

Universidade Federal do Amazonas  
Instituto de Ciências Exatas  
Departamento de Ciência da Computação  
Programa de Pós-Graduação em Informática

# **Classificação de Patologias em imagens médicas do Sangue.**

Luiz Wanderley Nagata Balduino

Manaus – Amazonas  
Outubro de 2006

Universidade Federal do Amazonas  
Departamento de Ciência da Computação

**Autor: Luiz Wanderley Nagata Balduino**  
**Orientador: Prof. Dr. Ayres Mardem Almeida do Nascimento**

**Dissertação de Mestrado** apresentada ao Programa de Pós-Graduação em Informática do Departamento de Ciência da Computação da UFAM, como parte dos requisitos necessários para a obtenção do título de Mestre em Informática. Área de concentração: **Engenharia da Computação**.

Banca Examinadora

Ayres Mardem Almeida do Nascimento, Dr. .... UFAM/PPGI - Orientador  
José Pinheiro Queiroz Neto, Dr. .... CEFET-AM - Membro  
José Luiz de Souza Pio, Dr. .... UFAM/PPGI - Membro  
José Raimundo Gomes Pereira, Dr. .... UFAM/PPGI - Membro

Manaus – Amazonas

Outubro de 2006

*Não temas, porque eu sou contigo; não te assombres, porque eu sou teu Deus; eu te fortaleço, e te ajudo, e te sustento com a destra da minha justiça.*

*Isaías 41:10*

*A vitória não pertence aos fortes, nem aos mais capacitados, mas a aqueles que decidiram perseverar nos seus sonhos e nunca desistiram.*

*Luiz Nagata*

*Ao ...*

Ao meu Deus que me deu a promessa e a cumpriu, nunca me abandonou nestes dias de lutas e guerras pelas quais passei neste mestrado. Louvo a Deus por ter me dado mais esta vitória. Obrigado Jesus.

# Agradecimentos

Ao Meu Deus por mais esta vitória, por me mostrar a riqueza da sua bondade, a força do seu eterno amor e pela grandiosidade da sua graça.

A minha família, e em especial a minha mãe, por todo o apoio a mim dedicado.

Ao meu orientador Ayres Mardem por acreditar em mim e ter me ajudado neste mestrado.

A Virginia, Clésio e Paula amigos e irmãos em Cristo que me incentivaram neste mestrado.

Ao grupo Crescendo com Cristo pelo pelo empenho destes amados irmãos em pregar a palavra de Deus dentro da UFAM, e em especial a irmão Antonia, uma amigona e irmã em Cristo.

Ao Arelian que me ajudou muito no início do mestrado.

Ao Marco Sposito por me ajudar na correção dos erros gramaticais da minha dissertação.

A minha amiga Úrsula, companheira de lutas e guerras que passamos juntos neste mestrado, sucesso a você amiguinha.

A minha amiga Desirré, que se tornou uma amiga muito especial neste mestrado, no meio de tanta turbulência que passamos também neste mestrado.

A Elienai e todo o pessoal da secretária pelo apoio.

Ao Meredith e o pessoal do CPD que me ajudarem neste mestrado.

Aos Professores José Raimundo e José Pio pela dicas e conselhos preciosos para mim

neste mestrado.

A prof. e Dr.a Virginia Brilhante pela sua ajuda em IA.

Ao Prof. Edleno pela paciência comigo em PAA e no mestrado.

A todos os meus irmão de células e pessoas que intercederam em oração por mim  
diante de Deus.

À FAPEAM, pelo apoio financeiro.

# Resumo

No presente trabalho mostra-se a proposta e o desenvolvimento de extração de imagens, bem como algumas melhorias propostas em outras técnicas de segmentação e classificação, com objetivo de reconhecer hematopatologias em imagens médicas do sangue. A identificação automática dessas doenças viabiliza o diagnóstico mais preciso e com maior rapidez. A abordagem do presente trabalho segue as etapas de segmentação das imagens, extração das características visuais e a classificação da patologia.

Na etapa de segmentação é aplicada a técnica de separação por cor para obtenção das bordas de hemácias, dos leucócitos e das plaquetas. A extração é a etapa onde as células são separadas da imagem, esta extração é realizada através de procedimentos propostos e desenvolvidos para este objetivo. Por fim, a classificação é executada por meio de similaridades entre as imagens dos leucócitos, usando técnicas estatísticas entre spatiograms.

Os resultados mostraram-se promissores quanto a viabilidade da aplicação da metodologia em sistemas de apoio ao diagnóstico por imagens, devido a simplicidade da aplicação das mesmas bem como dos resultados obtidos.

Manaus – Amazonas

Maior de 2006

Abstract of Thesis presented to PPGI/UFAM as a partial fulfillment of the requirements for the degree of Master of Informatics (M.Sc.)

## **Pathology Classification on Medical Images of the Blood**

In the present work it is shown the proposal and the development of extraction of images, as well as some improvements proposed in other segmentation techniques and classification, with objective of recognizing pathology in medical images of the blood. The automatic identification of those diseases makes possible the most necessary diagnosis and with larger speed. The approach of the present work follows the stages of segmentation of the images, extraction of the visual characteristics and the classification of the pathology.

In the segmentation stage the separation technique is applied by color for obtaining of the red blood cells borders, of the leukocytes and of the platelets. The extraction is the stage where the cells are separate from the image, this extraction is accomplished through proposed procedures and developed for this aim at. Finally, the classification is executed through similarities among the images of the leucócitos, using statistical techniques among spatiograms.

The results were shown promising as the viability of the application of the methodology in support systems to the diagnosis for images, due to simplicity of the application of the same ones as well as of the obtained results.

**Luiz Wanderley Nagata Balduino**

May/2006

Advisors: Ayres Mardem Almeida do Nascimento

Departments: Informatics

# Sumário

<b>Resumo</b>	<b>vii</b>
<b>Abstract</b>	<b>viii</b>
<b>Lista de Figuras</b>	<b>xii</b>
<b>Lista de Tabelas</b>	<b>xviii</b>
<b>Lista de Acrônimos</b>	<b>xix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Considerações Iniciais . . . . .	1
1.2 Motivação e Objetivