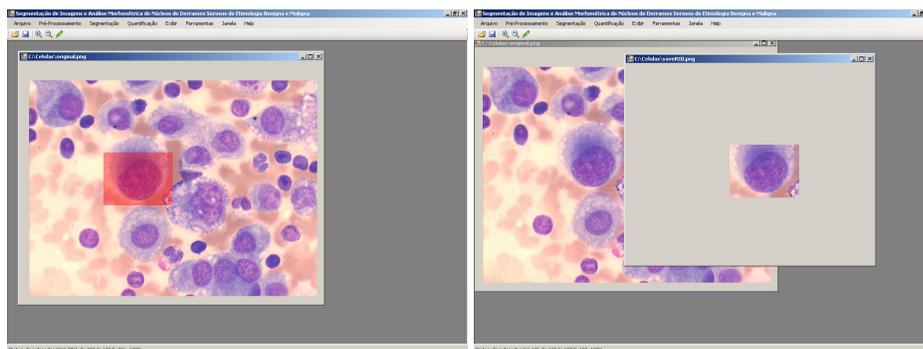


Figura 5.2 - Seleção da área de interesse

5.3.2 Passar para o canal verde

O primeiro teste realizado na figura original teve o objetivo de alisar preservando os limites das estruturas presentes na imagem e assim, após sua aplicação, diminuir a incerteza no computo de um algoritmo com a finalidade de destacar as bordas ou regiões de interesse. Os testes realizados são demonstrados na Figura 5.4 e a Figura 5.5 é resultado da aplicação dos *Smoothing Filters* presentes no ITK. Enquanto que a Figura 5.4 demonstra a aplicação do

Curvature Anisotropic Diffusion com 25 e 50 iterações, a Figura 5.5 é o *Gradient Anisotropic Diffusion* onde varia o número de iterações entre 25, 50 e 70.

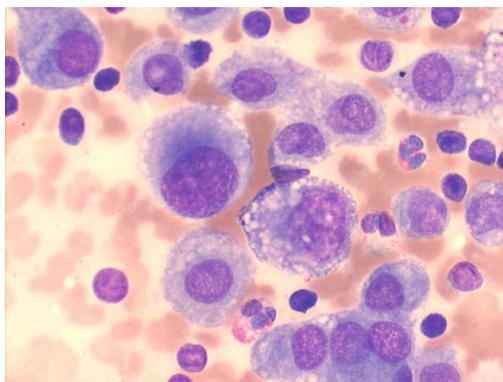


Figura 5.3 - Imagem 277448-P-01 utilizada como base para testes

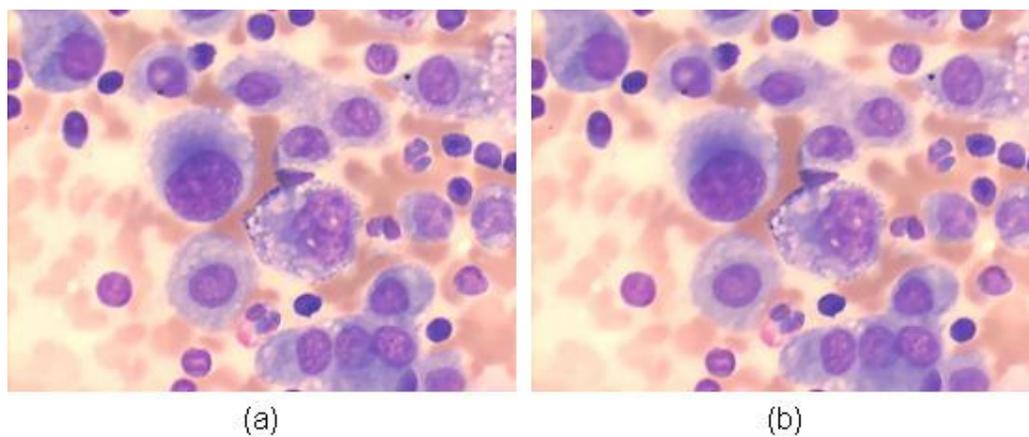


Figura 5.4 - Imagem resultante da operação *Curvature Anisotropic Diffusion*; (a) Operação com 25 iterações; (b) Operação com 50 iterações

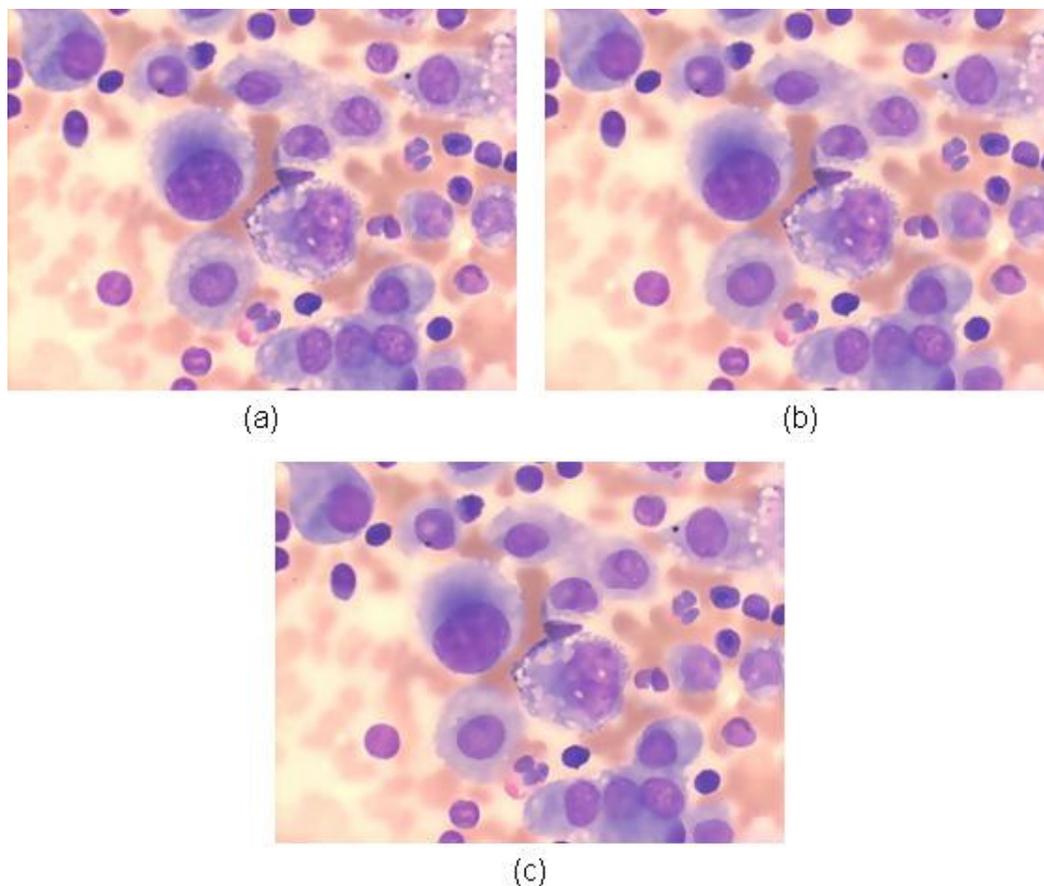


Figura 5.5 - Imagem resultante da operação *Gradient Anisotropic Diffusion*; (a) Operação com 25 iterações; (b) Operação com 50 iterações; (c) Operação com 70 iterações

Conforme pode ser observado, a imagem original apresenta um baixo contraste entre as regiões de interesse, núcleo e o citoplasma, dificultado assim o sucesso de um algoritmo que intuíva a segmentação automática das áreas de interesse.

Na tentativa de verificar um melhor contraste entre a área de interesse e o fundo da imagem foi construído um algoritmo para extrair as camadas RGB da imagem original, tendo os seguintes resultados apresentados na Figura 5.6.

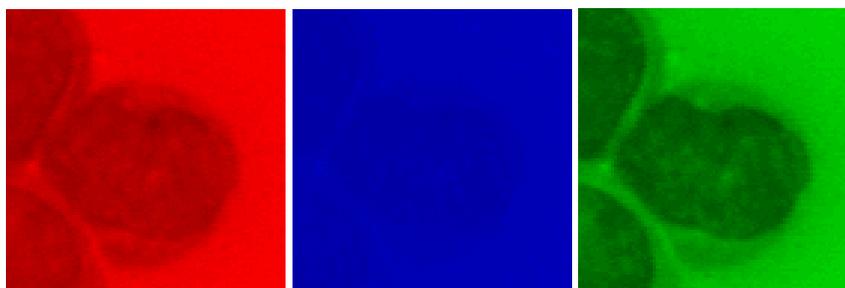


Figura 5.6 - Canal vermelho; Canal azul; Canal verde

O maior contraste entre a região de interesse (ROI) e o fundo da imagem é visualizado no componente verde da imagem, enquanto que o componente azul não apresenta quase nenhum contraste visível e o vermelho é de pouca qualidade.

O algoritmo utilizado para a separação do canal RGB (GONZALES, 2000, p. 157) e o canal isolado é obtido com a atribuição do valor zero na intensidade dos componentes espectrais não necessários. Assim para uma imagem formada apenas com componente verde, este permanece inalterado, enquanto que nos outros planos, azul e vermelho, o valor da intensidade de todos os *pixels* são alterados para zero. Adota-se neste trabalho o padrão RGB (GONZALES, 2000, p. 160) para manipulação das imagens coloridas.

Após verificar que o resultado do canal verde possui melhor contraste foi desenvolvido um algoritmo para manipular de forma local o contraste deste canal, sendo este detalhado a seguir.

5.3.3 Ajuste de brilho e contraste

A imagem na tonalidade verde pura ainda não era suficientemente nítida para a extração das bordas, motivo pelo qual foi necessário o ajuste do brilho e contraste das mesmas.

Para separar a região de interesse do fundo da imagem e de outros objetos que não o núcleo das células, aplicou-se um contraste linear, modificando o brilho e contraste.

Existem várias técnicas para modificar tanto o brilho quanto o contraste da imagem, podendo ser citado como um dos principais as alterações aplicadas ao histograma da imagem (Vide capítulo 2.2 deste trabalho).

Para a melhora do contraste da imagem, para cada *pixel* foi aplicada a seguinte fórmula:

$$g(x, y) = \left(\left(\left(\left(\frac{f(x, y)}{255} \right) - 0.5 \right) \times 1.33 \right) + .05 \right) \times 255$$

Para

$$se \ g(x, y) < 0 \quad g(x, y) = 0$$

$$se \ g(x, y) > 255 \quad g(x, y) = 255$$

(5.1)

Onde

$g(x, y)$: é uma nova imagem;

$f(x, y)$: imagem original;

0.5: parâmetro de normalização;

1.33: é um valor constante obtido através de experimentos na melhoria do contraste.

Quanto ao brilho, este obtém-se adicionando um valor constante predefinido ao *pixel* da imagem, conforme segue.

$$g(x, y) = f(x, y) + 100$$

Para

$$\text{se } g(x, y) < 0 \quad g(x, y) = 0$$

$$\text{se } g(x, y) > 255 \quad g(x, y) = 255$$

(5.2)

Na seqüência é apresentada a imagem, Figura 5.7, no canal verde, seguida da imagem com o ajuste do brilho e contraste, verifica-se que após a aplicação das equações, definidas anteriormente, o núcleo na imagem resultante possui melhor contraste com o fundo da imagem.

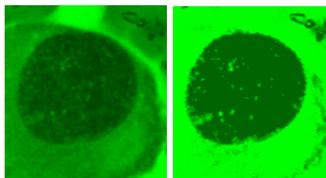


Figura 5.7 - A esquerda a imagem no canal verde e a direita com o ajuste do brilho e contraste

5.3.4 Detecção de bordas

Com a imagem ajustada em seu brilho e contraste, foi possível a detecção das bordas da mesma.

Na detecção foi utilizado o algoritmo desenvolvido por Meer e Georgescu (2001) no *Robust Image Understanding Laboratory* (RIUL) da *Rutgers University*. Este consiste em uma variação do detector de bordas de Canny (CANNY, 1986). Em (MEER e GEORGESCU, 2001) o destaque das bordas é realizado através da correlação entre a borda determinada e um molde ideal da borda que refere-se ao mesmo sentido do gradiente. Isso é possível com um cálculo da média da “confiança”, entre a porcentagem dos *pixels* que têm um valor do

gradiente próximos daqueles que a borda considera. Com esta técnica (MEER e GEORGESCU, 2001) não negligencia detalhes importantes para determinar os limites das estruturas alvo.

Na grande maioria das imagens a detecção de bordas gera um contorno exato do núcleo, porém, em alguns casos, quando o citoplasma possui uma coloração muito próxima do núcleo, ou quando as imagens não possuem uma boa definição, a detecção de bordas apresenta pequenas falhas, que tiveram que ser corrigidas ainda manualmente. A seguir são apresentados alguns casos de sucesso na detecção de bordas e alguns casos em que foram necessários ajustes.

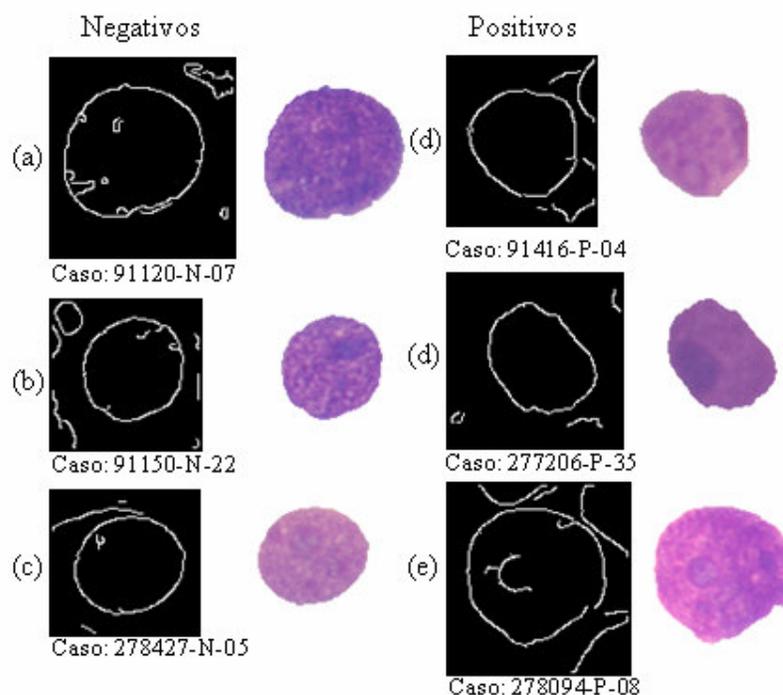


Figura 5.8 - Sucesso na detecção de bordas, primeiros três para os Casos Negativos de câncer e as três últimas para Casos Positivos

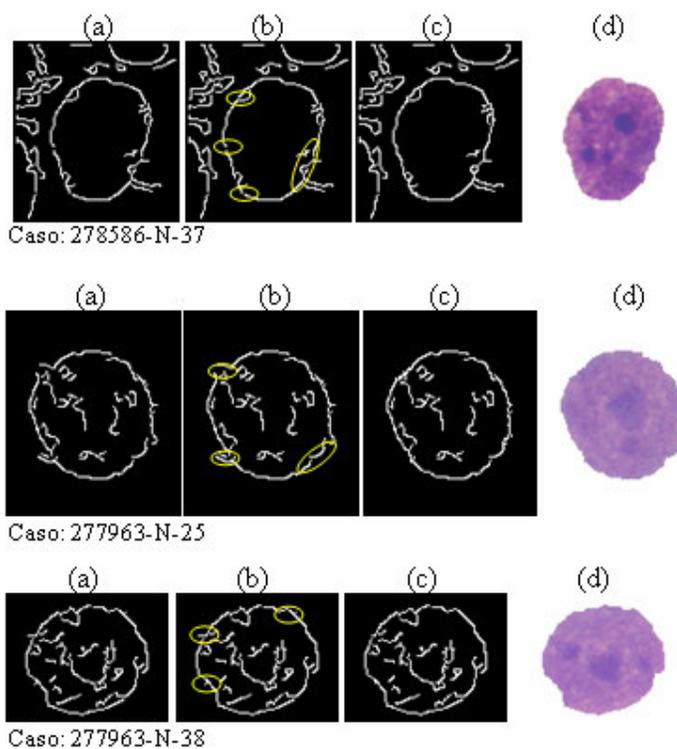


Figura 5.9 - Casos negativos com ajustes manuais; (a) Detecção da borda; (b) Local dos ajustes manuais; (c) Borda ajustada; (d) Segmentação

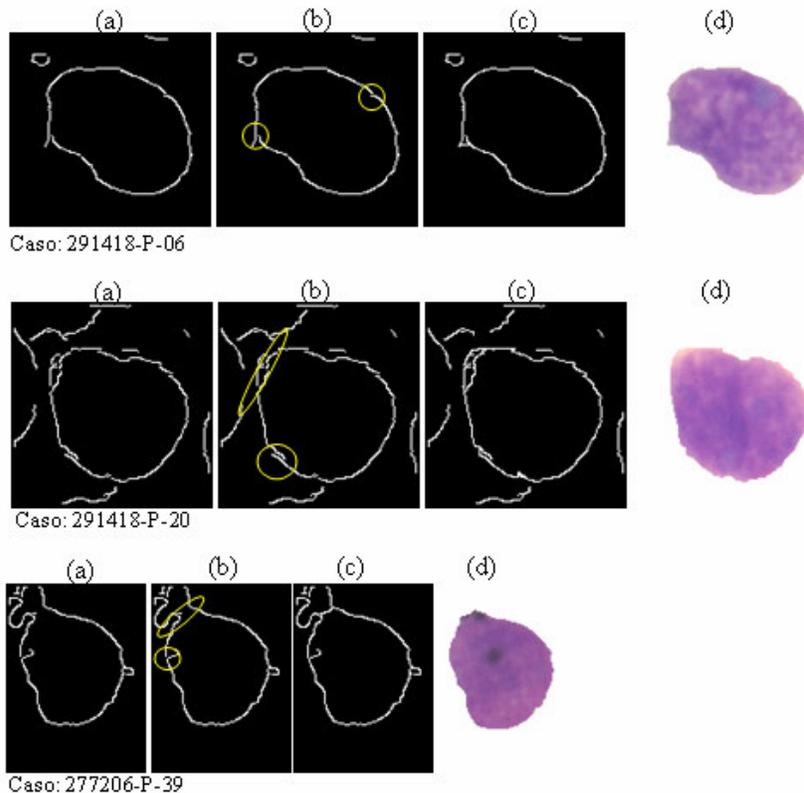


Figura 5.10 - Casos positivos com ajustes manuais; (a) Detecção da borda; (b) Local dos ajustes manuais; (c) Borda ajustada; (d) Segmentação

5.3.5 Linking Edge

O propósito da etapa de ligação de bordas (GONZALES, 2000, p. 306) é unir as descontinuidades apresentadas na etapa da detecção de bordas. As descontinuidades normalmente são formadas pela presença de ruído e ou iluminação não uniforme. Conforme pode ser visto, nos casos de teste deste estudo, aqui a descontinuidade é formada normalmente nos casos onde o núcleo não possui um contraste significativo com o fundo da imagem, no caso, o citoplasma.

Na tentativa de minimizar o efeito de fragmento de borda, aplicou-se o algoritmo desenvolvido na *The University of Western Australia* por (KOVESI, 2007) inspirado no trabalho realizado no *The Laboratory for Computational Intelligence* da *University of British Columbia - UBC.ca* (LCI, 2007) que é apresentado nos anexos E, F e G.

Um dos testes realizado no decorrer deste trabalho é apresentado a seguir, onde a primeira Figura 5.11 (a) representa uma imagem de teste com pequenas descontinuidades nas bordas. Pode-se observar melhor em (b) onde as falhas nas bordas são identificadas pelos índices, 1; 2 e 3. A Figura 5.11 (c), resultante do computo do algoritmo anterior, apresentada a união das descontinuidades 1 e 2, contudo o índice 3 presente em uma estrutura curva não foi unida. O afinamento das linhas é resultado da aplicação dos operadores de morfologia matemática.

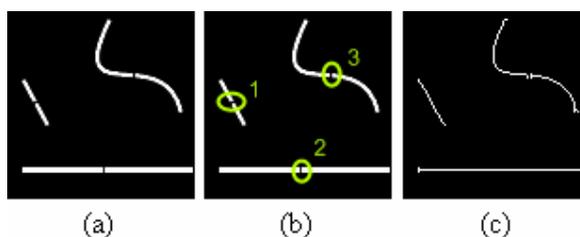


Figura 5.11 - Testes de união de bordas

Por inúmeras vezes o problema ilustrado na Figura 5.11 (b), como índice 3 é observado no processo de segmentação dos núcleos celulares, estes ainda no presente momento foram resolvidos com a interação manual do patologista. Pretende-se melhorar este cenário com uma abordagem global (GONZALES, 2000, p. 308), em oposição a local (GONZALES, 2000, p. 306) aplicada na vizinhança como aqui demonstrado. Outra possível técnica para solucionar o problema é o processamento global através de técnicas baseados em grafos, onde se localiza caminhos baratos que correspondam às bordas significativas.

5.3.6 Seleção da área

A partir da seleção das bordas onde são delimitados os limites do objeto principal, neste caso o núcleo selecionado para estudo, é necessário criar uma etiqueta que indique os *pixels* que fazem parte da área delimitada pelo processo de detecção.

Os segmentos parciais de bordas no interior do núcleo são desconsiderados nesta etapa. Uma única etiqueta é criada e nestes casos os segmentos são incorporados.

Os anexos H e I apresentam os algoritmos utilizados para etiquetar os *pixels* de uma determinada região.

5.3.7 Negativo da Imagem

O negativo (NOT) de uma imagem é um operador lógico. As operações lógicas são AND, OR, Ex-OR (OR exclusivo) e NOT. A aplicação de alguns destes operadores (AND, OR e Ex-OR) requer duas imagens de entrada e resultam em apenas uma imagem de saída. Já o operador NOT é realizado apenas sobre uma única imagem, é uma simples inversão do valor do *pixel*: onde estiver ligado passa para desligado e vice e versa. (RUSS, 1998)

Para realizar a operação NOT descrita anteriormente, em cada uma dos canais (RGB) executou-se a operação de $g(x, y) = 255 - f(x, y)$, sendo (x, y) a posição local de cada *pixel* na imagem original.

5.3.8 Soma com o ROI

A soma de imagem pode ser utilizada, em vários casos, como a média das imagens, onde somam-se localmente os *pixels* e divide-se pela quantidade de imagens, esta técnica é utilizada, principalmente, na supressão de ruído.

Aplicado a este problema específico, soma-se o negativo da imagem computada pelo algoritmo de segmentação ao ROI selecionado. O resultado deste cálculo é a segmentação completa do núcleo e a formulação matemática pode ser dada por: $g(x, y) = \text{areaNegativo}(x, y) + \text{ROI}(x, y)$, onde se $g(x, y) > 255$ então $g(x, y) = 255$ ou $g(x, y) < 0$ logo $g(x, y) = 0$. Na Figura 5.19 pode-se verificar o processo anteriormente descrito onde (a) é o ROI (g) é a negativo da área e (h) é o resultado do computa da soma entre (a) mais (g).

5.4 Análise morfométrica

A assinatura digital de um núcleo é um conjunto de características densimétricas baseadas na distribuição da cromatina nuclear. Na morfometria de área, superfícies podem ser aplicadas em inúmeros casos de análise histológica e auxiliar nos prognósticos. (ROSITO, 2002).

Conforme apresentado na página 58, capítulo 4.2.2 este trabalho preocupou-se em desenvolver primeiramente três características morfométricas utilizadas no estudo biomédico, estas são descritas a seguir.

As características morfométrica apresentadas nesta seção são todas computadas sobre o artefato gerado após a etapa da seleção da área, um exemplo desta pode ser verificado na Figura 5.19 (f).

5.4.1 Cálculo da Área

Defini-se área como o número de pixels contidos dentro de uma determinada fronteira (GONZALES, 2000, p. 359).

Como este cálculo é realizado sobre a imagem retornada do processo de seleção de área a resposta é dada pela contagem da quantidade de *pixels* brancos da imagem, uma vez que na imagem binária, base para o cálculo, os *pixels* etiquetados como pretos, representam o fundo da imagem, já os brancos são o núcleo da célula estrutura alvo para cálculo.

5.4.2 Cálculo do Perímetro

O perímetro é tamanho da fronteira do objeto (GONZALES, 2000, p. 359). O perímetro é obtido no cálculo da distância entre cada par adjacente de *pixels* que se localiza na fronteira, borda, do objeto.

5.4.3 Cálculo da Circularidade

A circularidade de um objeto é encontrada tendo como base as duas outras medidas anteriormente delineadas, através da seguinte formulação matemática:

$$c = \frac{P^2}{4 \times \pi \times A} \quad (5.3)$$

Onde:

c : é a circularidade calculada

P : é o perímetro do objeto

A : é a área do objeto

Detalhes sobre a circularidade vide capítulo 4.1.2.

5.5 Protótipo Desenvolvido

O ambiente de desenvolvimento foi composto pela integração de algoritmos desenvolvidos para MATLAB com classes e camada de apresentação desenvolvida em C#, linguagem da Microsoft. A Mathworks, empresa responsável pelo desenvolvimento do MATLAB, criou uma extensão do compilador do MATLAB, chamado de "MATLAB Builder for .NET 2.2" (MATLAB BUILDER FOR .NET, 2007). Este gera .NET Assemblies de forma automática e pode ser chamado de qualquer *Common Language Specification* (CLS) - *compliant language* (CLS, 2007), tais como C#, C++ ou VB .Net. Além deste, ele permite a geração de componentes COM - *Component Object Model Technologies* (COM, 2007), a tecnologia da Microsoft que fornece propriedades para a criação de componentes de software reusáveis em sistemas que rodam sobre o sistema operacional Windows.

As principais funcionalidades do MATLAB Builder for .NET 2.2 são:

- Conversão de algoritmos originalmente escritos em MATLAB para componentes com suporte a .NET ou de COM.
- Suporte a conversão de tipo de dados entre as várias tecnologias, ou seja, entre os tipos de dados nativos de .NET e COM e o MATLAB.
- Os .NET Assemblies gerados podem ser distribuídos e não requerem o MATLAB instalado localmente para funcionar.

Essas tecnologias, principalmente o C#, foram selecionadas tendo em vista o conhecimento prévio do autor projeto, minimizando assim a curva de aprendizagem.

A tela do protótipo desenvolvido é apresentada pela Figura 5.12. Nesta são explicados os principais componentes da camada de apresentação do sistema, os menus de acesso, barra de ferramentas e status.

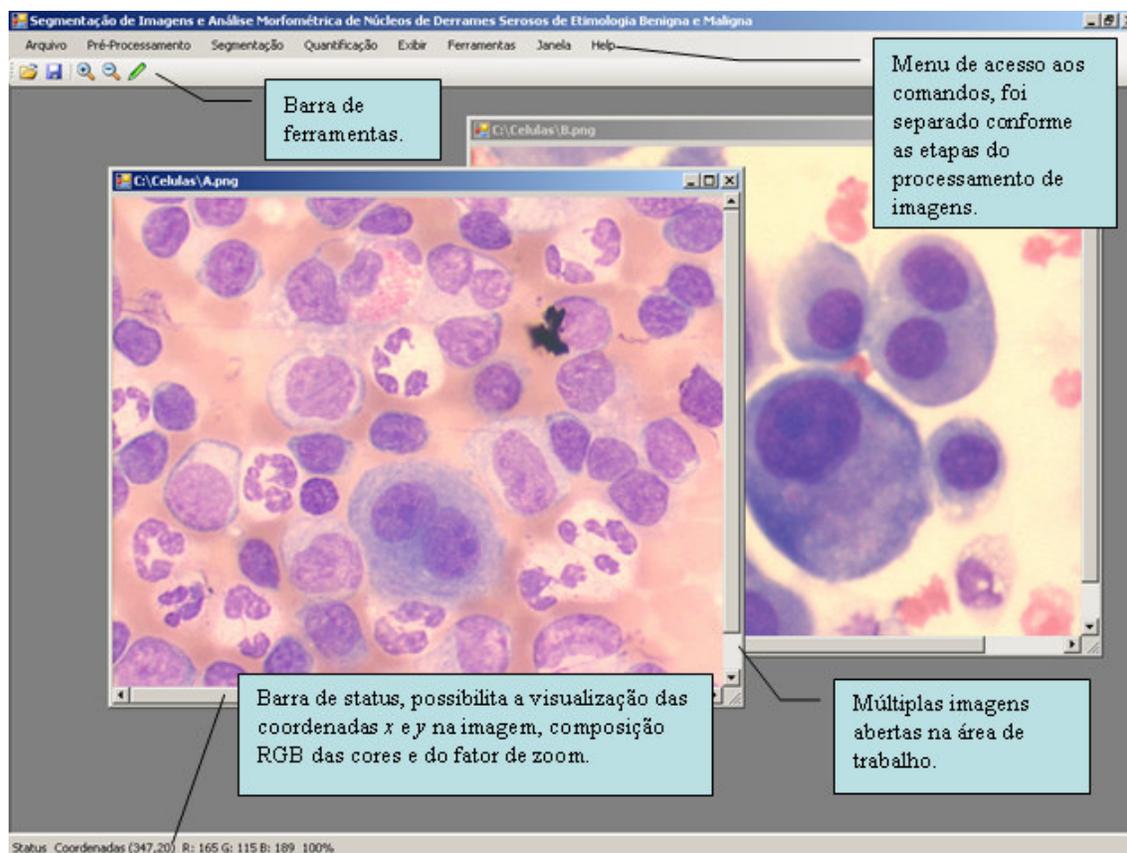


Figura 5.12 - Tela do protótipo computacional desenvolvido

A seguir são demonstrados os acessos aos principais comandos do sistema, que são realizados pelos menus: Pré-Processamento, Segmentação e Quantificação.

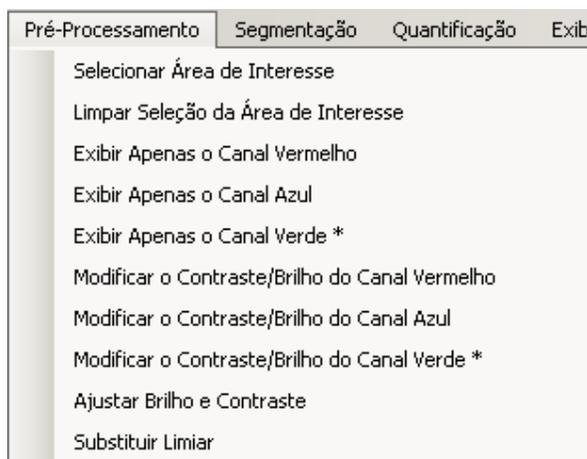


Figura 5.13 - Menu Pré-Processamento

O menu pré-processamento, conta com as funcionalidades listadas na Figura 5.13. São elas:

Selecionar Área de Interesse: Vide capítulo 5.3.1 na página 63.

Limpar Seleção de Interesse: Anula o comando executado pela opção anterior.

As opções **Exibir Apenas o Canal Vermelho, Azul e Verde** na imagem selecionada mostra apenas um dos canais RGB, conforme a opção selecionada. Para mais detalhes do canal verde vide capítulo 5.3.2 na página 63.

Modificar o Contraste/Brilho do canal Vermelho, Azul e Verde: modifica os valores de intensidade do *pixels*, conforme o canal selecionado. Detalhes podem ser vistos na página 66 no capítulo 5.3.3.

Ajustar Brilho e Contraste: permite que o usuário do sistema selecione os parâmetros para alterar as intensidades dos *pixels*.

Substituir Limiar: caso exista a necessidade, essa funcionalidade permite que um dado limiar seja substituído por outro informado.

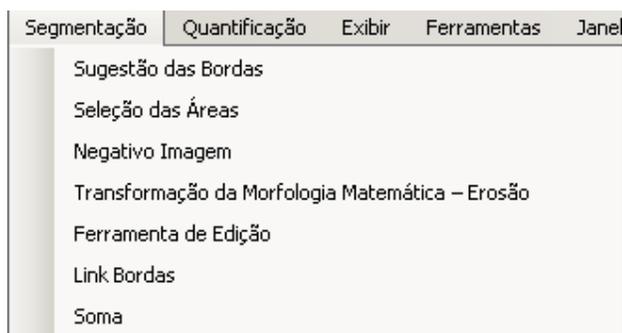


Figura 5.14 - Menu Segmentação

Na Figura 5.14 foram apresentadas as funcionalidade que pertencem a etapa de segmentação são estas:

Sugestão das Bordas: conforme apresentado no capítulo 5.3.4, página 67.

Seleção das Áreas: o funcionamento é delineado na página 70, capítulo 5.3.6.

Negativo Imagem: No capítulo 5.3.7, página 71 apresenta esta funcionalidade.

Transformação da Morfologia Matemática - Erosão: Implementa a possibilidade de erosão nas imagens.

Ferramenta de Edição: habilita a funcionalidade que permite a edição da imagem, utilizado para concertar imperfeições geradas na detecção das bordas.

Link Bordas: Vide página 70, capítulo 5.3.5.

Soma: Apresentado na página 71 capítulo 5.3.8.

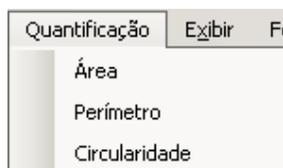


Figura 5.15 - Menu de Quantificação

A Figura 5.15 apresenta o menu que viabiliza o acesso as funcionalidades de quantificação construídas. Os detalhes das características morfométricas, área, perímetro e circularidade, são delineadas nos capítulos 5.4.1, 5.4.2 e 5.4.3.

5.6 Avaliação da segmentação

Existem muitos modelos para avaliação quantitativa de uma determina segmentação, dentre os vários modelos pode-se citar o índice de mérito de Pratt, índice de similaridade e o RUMA, estes são delineados nas seções a seguir.

5.6.1 Índice de Pratt

O Índice de Pratt proposto em (ABDOU, 1979) e utilizado em (SILVA, 2005) para avaliação de segmentação de contornos pulmonares em tomográficas computadorizadas, pode ser definido matematicamente por:

$$F_{pratt} = \frac{1}{N} \sum_{i=1}^N \frac{1}{1 + \alpha \times d_i^2} \quad (5.4)$$

Onde

N : é o número de distâncias usadas;

d_i : distância entre dois pontos correspondentes;

α : parâmetro associado ao tamanho do contorno;

O índice de Pratt define de forma global o comportamento das distâncias dos contornos.

5.6.2 Índice de Similaridade

O índice de similaridade mede a semelhança entre duas imagens binárias, o resultado pode variar entre 0° onde os contornos são idênticos e 90° para contornos diferentes. O índice de similaridade é calculado com a fórmula: (SILVA, 2005)

$$\theta = \arccos \left(\frac{\vec{H} \cdot \vec{I}}{\|\vec{H}\| \|\vec{I}\|} \right) \quad (5.5)$$

São vetores, \vec{H} e \vec{I} com os contornos e de tamanho $n \text{ pixels}$.

5.6.3 RUMA

O RUMA - *Relative Ultimate Measurement Accuracy*, medida de exatidão final relativa é um método de avaliação com critérios objetivos e quantitativos sua formulação matemática e definida por: (OLIVEIRA, 2003; ZHANG, 2004)

$$RUMA_f = \left(\frac{|R_f - S_f|}{R_f} \right) \times 100 \quad (5.6)$$

R_f : Valor de f na imagem de referência;

S_f : Valor de f na imagem segmentada;

f : Parâmetro de discrepância;

Os resultados gerados pelo cálculo do $RUMA_f$ próximos a zero indicam uma segmentação de qualidade, logo os resultados são inversamente proporcionais à qualidade da segmentação. (OLIVEIRA, 2003)

Inúmeros parâmetros mensuráveis podem ser utilizados para avaliação e comparação de algoritmos de segmentação. Normalmente a seleção dos parâmetros a ser avaliados está diretamente relacionada à necessidade e objetivos pretendidos. Uma das formas mais comuns de utilizar RUMA é através da comparação de fatores geométricos da imagem, tais como área e perímetro. (OLIVEIRA, 2003)

Os artefatos de referência são imagens binárias constituídas a partir das segmentadas manualmente pelo patologista, onde os *pixels* identificados pela cor branca representam o espaço ocupado pelo núcleo que será comparado com o resultado da segmentação realizada pelo protótipo construído neste estudo.

Para auxiliar no processo de avaliação foi construído, durante este trabalho, um software com este propósito. Este poderá ser expandido contemplando diversas técnicas de

avaliação de segmentação, além do RUMA, e assim ser utilizado em futuros estudos com temas similares.

Pretendendo verificar a exatidão do sistema de avaliação foram criadas duas imagens binárias para executar os testes onde a estrutura que será analisada é identificada pela cor branca.

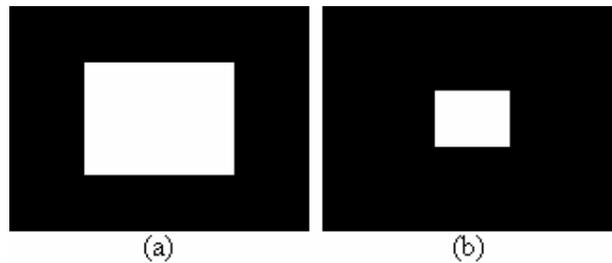


Figura 5.16 - Imagens de teste para o sistema de avaliação

A Figura 5.16 (a) que doravante será denominada de I_a possui um tamanho total de 160×120 e o retângulo central branco uma dimensão de 80×60 . Já a Figura 5.16(b), I_b é formada pelo mesmo tamanho total, e o objeto no centro tem uma dimensão de 40×30 . Os retângulos etiquetados com a cor branca representam os artefatos segmentados em ambas as imagens.

Dois testes foram realizados, onde no primeiro é computando o RUMA através da área, considerando-se a imagem I_a como referência, assim como segmentação, onde o resultado deverá ser zero, pois:

Área da segmentação em I_a : 4800 *pixels* quadrados;

Logo:

$$RUMA_{area} = \left(\frac{|4800 - 4800|}{4800} \right) \times 100 \quad (5.7)$$

$$RUMA_{area} = 0\%$$

O segundo teste é computado onde a imagem dada como referência é I_a e a segmentada I_b , então se têm que;

Área da segmentação de referência em I_a : 4800 *pixels* quadrados;

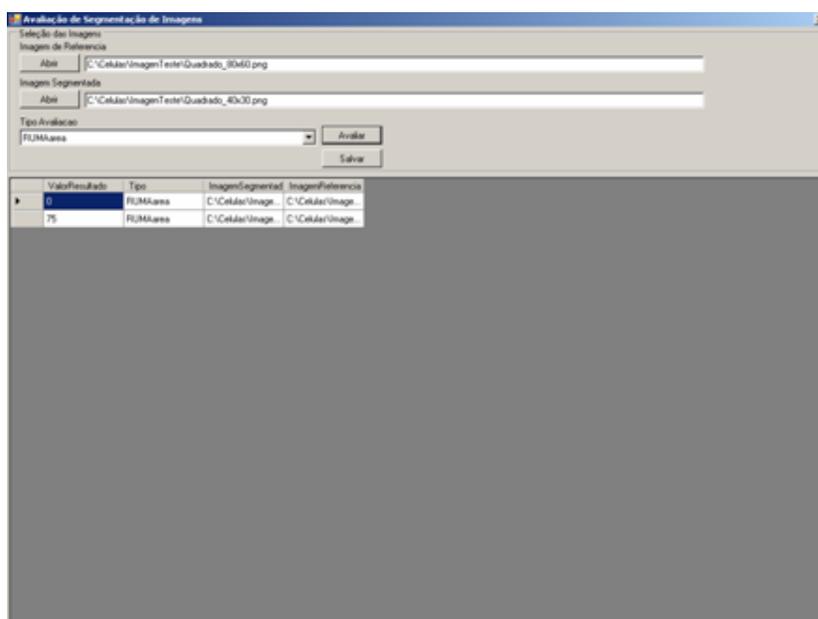
Área da segmentação em I_b : 1200 *pixels* quadrados;

Logo:

$$RUMA_{area} = \left(\frac{|4800 - 1200|}{4800} \right) \times 100 \quad (5.8)$$

$$RUMA_{area} = 75\%$$

Os resultados aqui demonstrados podem ser aferidos na tela do sistema de avaliação construído, exibido pela Figura 5.17 onde (a) é a interface do sistema de avaliação de segmentação e (b) demonstra os resultados obtidos pelos testes anteriormente descritos.



(a)

	ValorResultado	Tipo	ImagemSegmentada	ImagemReferencia
▶	0	RUMArea	C:\Celulas\Image...	C:\Celulas\Image...
	75	RUMArea	C:\Celulas\Image...	C:\Celulas\Image...

(b)

Figura 5.17 - Sistema de avaliação de segmentação de imagens

Para aferir o funcionamento do RUMA em perímetro foi aplicado um procedimento de forma similar a área, considerando o valor calculado do perímetro, em oposição ao da área em cada uma das imagens.

Vale ressaltar aqui que a avaliação de segmentações de imagens médicas apresenta uma grande dificuldade dado que as imagens de referência são influenciadas pelos problemas inter-observador e intra-observador (FERREIRA, 2001). Nesta primeira, diferentes

patologistas podem chegar a segmentações divergentes de um mesmo artefato, já no segundo diferentes segmentações podem ser obtidas da mesma estrutura por um único patologista em espaços diferentes de tempo.

Com o propósito de avaliação quantitativa das estruturas segmentadas neste trabalho adotou-se o RUMA. As imagens de referência são segmentadas por um citopatologista e formam a amostra de testes descrita na página 62.

O citopatologista, a partir de um ROI pertencente à amostra, gerou a segmentação completamente manual do núcleo, utilizando-se de um software de editoração eletrônica de imagens. Partindo do modelo gerado pelo profissional da área da saúde foi construída uma imagem binária, conforme pode ser verificado na Figura 5.18.

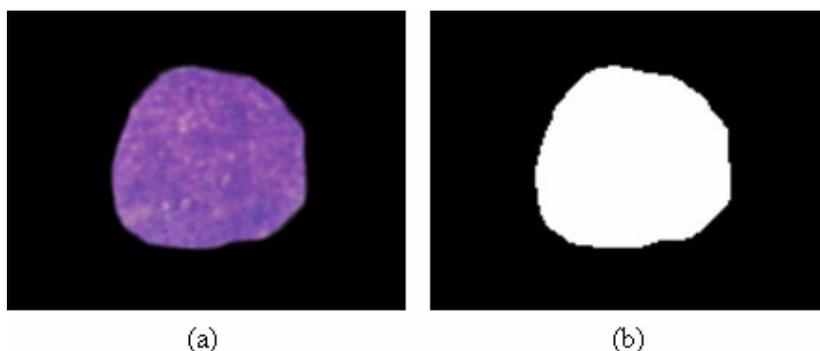


Figura 5.18 - (a) Imagem segmentada manualmente pelo patologista; (b) Máscara binária baseada na segmentação manual

A máscara binária exemplificada na Figura 5.18 (b) é usada como referência, R_f , para o cálculo do RUMA, área e perímetro, já o parâmetro S_f é obtido no *pipeline* apresentado na seção 5.3 na página 62 e descrito na etapa de seleção da área. Os dados obtidos a partir deste protocolo encontram-se tabulados a seguir.

Tabela 5.2 - Resultados do RUMA em casos negativos

Paciente	RUMA área %	RUMA perímetro %
91120-N-07	11,164	6,247
91120-N-14	12,228	4,564
91120-N-32	8,462	4,851
91150-N-22	7,687	4,526
91150-N-23	6,228	3,375
277883-N-21	11,058	4,318
277883-N-22	12,793	7,062

Paciente	RUMA área %	RUMA perímetro %
277883-N-31	10,734	4,620
277963-N-25	18,610	12,739
277963-N-28	7,968	9,450
277963-N-38	7,787	9,053
278046-N-05	6,791	2,767
278046-N-06	1,149	1,268
278046-N-14	14,232	3,834
278331-N-11	11,420	7,217
278331-N-15	10,333	7,800
278331-N-19	26,107	13,161
278427-N-05	0,852	1,021
278427-N-23	1,212	2,038
278427-N-26	3,632	1,675
278444-N-10	12,896	8,509
278444-N-11	3,081	3,625
278444-N-26	21,713	12,267
278501-N-11	4,396	1,471
278586-N-27	4,092	3,784
278586-N-37	10,305	5,691
278586-N-40	7,172	6,486

Tabela 5.3 - Resultados do RUMA em casos positivos

Paciente	RUMA área %	RUMA perímetro %
91416-P-01	6,691	4,300
91416-P-08	4,322	0,573
91416-P-31	25,148	19,186
91421-P-04	3,860	12,768
91421-P-34	17,819	17,744
276691-P-04	13,404	15,043
276691-P-30	2,703	5,280
276942-P-07	0,189	1,950
276942-P-08	4,427	7,484
276942-P-09	4,264	4,807
277206-P-30	4,183	10,085

Paciente	RUMA área %	RUMA perímetro %
277206-P-35	4,167	3,803
277206-P-39	2,444	3,895
277395-P-04	4,322	5,975
277395-P-10	6,298	5,140
277448-p-15	19,794	10,111
277448-p-25	22,015	11,633
277778-P-19	3,969	2,871
277778-P-35	7,982	6,339
277867-P-17	9,337	6,723
277867-P-22	15,125	7,541
278094-P-08	0,626	3,169
278094-P-19	3,236	3,190
278094-P-24	10,275	3,271
291418-P-06	13,478	8,145
291418-P-20	24,243	14,736
291418-P-27	4,811	7,730
291418-P-36	9,778	7,316

A Tabela 5.2 e a Tabela 5.3 demonstram os dados resultantes de experimentos onde a segmentação estudada e proposta obteve sucesso, a nomenclatura utilizada para identificar o paciente é um número arábico, que representa o código do paciente no hospital onde as imagens foram obtidas, seguido da letra P ou N; onde P é para casos positivos de câncer e N para negativos. O segundo número arábico denota a imagem estudada.

Verifica-se que pela segmentação, semi-automática se obteve cerca de 80% de sucesso na separação dos núcleos. Os valores encontrados a partir do RUMA são muito próximos das imagens de referência, sendo que muitos apresentam RUMA inferior a 10%.

5.7 Discussões

Esta seção do trabalho pretende detalhar o processamento de imagens executado em dois casos pertencentes à amostra estudada. O primeiro caso negativo enquanto que o segundo é um caso positivo de câncer.

A seguir, na Figura 5.19, é demonstrado o caso negativo, identificado como 278444-N-26, pertencente à amostra de testes executados, nesta pode-se verificar detalhadamente os passos executados.

Figura 5.19 (a): ROI selecionado pelo patologista

Figura 5.19 (b): Canal verde da imagem

Figura 5.19 (c): Ajuste de brilho e contraste

Figura 5.19 (d): Detecção das bordas

Figura 5.19 (e): *Linking edge*

Figura 5.19 (f): Seleção da área

Figura 5.19 (g): Negativo da Imagem

Figura 5.19 (h): Soma com o ROI

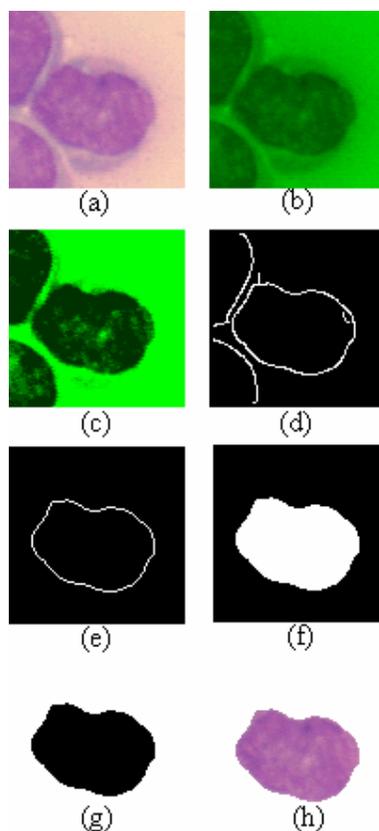


Figura 5.19 - Caso negativo para discussão

Como resultado das características morfométricas obteve-se: a área como 2388, perímetro no valor de 191 e circularidade igual a 1,21.

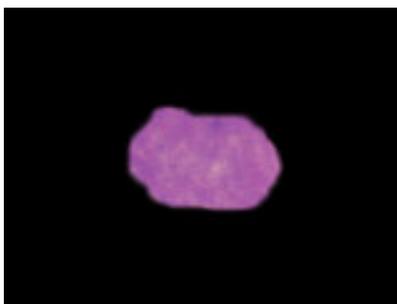


Figura 5.20 - Imagem de referência

O ROI selecionado e exibido na Figura 5.19 (a) foi segmentado de forma manual por um patologista. O resultado desta operação pode ser verificado na Figura 5.20, o compute do RUMA área e perímetro envolvendo este caso de teste é demonstrado na Tabela 5.2, sendo 21,713 e 12,267 respectivamente.

Pode-se verificar que os números elevados neste caso de teste são facilmente justificáveis quando observados a Figura 5.21.

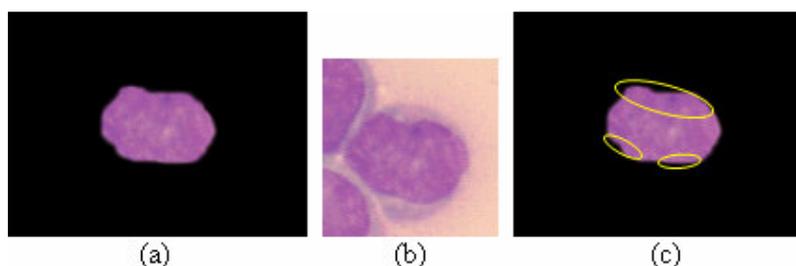


Figura 5.21 - Comparação entre a imagem segmentada manualmente e o ROI

A Figura 5.21 (a) é a imagem segmentada de forma manual pelo patologista, já (b) é o ROI selecionado, verifica-se grandes diferenças em todos os limites, quando comparamos (a) com (b), as divergências mais elevadas são assinaladas em (c).

O caso positivo selecionado para discussão é identificado com 291418-P-06. A Figura 5.22 (a) apresenta o ROI e (b) é o resultado da segmentação executada pelo protótipo computacional construído neste trabalho.

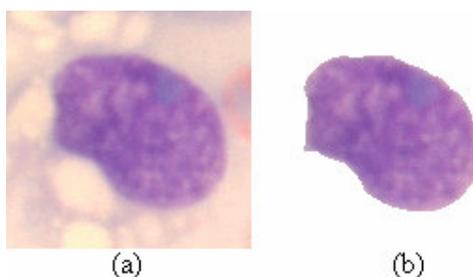


Figura 5.22 - Caso 291418-P-06

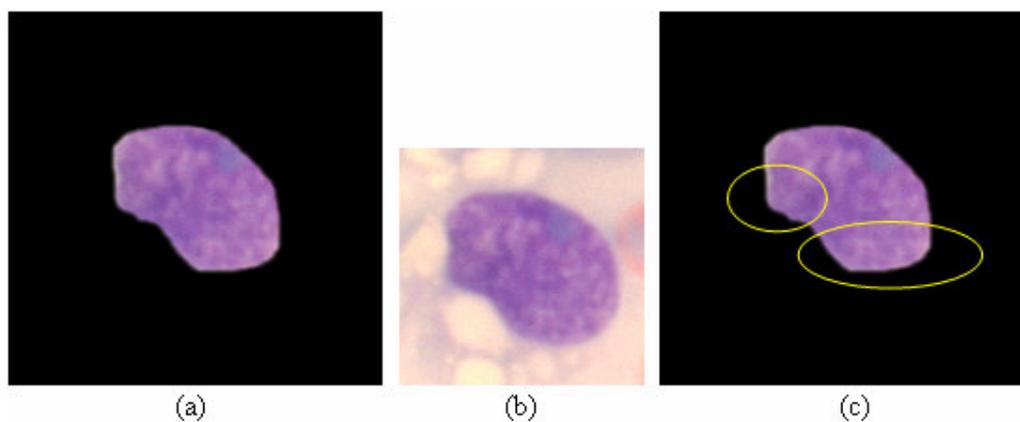


Figura 5.23 - Comparação entre a imagem segmentada manualmente e o ROI

Observa-se pela Figura 5.23 que muitos detalhes, principalmente nas bordas que delimitam as partes curvas do objeto foram extraídas de forma errônea, tais detalhes estão presente em grande parte na Figura 5.22 (b).

Esta seção objetivou esclarecer as técnicas utilizadas para construção do protótipo computacional aplicada no estudo biomédico assim como e resultados conquistados a partir do mesmo, além da avaliação realizada com as segmentações provenientes deste estudo. O próximo capítulo descreve as considerações finais e trabalhos futuros ligados ao estudo aqui desenvolvido.

CONSIDERAÇÕES FINAIS

Para o objetivo de embasamento teórico sobre os temas de segmentação, realce e técnicas de quantificação de estruturas em imagens médicas realizou-se uma pesquisa geral dos modelos existentes em cada uma dessas áreas através da literatura e trabalhos publicados referentes ao tema estudado.

O primeiro passo do trabalho, descrito no capítulo dois, deteve-se exatamente na busca de informações de relevância técnica a respeito de métodos usados em imagens médicas objetivando um conhecimento geral dessas. Dentre essas, muitas estão presentes na maioria dos sistemas que possuem o propósito da análise de imagens médicas com o intuito de inferir dados sobre as mesmas e com isso auxiliar na melhoria do processo de diagnóstico.

Seguindo o conteúdo apresentado nesse projeto, a sessão três procurou construir, representar e reproduzir ensaios existentes na literatura pesquisada. Para tanto, ocorreu à seleção de duas ferramentas amplamente utilizadas em pesquisas com propósitos similares aos objetivos desse estudo.

A seção 4 dessa monografia preocupou-se em delinear dois estudos, um que possui informações de domínio da área da saúde onde os focos de atuação são definidos tendo como base motivações dessa área, outro foi relatado com fins tecnológicos e com objetivos relacionados a interesses provenientes da área de exatas, mais especificamente, ciência da computação. Tais propostas relacionam-se conforme apresentado neste mesmo capítulo, que também define os requisitos necessários para construção de um protótipo computacional com aderência as necessidades do estudo biomédico, além dos objetivos da ferramenta computacional.

Quanto aos objetivos alcançados com o desenvolvimento do sistema apresentado pelo capítulo 5, pode-se verificar que os específicos definidos:

1 - Pré-Processamento: nesta etapa se verificou que o canal verde possui um maior contraste nas imagens obtidas no estudo biomédico e ainda o ajuste de brilho e contraste que removem grande parte das estruturas destoantes e não necessárias da imagem de trabalho.

2 - A segmentação realizada a partir da detecção de bordas mostrou-se eficiente na maioria dos casos da amostra de estudo, entretanto, ajustes manuais foram necessários. Em futuros estudos pretende-se, a partir da ferramenta aqui construída, disponibilizar outras técnicas que, de forma híbrida, possam ajudar na segmentação automática ou semi-automática dos núcleos celulares. Entretanto, para casos não sucessíveis a estes modelos de segmentação é necessária a construção de rastreadores de borda que facilitam a seleção manual do núcleo.

3 - Quantificação: esta por sua vez foi conduzida tendo como base os estudos desenvolvidos e publicados pelo TICAS, contudo primeiramente optou-se por construir apenas três características morfométricas, as demais necessárias são continuação deste estudo.

Com as afirmações anteriores se verifica que este estudo contribui para elucidar algumas das muitas técnicas que podem ser aplicadas nas segmentações e na quantificação de estruturas celulares, permitindo que a pesquisa seja estendida abrangendo as dificuldades aqui encontradas, tais como novos modelos de segmentação e rastreadores de borda, além da construção de outras características morfométricas necessárias para completa assinatura nuclear. Ainda na parte de possibilidade de trabalhos futuros destaca-se a pesquisa de modelos de inferência, tais como redes neurais, que adotadas em situações específicas de escopo limitado podem ajudar o profissional da saúde na etapa de diagnóstico.

Fundamentalmente a pesquisa de segmentação deve ser estudada e aprimorada, através do uso do casamento de técnicas que permitam facilitar o trabalho dos patologistas, além de fornecerem artefatos mais confiáveis para a etapa de quantificação.

Na contribuição deste trabalho fica ainda o sistema e o estudo realizado para avaliação quantitativa de segmentações. O software aqui idealizado e construído pode ser aprimorado para atender outras técnicas de avaliação, podendo assim ser utilizado em outros projetos que objetivam a avaliação ou comparação quantitativa entre algoritmos de segmentação.

Conclui-se esse trabalho tendo adquirido conhecimento nas etapas de análise e processamento de imagens médicas abordadas, sendo clara a pretensão de continuidade do trabalho em momentos futuros.

REFERÊNCIAS BIBLIOGRÁFICAS

ABDOU. Ikram E. and PRATT. William K. Quantitative Design and Evaluation of Enhancement / Thresholding Edge Detectors, Proceedings of the IEEE, vol. 67, no. 5, pp. 753-763, 1979.

AMBRÓSIO, Paulo E. Redes neurais artificiais no apoio ao diagnóstico diferencial de lesões intersticiais. São Paulo: 2002. Dissertação (Mestrado em Ciências, Física Aplicada à Medicina e Biologia) – Departamento de Física e Matemática, USP, 2002.

ANDREOLLA, Huander F. Correlação da Infecção por *Helicobacter pylori* com a ocorrência de processo aterosclerótico em camundongos C57BL/6 submetidos à dieta hiperlipidêmica. Novo Hamburgo: 2006. Projeto de Trabalho de Conclusão de Curso – Instituto de Ciências da Saúde, Centro Universitário Feevale, 2006.

ANDREOLLA F. e BONA L. A relação da infecção por *Helicobacter pylori* por ocorrência de aterosclerose em camundongos C57BL/6 submetidos à dieta hiperlipidêmica In II Congresso Sul-Brasileiro de Biomedicina. 2006.

BANKMAN, I. Segmentation in Handbook of medical imaging: processing and analysis, San Diego, CA: Academic Press, 2000, Johns Hopkins University, p. 67-68.

BANKMAN, I, SPISZ, T. e PAVLOPOULOS S. TWO-Dimensional Shape and Texture Quantification in Handbook of medical imaging: processing and analysis, San Diego, CA: Academic Press, 2000, Johns Hopkins University, p. 215-230.

BANKS, E.R.; JENNINGS, C.D.; JACOBS, S.; DAVEY, D.D. Comparative assessment of DNA analysis in effusions by image analysis and flow cytometry. Diagn Cytopathol. 1994;10:62-66; discussion 66-67.

BARTELS, P.H.; SILVA, V.D.; MONTIRONI, R.; HAMILTON, P.W.; THOMPSON, D.; VAUGHT, L. et al. Chromatin texture signatures in nuclei from prostate lesions. Anal Quant Cytol Histol 1998; 20(5):397-406.

BEARE R. and LEHMANN G. The watershed transform in ITK - discussion and new developments. Insight Journal - Junho 2006. Disponível em: <http://insight-journal.org/dspace/bitstream/1926/202/2/The_watershed_transform_in_ITK_-_discussion_and_new_developments.pdf> Acesso em: 15 Nov. 2006.

BEDROSSIAN C.W.: Malignant Effusions. A multimodal Approach to Cytologic Diagnosis. Igaku-Shoin Medical Publishers, Inc. New York, N.Y., 1991.

BERSCH, V.P.; SILVA, V.D.; OSVALDT, A.B.; da COSTA, M.S.; ROHDE, L., Mossmann D. Digital karyometry in pancreatic adenocarcinoma. *Anal Quant Cytol Histol.* 2003 Apr;25(2):108-14.

BODDINGTON, M.M.; IN COLEMAN, D.V.; CHAPMAN P.A.: *Clinical Cytotechnology. Serous Effusions.* Butterworths, London, UK, 1989; p 271-282.

CANNY, John (1986). A computational approach to edge detection. In *IEEE: Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679-698.

CARVALHO, Luis Alberto Vieira de; BRUNO, Odemir Martinez. Técnicas diferentes para análise de imagens de Plácido podem melhorar precisão da videoceratografia. *Arq. Bras. Oftalmol.*, São Paulo, v. 68, n. 6, 2005. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0004-27492005000600016&lng=pt&nrm=iso>. Acesso em: 24 Nov 2006. doi: 10.1590/S0004-27492005000600016.

CASELLES V., KIMMEL R. and SAPIRO G. Geodesic active contours. *International Journal In Computer Vision*, 22(1):61–97, 1997. 9.3.3

COM. COM: Component Object Model Technologies. Disponível em:<<http://www.microsoft.com/com/default.aspx>>. Acesso em 30 maio 2007.

CLS. What is the Common Language Specification? Disponível em: <[http://msdn2.microsoft.com/en-us/library/12a7a7h3\(VS.71\).aspx](http://msdn2.microsoft.com/en-us/library/12a7a7h3(VS.71).aspx)>. Acesso em 20 maio. 2007.

DAVIS, Clodoveu A. Jr. PixelWare: Um Sistema de Processamento Digital de Imagens. Minas Gerais: 1992. Dissertação (Mestrado em Ciência da Computação) - Departamento de Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais, 1992.

EDDINS Steve. Cell segmentation. Disponível em:<<http://blogs.mathworks.com/steve/?p=60>>. Acesso em 30 maio 2007.

EDDINS Steve. Image overlay. Disponível em:<<http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=10502>>. Acesso em 30 maio 2007.

FALCÃO, Alexandre X. Visualização de Volumes Aplicada à Área Médica. São Paulo: 1993. Tese (Mestrado em Engenharia Elétrica) – Departamento de Engenharia de Computação e Automação Industrial, UNICAMP, 1993.

FERREIRA. Carlos, SANTOS Beatriz S., SILVA José S., SILVA Augusto, AGOSTINHO Pedro e TEIXEIRA Luisa. Avaliação Quantitativa de um Método Automático de Extração de Contornos em Tomogramas Pulmonares, *Actas das VIII Jornadas de Classificação e Análise de Dados - JOCLAD 2001*, pp. 62-65, Porto - Portugal, 2001 – Disponível em: <<http://www.uc.pt/pessoal/jsilva/pdf/Ferreira2001-JOCLAD.pdf>> Acesso em: 5 junh. 2007.

FRANTZ, M.A. Caracterização de núcleos celulares benignos e malignos por análise de imagem digital em derrames pleurais (Tese de doutorado). Porto Alegre; RS: Universidade Federal do Rio Grande do Sul, 2004.

GONZALES, Rafael. C. e WOODS Richard E. *Processamento de Imagens Digitais*. São Paulo: Edgard Blücher, 2000.

GONZALES, Rafael. C., WOODS Richard E e EDDINS Steven L *Digital Image processing using MATLAB*. Upper Saddle River, N.J: Prentice Hall, 2002.

IBÁÑEZ Luis et all. *The ITK Software Guide*. 2nd. Edition. 2005.

KASS M., WITKIN A. and TERZOPOULOS D. Snakes: active contour models In: *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1987.

KERWIN W., OSRRIAN N. e PRINCE J. *Image Processing and Analysis in Tagged Cardiac MRI in Handbook of medical imaging: processing and analysis*, San Diego, CA: Academic Press, 2000, Johns Hopkins University, p. 375-392.

KOVALEV, V. e PETROU M., *Texture Analysis in Three Dinlensions as a Cue to Medical Diagnosis in Handbook of medical imaging: processing and analysis*, San Diego, CA: Academic Press, 2000, Johns Hopkins University, p. 231-248.

KOVESI P. D. *MATLAB and Octave Functions for Computer Vision and Image Processing*. School of Computer Science & Software Engineering, The University of Western Australia. Disponível em: <<http://www.csse.uwa.edu.au/~pk/research/matlabfns/>>. Acesso em 9 de Abril de 2007.

LCI. The Laboratory for Computational Intelligence da University of British Columbia - UBC.ca. Disponível em:<<http://www.cs.ubc.ca/nest/lci/>>. Acesso em 30 maio 2007.

MARQUES, P. M. de A. Diagnóstico auxiliado por computador na radiologia. *Radiol Bras.* [online]. set./out. 2001, vol.34, no.5 p.285-293. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-39842001000500008&lng=pt&nrm=iso>. ISSN 0100-3984.Acesso em: 20 out de 2006

MATLAB. MATLAB - The Language of Technical Computing. Disponível em:<<http://www.mathworks.com/products/matlab/>>. Acesso em 22 junho 2007.

MATLAB BUILDER FOR .NET. Disponível em: <<http://www.mathworks.com/products/netbuilder/>>. Acesso em 20 abr. 2007.

MCKEE, G.T. *Citopatologia*, pág 292. Editora Artes Médicas. São Paulo 1997.

MEER, P., AND GEORGESCU, B. 2001. Edge detection with embedded confidence. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 23, 12, 1351–1365.

MOSSMANN, J. & ANDREOLLA F. Sistema para Auxílio ao Diagnóstico (SAD) de Aterosclerose em Modelo Animal In II Congresso Sul-Brasileiro de Biomedicina. 2006.

OLIVEIRA, Julio C. de. Índice para Avaliação de Segmentação (IAVAS): Uma Aplicação em Agricultura. São José dos Campos: 2003. Dissertação (Mestrado em Sensoriamento Remoto), INPE, 2003.

PARANJAPE, Raman B. Fundamental Enhancement Techniques in Handbook of medical imaging: processing and analysis, San Diego, CA: Academic Press, 2000, Johns Hopkins University, p. 3-18.

QIU, Peihua. Image processing and jump regression analysis. Hoboken, N.J. John Wiley, 2005.

RATIB O. Quantitative Analysis of Cardiac Function in Handbook of medical imaging: processing and analysis, San Diego, CA: Academic Press, 2000, Johns Hopkins University, p. 359-374.

RANGAYYAN, M. Enhancement in Handbook of medical imaging: processing and analysis, San Diego, CA: Academic Press, 2000, Johns Hopkins University, p. 1-2.

ROBBINS, G.T.; SHOEN, F.T.; COTRAN, R.S.; KUMAR, V. Patologia Estrutural e Funcional, pág 747-748, 5ª Edição, Editora Guanabara Koogan 1996.

ROGOWSKA, J. Overview and Fundamentals of Medical Image Segmentation in Handbook of medical imaging: processing and analysis, San Diego, CA: Academic Press, 2000, Johns Hopkins University, p. 69-85.

ROSITO, Mário Antonello. Caracterização de Núcleos Celulares no Adenocarcinoma Primário de Reto por Análise de Imagem Digital. Porto Alegre: 2002. Tese (Doutorado em Medicina) – Departamento de Medicina, UFRGS, 2002.

ROSITO, M.A.; MOREIRA, L.F.; SILVA, V.D.; DAMIN, D.C.; PROLLA, J.C. Nuclear chromatin texture in rectal cancer. Relationship to tumor stage. Anal Quant Cytol Histol. 2003 Feb;25(1):25-30.

RUSS, John C. The image processing handbook - 3rd ed. Materials Science and Engineering Department North Carolina State University Raleigh - North Carolina. 1998.

SILVA, José S. S. da. Segmentação Pulmonar em Estudos de Tomografia Axial Computadorizada. Portugal - Aveiro: 2005. Dissertação (Doutorado em Engenharia Electrotécnica) – Departamento de Eletrônica e Telecomunicações, Universidade de Aveiro, 2005.

SILVA, V.D.; PROLLA, J.C.; SHARMA, P.; SAMPLINER, R.; THOMPSON, D., BARTELS, P.H. Karyometry in Barrett's esophagus. Anal Quant Cytol Histol. 2001 Feb;23(1):40-6.

SILVA, V.D.; PROLLA, J.C.; DIEHL, A.R.; BALDO, M.F.; MULLER, R.L. Comparison of conventional microscopy and digitized imaging for diagnosis in serous effusions. Anal Quant Cytol Histol 1997;19(3):202-6.

SILVA, V.D. Digital Video Microscopy in Pathology. Pathologica 1999;91(2):124-127.

SILVA, V.D.; MONTIRONI, R.; THOMPSON, D.; BARTELS, H.G.; VAUGHT, L.; HAMILTON, P.W.; BARTELS, P.H. Chromatin texture in high grade prostatic intraepithelial neoplasia and early invasive carcinoma. Anal Quant Cytol Histol. 1999 Apr;21(2):113-20.

YOO T. S. Introduction In Insight into images : principles and practice for segmentation, registration, and image analysis, Wellesley, Mass. A K Peters, 2004, National Library of Medicine, p. 3 - 18.

YUAN, Bo, GALLAGHER, Marcus, and CROZIER, Stuart. MRI magnet design: search space analysis, EDAs and a real-world problem with significant dependencies. In Proceedings of the 2005 Conference on Genetic and Evolutionary Computation (Washington DC, USA, June 25 - 29, 2005). H. Beyer, Ed. GECCO '05. ACM Press, New York, NY, 2141-2148.

TAO, L.C. Cytopathology of Malignant Effusions. American Society of Clinical Pathologists Chicago, USA, 1996.

WIED, G.L.; BARTEL, P.H.; BAHR, G.F.; OLDFIELD, D.G. Taxonomic Intra-Cellular Analytic Systems (TICAS) for cell identification. Acta Cytol 1968;12:180-204.

XU, Chenyang; PHAM, Dzung L and PRINCE, Jerry L. Medical Image Segmentation Using Deformable Models. Handbook on Medical Imaging, v. 3, p. 129 -168, maio 2000. Disponível em: <<http://iacl.ece.jhu.edu/pubs/p119b.pdf>>. Acesso em: 20 julh.2006.

XU C, YEZZI A., Jr., and PRINCE J. L., A Summary of Geometric Level-Set Analogues for a General Class of Parametric Active Contour and Surface Models, in Proc. of 2001 IEEE Workshop on Variational and Level Set Methods in Computer Vision, p. 104-111, July 2001. Disponível em: < <http://iacl.ece.jhu.edu/~chenyang/research/pubs/vlsm01.pdf>>. Acesso em: 24 julh.2006.

XU CY and PRINCE JL Active Contours, Deformable Models, and Gradient Vector Flow. Disponível em: <<http://iacl.ece.jhu.edu/projects/gvf/>>. Acesso em 20 mar.2006

ZHANG, X.W., SONG, J.Q., LYU, M.R., CAI, S.J. Extraction of karyocytes and their components from microscopic bone marrow images based on regional color features. Pattern Recognition, Volume 37, Number 2, February 2004, pp. 351-361(11). Disponível em: Acesso em: 5 de junh. 2007

ANEXOS

ANEXO A ALGORITMO DA MÉDIA

Ensaio:	Realce – Média
Fonte:	Adaptado de exemplos em The ITK Software Guide (IBÁÑEZ, 2005, p. 164-166)
Linguagem:	C++

```

#include "itkImage.h"
#include "itkImageRegionIterator.h"
#include "itkMeanImageFilter.h"
#include "itkImageFileReader.h"
#include "itkImageFileWriter.h"

int main()
{

    typedef unsigned char InputPixelType;
    typedef unsigned char OutputPixelType;

    typedef itk::Image< InputPixelType, 2 > InputImageType;
    typedef itk::Image< OutputPixelType, 2 > OutputImageType;

    typedef itk::ImageFileReader< InputImageType > LeituraTipo;
    typedef itk::ImageFileWriter< OutputImageType > EscritaTipo;

    LeituraTipo::Pointer leitura = LeituraTipo::New();
    EscritaTipo::Pointer escrita = EscritaTipo::New();

    //Caminho físico da imagem original
    //Exemplo: leitura->SetFileName( "C:/EnsaiosTC/original.png" );
    leitura->SetFileName( "" );

    //Caminho físico para gravação da imagem de destino
    //Exemplo: escrita->SetFileName( "C:/EnsaiosTC/saida.png" );
    escrita->SetFileName( "" );

    typedef itk::MeanImageFilter<InputImageType, OutputImageType >
    TipoFiltro;

    TipoFiltro::Pointer filtro = TipoFiltro::New();

    InputImageType::SizeType indexRadius;

    indexRadius[0] = 1;
    indexRadius[1] = 1;

    filtro->SetRadius( indexRadius );

    filtro->SetInput( leitura->GetOutput() );
    escrita->SetInput( filtro->GetOutput() );
    escrita->Update();
    return 0;
}

```

ANEXO B ALGORITMO - CONNECTED THRESHOLD

Ensaio:	Segmentação - Connected Threshold
Fonte:	Adaptado de exemplos em The ITK Software Guide (IBÁÑEZ, 2005, p. 494-497)
Linguagem:	C++

```

#include "itkConnectedThresholdImageFilter.h"
#include "itkImage.h"
#include "itkCastImageFilter.h"
#include "itkCurvatureFlowImageFilter.h"
#include "itkImageFileReader.h"
#include "itkImageFileWriter.h"

int main()
{
    // Pontos sementes
    int seedX = 48;
    int seedY = 67;

    typedef float InternalPixelType;
    const unsigned int Dimension = 2;
    typedef itk::Image< InternalPixelType, Dimension > InternalImageType;

    //Limiar Mínimo e Máximo
    const InternalPixelType lowerThreshold = 180;
    const InternalPixelType upperThreshold = 210;

    typedef unsigned char OutputPixelType;
    typedef itk::Image< OutputPixelType, Dimension > OutputImageType;
    typedef itk::CastImageFilter< InternalImageType, OutputImageType >
        CastingFilterType;
    CastingFilterType::Pointer caster = CastingFilterType::New();

    typedef itk::ImageFileReader< InternalImageType > ReaderType;
    typedef itk::ImageFileWriter< OutputImageType > WriterType;

    ReaderType::Pointer reader = ReaderType::New();
    WriterType::Pointer writer = WriterType::New();

    //Caminho físico da imagem original
    //Exemplo: reader->SetFileName(
"C:/EnsaiosTC/Regiao/_cThreshold/original.png" );
    reader->SetFileName( "" );

    //Caminho físico para gravação da imagem de destino
    //Exemplo: writer->SetFileName(
"C:/EnsaiosTC/Regiao/_cThreshold/saida3.png" );
    writer->SetFileName( "" );

    typedef itk::CurvatureFlowImageFilter< InternalImageType,
InternalImageType >
        CurvatureFlowImageFilterType;

    CurvatureFlowImageFilterType::Pointer smoothing =
        CurvatureFlowImageFilterType::New();

```

```

typedef itk::ConnectedThresholdImageFilter< InternalImageType,
                                           InternalImageType
ConnectedFilterType;
                                           >

ConnectedFilterType::Pointer      connectedThreshold
ConnectedFilterType::New();
                                           =

smoothing->SetInput( reader->GetOutput() );
connectedThreshold->SetInput( smoothing->GetOutput() );
caster->SetInput( connectedThreshold->GetOutput() );
writer->SetInput( caster->GetOutput() );

smoothing->SetNumberOfIterations( 5 );
smoothing->SetTimeStep( 0.125 );

connectedThreshold->SetLower( lowerThreshold );
connectedThreshold->SetUpper( upperThreshold );

connectedThreshold->SetReplaceValue( 255 );

InternalImageType::IndexType  index;

index[0] = seedX;
index[1] = seedY;

connectedThreshold->SetSeed( index );

try
{
    writer->Update();
}
catch( itk::ExceptionObject & excep )
{
    std::cerr << "Exception caught !" << std::endl;
    std::cerr << excep << std::endl;
}

return 0;
}

```

ANEXO C ALGORITMO - NEIGHBORHOOD CONNECTED

Ensaio:	Segmentação – Neighborhood Connected
Fonte:	Adaptado de exemplos em The ITK Software Guide (IBÁÑEZ, 2005, p. 500-503)
Linguagem:	C++

```

#include "itkNeighborhoodConnectedImageFilter.h"
#include "itkImage.h"
#include "itkCastImageFilter.h"
#include "itkCurvatureFlowImageFilter.h"
#include "itkImageFileReader.h"
#include "itkImageFileWriter.h"

int main()
{

    typedef float InternalPixelType;
    const unsigned int Dimension = 2;
    typedef itk::Image< InternalPixelType, Dimension > InternalImageType;

    typedef unsigned char OutputPixelType;
    typedef itk::Image< OutputPixelType, Dimension > OutputImageType;

    typedef itk::CastImageFilter< InternalImageType, OutputImageType >
        CastingFilterType;
    CastingFilterType::Pointer caster = CastingFilterType::New();

    typedef itk::ImageFileReader< InternalImageType > ReaderType;
    typedef itk::ImageFileWriter< OutputImageType > WriterType;

    ReaderType::Pointer reader = ReaderType::New();
    WriterType::Pointer writer = WriterType::New();

    //Caminho físico da imagem original
    //Exemplo: reader->SetFileName(
"C:/EnsaiosTC/Regiao/neighborhood/original.png" );
    reader->SetFileName( "" );

    //Caminho físico para gravação da imagem de destino
    //Exemplo: writer->SetFileName(
"C:/EnsaiosTC/Regiao/neighborhood/saida3.png" );
    writer->SetFileName( "" );

    // Valores de Limiar
    InternalPixelType lowerThreshold = 180;
    InternalPixelType upperThreshold = 210;

    // Ponto Semente
    InternalImageType::IndexType index;

    index[0] = 107;
    index[1] = 69;

    typedef itk::CurvatureFlowImageFilter<InternalImageType,
InternalImageType>

```

```

    CurvatureFlowImageFilterType;

    CurvatureFlowImageFilterType::Pointer smoothing =
        CurvatureFlowImageFilterType::New();

    typedef itk::NeighborhoodConnectedImageFilter<InternalImageType,
                                                InternalImageType
    ConnectedFilterType;
    ConnectedFilterType::Pointer neighborhoodConnected =
    ConnectedFilterType::New();

    smoothing->SetInput( reader->GetOutput() );
    neighborhoodConnected->SetInput( smoothing->GetOutput() );
    caster->SetInput( neighborhoodConnected->GetOutput() );
    writer->SetInput( caster->GetOutput() );

    smoothing->SetNumberOfIterations( 5 );
    smoothing->SetTimeStep( 0.125 );

    neighborhoodConnected->SetLower( lowerThreshold );
    neighborhoodConnected->SetUpper( upperThreshold );

    // MASCARA 5 x 5
    InternalImageType::SizeType radius;

    radius[0] = 2;
    radius[1] = 2;

    neighborhoodConnected->SetRadius( radius );

    neighborhoodConnected->SetSeed( index );
    neighborhoodConnected->SetReplaceValue( 255 );

    try
    {
        writer->Update();
    }
    catch( itk::ExceptionObject & excep )
    {
        std::cerr << "Exception caught !" << std::endl;
        std::cerr << excep << std::endl;
    }

    return 0;
}

```

ANEXO D ALGORITMO - WATERSHED

Ensaio:	Segmentação – Watershed
Fonte:	Adaptado de exemplos em The ITK Software Guide (IBÁÑEZ, 2005, p. 516-520)
Linguagem:	C++

```

#include <iostream>
#include "itkVectorGradientAnisotropicDiffusionImageFilter.h"
#include "itkVectorGradientMagnitudeImageFilter.h"
#include "itkWatershedImageFilter.h"
#include "itkImageFileReader.h"
#include "itkImageFileWriter.h"
#include "itkVectorCastImageFilter.h"
#include "itkUnaryFunctorImageFilter.h"
#include "itkScalarToRGBPixelFunctor.h"

int main()
{
    // Parâmetros de entrada para a segmentação
    //*****
    float conductance = 2.0;
    int iterations = 10;
    float threshold = 0.001 ;
    float level = 0.15;
    int pComponents = 0;
    //*****

    typedef itk::RGBPixel<unsigned char>      RGBPixelType;
    typedef itk::Image<RGBPixelType, 2>      RGBImageType;
    typedef itk::Vector<float, 3>            VectorPixelType;
    typedef itk::Image<VectorPixelType, 2>   VectorImageType;
    typedef itk::Image<unsigned long, 2>     LabeledImageType;
    typedef itk::Image<float, 2>             ScalarImageType;

    typedef itk::ImageFileReader<RGBImageType> FileReaderType;
    typedef itk::VectorCastImageFilter<RGBImageType, VectorImageType>
        CastFilterType;
    typedef
itk::VectorGradientAnisotropicDiffusionImageFilter<VectorImageType,
        VectorImageType> DiffusionFilterType;
    typedef itk::VectorGradientMagnitudeImageFilter<VectorImageType>
        GradientMagnitudeFilterType;
    typedef itk::WatershedImageFilter<ScalarImageType> WatershedFilterType;

    typedef itk::ImageFileWriter<RGBImageType> FileWriterType;

    FileReaderType::Pointer reader = FileReaderType::New();

    //Leitura da imagem a ser segmentada
    //Exemplo: reader->SetFileName("C:/imgEnsaio/entrada.png");
    reader->SetFileName("");

    CastFilterType::Pointer caster = CastFilterType::New();

```

```

DiffusionFilterType::Pointer diffusion = DiffusionFilterType::New();

diffusion->SetNumberOfIterations( iterations );
diffusion->SetConductanceParameter( conductance );
diffusion->SetTimeStep( 0.125 );

GradientMagnitudeFilterType::Pointer gradient =
GradientMagnitudeFilterType::New();

gradient->SetUsePrincipleComponents( pComponents );

WatershedFilterType::Pointer watershed = WatershedFilterType::New();

watershed->SetLevel( level );
watershed->SetThreshold( threshold );

typedef itk::Functor::ScalarToRGBPixelFunctor<unsigned long>
    ColorMapFunctorType;
typedef itk::UnaryFunctorImageFilter<LabeledImageType,
    RGBImageType, ColorMapFunctorType> ColorMapFilterType;
ColorMapFilterType::Pointer colormapper = ColorMapFilterType::New();

FileWriterType::Pointer writer = FileWriterType::New();

//Saída da imagem, resultado da segmentação
//Exemplo: writer->SetFileName( "C:/imgEnsaio/saida.png" );
writer->SetFileName( "" );

caster->SetInput( reader->GetOutput() );
diffusion->SetInput( caster->GetOutput() );
gradient->SetInput( diffusion->GetOutput() );

watershed->SetInput( gradient->GetOutput() );
colormapper->SetInput( watershed->GetOutput() );
writer->SetInput( colormapper->GetOutput() );

try
{
    writer->Update();
}
catch (itk::ExceptionObject &e)
{
    std::cerr << e << std::endl;
}

return 0;
}

```

ANEXO E ALGORITMO – EDGELINK

Descrição:	Utilizado para união dos segmentos de bordas
Fonte:	(KOVESI, 2007)
Sistema:	MATLAB

```

% Acknowledgement:
% Some of this code is inspired by David Lowe's Link.c function from the
% Vista image processing library developed at the University of British
% Columbia
%   http://www.cs.ubc.ca/nest/lci/vista/vista.html

% Copyright (c) 2001-2007 Peter Kovesi
% School of Computer Science & Software Engineering
% The University of Western Australia
% http://www.csse.uwa.edu.au/
%
% Permission is hereby granted, free of charge, to any person obtaining a
% copy
% of this software and associated documentation files (the "Software"), to
% deal
% in the Software without restriction, subject to the following conditions:
%
% The above copyright notice and this permission notice shall be included
% in
% all copies or substantial portions of the Software.
%
% The Software is provided "as is", without warranty of any kind.

% February 2001 - Original version
% September 2004 - Revised to allow subpixel edge data to be used
% November 2006 - Changed so that edgelist starts and stop at every
% junction
% January 2007 - Prefiltering to to discard isolated pixels and the
% problems they cause(thanks to Jeff Copeland)
% January 2007 - Fixes to ensure closed loops are closed, and a few other
% cases are handled better

function [edgelist, edgeim] = edgelist(im, minlength, location)

    global EDGEIM;           % Some global variables to avoid passing (and
                            % copying) of arguments, this improves speed.
    global ROWS;
    global COLS;
    global RJ;
    global CJ;

    EDGEIM = im ~= 0;        % make sure image is binary.
    EDGEIM = bwmorph(EDGEIM, 'clean'); % Remove isolated pixels
    EDGEIM = bwmorph(EDGEIM, 'skel', Inf); % and make sure edges are
    thinned. I
                                % think using 'skel' is better
    than 'thin'

```

```

% Find endings and junctions in edge data
[RJ, CJ, re, ce] = findendsjunctions(EDGEIM);

EDGEIM = double(EDGEIM); % Convert to double to allow the use of -ve
labelings
[ROWS, COLS] = size(EDGEIM);
edgeNo = 0;

% Perform raster scan through image looking for edge points. When a
% point is found trackedge is called to find the rest of the edge
% points. As it finds the points the edge image pixels are labeled
% with the -ve of their edge No

for r = 1:ROWS
for c = 1:COLS
    if EDGEIM(r,c) == 1
        edgepoints = trackedge(r,c, edgeNo);
        if ~isempty(edgepoints)
            edgeNo = edgeNo + 1;
            edgelist{edgeNo} = edgepoints;
        end
    end
end
end

edgeim = -EDGEIM; % Finally negate image to make edge encodings +ve.

% Eliminate isolated edges and spurs that are below the minimum length

if nargin >= 2 && ~isempty(minlength)
    edgelist = cleanedgelist(edgelist, minlength);

else % Call cleanedgelist with 0 minlength anyway to fix spurrious
nodes
    % that may exist due to problem with EDGELINK at points where
    % junctions are adjacent.
    edgelist = cleanedgelist(edgelist, 0);
end

% If subpixel edge locations are supplied upgrade the integer precision
% edgelist that were constructed with data from 'location'.
if nargin == 3
for I = 1:length(edgelist)
    ind = sub2ind(size(im),edgelist{I}(:,1),edgelist{I}(:,2));
    edgelist{I}(:,1) = real(location(ind))';
    edgelist{I}(:,2) = imag(location(ind))';
end
end

%-----
% TRACKEDGE
%
% Function to track all the edge points associated with a start point.
From a
% given starting point it tracks in one direction, storing the coords of
the

```

```

% edge points in an array and labelling the pixels in the edge image with
the
% -ve of their edge number. This continues until no more connected points
are
% found, or a junction point is encountered. At this point the function
returns
% to the start point and tracks in the opposite direction.
%
% Usage:   edgepoints = trackedge(rstart, cstart, edgeNo)
%
% Arguments:  rstart, cstart  - row and column No of starting point
%              edgeNo        - the current edge number
%              minlength     - minimum length of edge to accept
%
% Returns:   edgepoints      - Nx2 array of row and col values for
%                               each edge point.

function edgepoints = trackedge(rstart, cstart, edgeNo)

    global EDGEIM;
    global RJ;
    global CJ;
    global noPoint;
    global thereIsAPoint;
    global lastPoint;

    noPoint = 0;
    thereIsAPoint = 1;
    lastPoint = 2;

    edgepoints = [rstart cstart];      % Start a new list for this edge.
    EDGEIM(rstart,cstart) = -edgeNo;   % Edge points in the image are
                                        % encoded by -ve of their edgeNo.

    [status, r, c] = nextpoint(rstart,cstart, edgeNo); % Find next
connected                                     % edge point.

    while status ~= noPoint
        edgepoints = [edgepoints      % Add point to point list
                       r      c      ];
        EDGEIM(r,c) = -edgeNo;        % Update edge image

        if status == lastPoint        % We have hit a junction point
            status = noPoint;         % make sure we stop tracking here
        else
            [status, r, c] = nextpoint(r,c, edgeNo); % Otherwise keep going
        end
    end

    % Now track from original point in the opposite direction - but only if
    % the starting point was not a junction point

    if ~isjunction(rstart,cstart)
        % First reverse order of existing points in the edge list
        edgepoints = flipud(edgepoints);

        % ...and start adding points in the other direction.
        [status, r, c] = nextpoint(rstart,cstart, edgeNo);
    end

```

```

while status ~= noPoint
    edgepoints = [edgepoints
                 r   c   ];
    EDGEIM(r,c) = -edgeNo;
    if status == lastPoint
        status = noPoint;
    else
        [status, r, c] = nextpoint(r,c, edgeNo);
    end
end
end

% Final check to see if this edgelist should have start and end points
% matched to form a loop. If the number of points in the list is four
or
% more (the minimum number that could form a loop), and the endpoints
are
% within a pixel of each other, append a copy of the first point to the
% end to complete the loop

if length(edgepoints) >= 4
if abs(edgepoints(1,1) - edgepoints(end,1)) <= 1 && ...
    abs(edgepoints(1,2) - edgepoints(end,2)) <= 1
    edgepoints = [edgepoints
                  edgepoints(1,:)];
end
end

%-----
%
% NEXTPOINT
%
% Function finds a point that is 8 connected to an existing edge point
%

function [status, nextr, nextc] = nextpoint(rp,cp, edgeNo)

    global EDGEIM;
    global ROWS;
    global COLS;
    global RJ;
    global CJ;
    global noPoint;
    global thereIsAPoint;
    global lastPoint;

    % row and column offsets for the eight neighbours of a point
    roff = [-1  0  1  0 -1 -1  1  1];
    coff = [ 0  1  0 -1 -1  1  1 -1];

    r = rp+roff;
    c = cp+coff;

    % Find indices of arrays of r and c that are within the image bounds
    ind = find((r>=1 & r<=ROWS) & (c>=1 & c<=COLS));

```

```

% Search through neighbours and see if one is a junction point
for i = ind
if (any(c(i) == CJ(RJ==r(i)))) && (EDGEIM(r(i),c(i)) ~= -edgeNo)
% This is a junction point that we have not marked as part of
% this edgelist
nextc = r(i);
nextc = c(i);
status = lastPoint;
return; % break out and return with the data
end
end

% If we get here there were no junction points. Search through
neighbours
% and return first connected edge point that itself has less than two
% neighbours connected back to our current edge. This prevents
occasional
% erroneous doubling back onto the wrong segment

checkFlag = 0;
for i = ind
if EDGEIM(r(i),c(i)) == 1
n = neighbours(r(i),c(i));
if sum(n== -edgeNo) < 2
nextc = r(i);
nextc = c(i);
status = thereIsAPoint;
return; % break out and return with the data

else % Remember this point just in case we
checkFlag = 1; % have to use it
rememberr = r(i);
rememberc = c(i);
end

end
end

% If we get here (and 'checkFlag' is true) there was no connected edge
point
% that had less than two connections to our current edge, but there was
one
% with more. Use the point we remembered above.
if checkFlag
nextc = rememberr;
nextc = rememberc;
status = thereIsAPoint;
return; % Break out
end

% If we get here there was no connecting next point at all.
nextc = 0;
nextc = 0;
status = noPoint;

%-----
% Function to test whether a location in the image is a junction point.
% Note that for speed this code has been hard wired into NEXTPOINT.

```

```

function b = isjunction(r,c)
    global RJ;
    global CJ;

    b = any(c == CJ(RJ==r));

%-----
% Function to get the values of the 8 neighbouring pixels surrounding a
point
% of interest. The values are ordered from the top-left point going
% anti-clockwise around the pixel.
function n = neighbours(rp, cp)

    global EDGEIM;
    global ROWS;
    global COLS;

    % row and column offsets for the eight neighbours of a point
    roff = [-1 0 1 1 1 0 -1 -1];
    coff = [-1 -1 -1 0 1 1 1 0];

    r = rp+roff;
    c = cp+coff;

    % Find indices of arrays of r and c that are within the image bounds
    ind = find((r>=1 & r<=ROWS) & (c>=1 & c<=COLS));

    n = zeros(1,8);
    for i = ind
        n(i) = EDGEIM(r(i),c(i));
    end

```

ANEXO F ALGORITMO – CLEANEDGELIST

Descrição:	Utilizado pela função edgelist, responsável por eliminar as bordas isoladas, sem união.
Fonte:	(KOVESI, 2007)
Sistema:	MATLAB

```

% Copyright (c) 2006 Peter Kovesi
% School of Computer Science & Software Engineering
% The University of Western Australia
% pk at csse uwa edu au
% http://www.csse.uwa.edu.au/
%
% Permission is hereby granted, free of charge, to any person obtaining a
copy
% of this software and associated documentation files (the "Software"), to
deal
% in the Software without restriction, subject to the following conditions:
%
% The above copyright notice and this permission notice shall be included
in
% all copies or substantial portions of the Software.
%
% The Software is provided "as is", without warranty of any kind.
%
% December 2006 Original version
% February 2007 A major rework to fix several problems, hope they really
are
% fixed!

```

```
function nedgelist = cleanedgelist(edgelist, minlength)
```

```

    Nedges = length(edgelist);
    Nnodes = 2*Nedges;

    % Each edgelist has two end nodes - the starting point and the ending
point.
    % We build up an adjacency/connection matrix for each node so that we
can
    % determine which, if any, edgelists are connected to a node. We also
% maintain an adjacency matrix for the edges themselves.
    %
    % It is tricky maintaining all this information but it does allow the
% code to run much faster.

    % First extract the end nodes from each edgelist. The nodes are
numbered
    % so that the start node has number 2*edgenumber-1 and the end node has
% number 2*edgenumber
    node = zeros(Nnodes, 2);
    for n = 1:Nedges
        node(2*n-1,:) = edgelist{n}(1,:);
        node(2*n ,:) = edgelist{n}(end,:);
    end

```

```

% Now build the adjacency/connection matrices.
A = zeros(Nnodes); % Adjacency matrix for nodes
B = zeros(Nedges); % Adjacency matrix for edges

for n = 1:Nnodes-1
for m = n+1:Nnodes
    % If nodes m & n are connected
    A(n,m) = node(n,1)==node(m,1) && node(n,2)==node(m,2);
    A(m,n) = A(n,m);

    if A(n,m)
        edgen = fix((n+1)/2);
        edgem = fix((m+1)/2);
        B(edgen, edgem) = 1;
        B(edgem, edgen) = 1;
    end
end
end

% If we sum the columns of the adjacency matrix we get the number of
% other edgelist that are connected to an edgelist
Nconnections = sum(A); % Connection count array for nodes
Econnections = sum(B); % Connection count array for edges

% Check every edge to see if any of its ends are connected to just one
edge.
% This should not happen, but occasionally does due to a problem in
% EDGELINK. Here we simply merge it with the edge it is connected to.
% Ultimately I want to be able to remove this block of code.
% I think there are also some cases that are (still) not properly
handled
% by CLEANEDGELIST and there may be a case for repeating this block of
% code at the end for another final cleanup pass
for n = 1:Nedges
    if ~B(n,n) && ~isempty(edgelist{n}) % if edge is not connected to
itself
        [spurdegree, spurnode, startnode, sconns, endnode, econns] =
connectioninfo(n);
        if sconns == 1
            node2merge = find(A(startnode,:));
            mergenodes(node2merge,startnode);
        end

        if ~isempty(edgelist{n}) % If we have not removed this edge in
% the code above check the other
end.
        if econns == 1
            node2merge = find(A(endnode,:));
            mergenodes(node2merge,endnode);
        end
    end
end
end

% Now check every edgelist, if the edglength is below the minimum
length
% check if we should remove it.

```

```

if minlength > 0

for n = 1:Nedges

[spurdegree, spurnode] = connectioninfo(n);

if ~isempty(edgelist{n}) && edgelistlength(edgelist{n}) < minlength

    % Remove unconnected lists, or lists that are only connected to
    % themselves.
    if ~Econnections(n) || (Econnections(n)==1 && B(n,n) == 1)
        removeedge(n);

    % Process edges that are spurs coming from a 3-way junction.
    elseif spurdegree == 2
        %fprintf('%d is a spur\n',n) %%debug

        linkingedges = find(B(n,:));

        if length(linkingedges) == 1 % We have a loop with a spur
            % coming from the join in the
            % loop
            % Just remove the spur, leaving the loop intact.
            removeedge(n);

        else % Check the other edges coming from this point. If any
            % are also spurs make sure we remove the shortest
one
            spurs = n;
            len = edgelistlength(edgelist{n});
            for i = 1:length(linkingedges)
                spurdegree = connectioninfo(linkingedges(i));
                if spurdegree
                    spurs = [spurs linkingedges(i)];
                    len = [len edgelistlength(edgelist{linkingedges(i)})];
                end
            end

            linkingedges = [linkingedges n];

            [mn,i] = min(len);
            edge2delete = spurs(i);
            [spurdegree, spurnode] = connectioninfo(edge2delete);

            nodes2merge = find(A(spurnode,:));

            if length(nodes2merge) ~= 2
                error('attempt to merge other than 2 nodes');
            end

            removeedge(edge2delete);
            mergenodes(nodes2merge(1),nodes2merge(2))

        end

    % Look for spurs coming from 4-way junctions that are below the
    minimum length

```

```

        elseif spurdegree == 3
            removeedge(n); % Just remove it, no subsequent merging needed.
        end
    end
end

% Final cleanup of any new isolated edges that might have been created
by
% removing spurs. An edge is isolated if it has no connections to
other
% edges, or is only connected to itself (in a loop).

for n = 1:Nedges
    if ~isempty(edgelist{n}) && edgelistlength(edgelist{n}) < minlength
        if ~Econnections(n) || (Econnections(n)==1 && B(n,n) == 1)
            removeedge(n);
        end
    end
end

end % if minlength > 0

% Run through the edgelist and extract out the non-empty lists
m = 0;
for n = 1:Nedges
    if ~isempty(edgelist{n})
        m = m+1;
        nedgelist{m} = edgelist{n};
    end
end

%-----
% Internal function to merge 2 edgelists together at the specified nodes
and
% perform the necessary updates to the edge adjacency and node adjacency
% matrices and the connection count arrays

function mergenodes(n1,n2)

    edge1 = fix((n1+1)/2); % Indices of the edges associated with the
nodes
    edge2 = fix((n2+1)/2);

    % Get indices of nodes at each end of the two edges
    s1 = 2*edge1-1; e1 = 2*edge1;
    s2 = 2*edge2-1; e2 = 2*edge2;

    if edge1==edge2
        % We should not get here, but somehow we occasionally do
        % fprintf('Nodes %d %d\n',n1,n2) %% debug
        % warning('Attempt to merge an edge with itself')
        return
    end

    if ~A(n1,n2)
        error('Attempt to merge nodes that are not connected');
    end
end

```

```

if mod(n1,2) % node n1 is the start of edge1
flippedge1 = 1; % edge1 will need to be reversed in order to join edge2
else
flippedge1 = 0;
end

if mod(n2,2) % node n2 is the start of edge2
flippedge2 = 0;
else
flippedge2 = 1;
end

% Join edgelist together - with appropriate reordering depending on
which
% end is connected to which. The result is stored in edge1

if ~flippedge1 && ~flippedge2
edgelist{edge1} = [edgelist{edge1}; edgelist{edge2}];

A(e1,:) = A(e2,:); A(:,e1) = A(:,e2);
Nconnections(e1) = Nconnections(e2);

elseif ~flippedge1 && flippedge2
edgelist{edge1} = [edgelist{edge1}; flipud(edgelist{edge2})];

A(e1,:) = A(s2,:); A(:,e1) = A(:,s2);
Nconnections(e1) = Nconnections(s2);

elseif flippedge1 && ~flippedge2
edgelist{edge1} = [flipud(edgelist{edge1}); edgelist{edge2}];

A(s1,:) = A(e1,:); A(:,s1) = A(:,e1);
A(e1,:) = A(e2,:); A(:,e1) = A(:,e2);
Nconnections(s1) = Nconnections(e1);
Nconnections(e1) = Nconnections(e2);

elseif flippedge1 && flippedge2
edgelist{edge1} = [flipud(edgelist{edge1}); flipud(edgelist{edge2})];

A(s1,:) = A(e1,:); A(:,s1) = A(:,e1);
A(e1,:) = A(s2,:); A(:,e1) = A(:,s2);
Nconnections(s1) = Nconnections(e1);
Nconnections(e1) = Nconnections(s2);

else
fprintf('merging edges %d and %d\n',edge1, edge2); %%debug
error('We should not have got here - edgelist cannot be merged');
end

% Now correct the edge adjacency matrix to reflect the new arrangement
% The edges that the new edge1 is connected to is all the edges that
% edge1 and edge2 were connected to
B(edge1,:) = B(edge1,:) | B(edge2,:);
B(:,edge1) = B(:,edge1) | B(:,edge2);
B(edge1, edge1) = 0;

% Recompute connection counts because we have shuffled the adjacency
matrices
Econnections = sum(B);

```

```

Nconnections = sum(A);

removeedge(edge2); % Finally discard edge2

end % end of mergenodes

%-----

% Function to provide information about the connections at each end of an
% edgelist
%
% [spurdegree, spurnode, startnode, sconns, endnode, econns] =
connectioninfo(n)
%
% spurdegree - If this is non-zero it indicates this edgelist is a spur,
the
%               value is the number of edges this spur is connected to.
% spurnode    - If this is a spur spurnode is the index of the node that is
%               connected to other edges, 0 otherwise.
% startnode  - index of starting node of edgelist.
% endnode    - index of end node of edgelist.
% sconns     - number of connections to start node.
% econns     - number of connections to end node.

function [spurdegree, spurnode, startnode, sconns, endnode, econns] =
connectioninfo(n)

    if isempty(edgelist{n})
        spurdegree = 0; spurnode = 0;
        startnode = 0; sconns = 0; endnode = 0; econns = 0;
        return
    end

    startnode = 2*n-1;
    endnode   = 2*n;
    sconns = Nconnections(startnode); % No of connections to start node
    econns = Nconnections(endnode);  % No of connections to end node

    if sconns == 0 && econns >= 1
        spurdegree = econns;
        spurnode = endnode;
    elseif sconns >= 1 && econns == 0
        spurdegree = sconns;
        spurnode = startnode;
    else
        spurdegree = 0;
        spurnode = 0;
    end
end

%-----

% Function to remove an edgelist and perform the necessary updates to the
% edge
% adjacency and node adjacency matrices and the connection count arrays

function removeedge(n)

    edgelist{n} = [];

```

```

Econnections = Econnections - B(n,:);
Econnections(n) = 0;
B(n,:) = 0;
B(:,n) = 0;

nodes2delete = [2*n-1, 2*n];

Nconnections = Nconnections - A(nodes2delete(1),:);
Nconnections = Nconnections - A(nodes2delete(2),:);

A(nodes2delete, :) = 0;
A(:, nodes2delete) = 0;

end

%-----
% Function to compute the path length of an edgelist

function l = edgelistlength(edgelist)
    l = sum(sqrt(sum((edgelist(1:end-1,:)-edgelist(2:end,:)).^2, 2)));
end

%-----
end % End of cleanedgelists

```

ANEXO G ALGORITMO - FINDENDSJUNCTIONS

Descrição:	Utilizado pela função edgeline, encontra junções das bordas.
Fonte:	(KOVESI, 2007)
Sistema:	MATLAB

```

% Copyright (c) 2006 Peter Kovesi
% School of Computer Science & Software Engineering
% The University of Western Australia
% pk at csse uwa edu au
% http://www.csse.uwa.edu.au/
%
% Permission is hereby granted, free of charge, to any person obtaining a
copy
% of this software and associated documentation files (the "Software"), to
deal
% in the Software without restriction, subject to the following conditions:
%
% The above copyright notice and this permission notice shall be included
in
% all copies or substantial portions of the Software.
%
% The Software is provided "as is", without warranty of any kind.

% November 2006

function [rj, cj, re, ce] = findendsjunctions(edgeim, disp)

    if nargin == 1
        disp = 0;
    end

    % Ensure edge/line image really is thinned otherwise tests for
junctions
    % and endings may fail.
    b = bwmorph(edgeim, 'skel', Inf);

    % Set up look up table to find junctions. To do this we use the
function
    % defined at the end of this file to test that the centre pixel within
a 3x3
    % neighbourhood is a junction.
    lut = makelut(@junction, 3);
    junctions = applylut(b, lut);
    [rj,cj] = find(junctions);

    % Set up a look up table to find endings.
    lut = makelut(@ending, 3);
    ends = applylut(b, lut);
    [re,ce] = find(ends);

    if disp
        show(edgeim,1), hold on
        plot(cj,rj, 'r+')
        plot(ce,re, 'g+')

```

```

end

%-----
% Function to test whether the centre pixel within a 3x3 neighbourhood is a
% junction. The centre pixel must be set and the number of
% transitions/crossings
% between 0 and 1 as one traverses the perimeter of the 3x3 region must be
% 6 or
% 8.
%
% Pixels in the 3x3 region are numbered as follows
%
%      1 4 7
%      2 5 8
%      3 6 9

function b = junction(x)

    a = [x(1) x(2) x(3) x(6) x(9) x(8) x(7) x(4)];
    b = [x(2) x(3) x(6) x(9) x(8) x(7) x(4) x(1)];
    crossings = sum(abs(a-b));

    b = x(5) && crossings >= 6;

%-----
% Function to test whether the centre pixel within a 3x3 neighbourhood is
% an
% ending. The centre pixel must be set and the number of
% transitions/crossings
% between 0 and 1 as one traverses the perimeter of the 3x3 region must be
% 2.
%
% Pixels in the 3x3 region are numbered as follows
%
%      1 4 7
%      2 5 8
%      3 6 9

function b = ending(x)

    a = [x(1) x(2) x(3) x(6) x(9) x(8) x(7) x(4)];
    b = [x(2) x(3) x(6) x(9) x(8) x(7) x(4) x(1)];
    crossings = sum(abs(a-b));

    b = x(5) && crossings == 2;

```

ANEXO H ALGORITMO - IMAGE OVERLAY

Descrição:	Gera um overlay, máscara binária em uma imagem usando uma cor especificada
Fonte:	(EDDINS, 2007)
Sistema:	MATLAB

```

function out = imoverlay(in, mask, color)
%IMOVERLAY Create a mask-based image overlay.
%   OUT = IMOVERLAY(IN, MASK, COLOR) takes an input image, IN, and a binary
%   image, MASK, and produces an output image whose pixels in the MASK
%   locations have the specified COLOR.
%
%   IN should be a grayscale or an RGB image of class uint8, uint16, int16,
%   logical, double, or single.
%
%   MASK should be a two-dimensional logical matrix.
%
%   COLOR should be a 1-by-3 vector of values in the range [0, 1]. [0 0 0]
%   is black, and [1 1 1] is white.
%
%   OUT is a uint8 RGB image.
%
%   Steven L. Eddins, The MathWorks, Inc.
%   $Revision: 1.1 $   $Date: 2006/03/23 13:53:38 $

% If the user doesn't specify the color, use white.
DEFAULT_COLOR = [1 1 1];
if nargin < 3
    color = DEFAULT_COLOR;
end

% Make the uint8 the working data class. The output is also uint8.
in_uint8 = im2uint8(in);
color_uint8 = im2uint8(color);

% Initialize the red, green, and blue output channels.
if ndims(in_uint8) == 2
    % Input is grayscale. Initialize all output channels the same.
    out_red = in_uint8;
    out_green = in_uint8;
    out_blue = in_uint8;
else
    % Input is RGB truecolor.
    out_red = in_uint8(:,:,1);
    out_green = in_uint8(:,:,2);
    out_blue = in_uint8(:,:,3);
end

% Replace output channel values in the mask locations with the appropriate
% color value.
out_red(mask) = color_uint8(1);
out_green(mask) = color_uint8(2);
out_blue(mask) = color_uint8(3);

% Form an RGB truecolor image by concatenating the channel matrices along

```

```
% the third dimension.  
out = cat(3, out_red, out_green, out_blue);
```

ANEXO I ALGORITMO – PARA SELEÇÃO DA ÁREA

Descrição:	Identifica e etiqueta uma área na imagem, utiliza a função do Anexo G
Fonte:	Baseado em Cell segmentation de (EDDINS, 2007)
Sistema:	MATLAB

```
function retorno = SelecaoAreaImg(Imagem)
    I = Imagem;

    bw = im2bw(I, graythresh(I));
    bw2 = imfill(bw, 'holes');
    bw3 = imopen(bw2, ones(5,5));
    bw4 = bwareaopen(bw3, 40);
    bw4_perim = bwperim(bw4);

    mask_em = imextendedmax(I, 30);
    mask_em = imclose(mask_em, ones(5,5));
    mask_em = imfill(mask_em, 'holes');
    mask_em = bwareaopen(mask_em, 40);

    overlay = imoverlay(I, bw4_perim | mask_em, [255 255 255]);

    retorno = overlay;
```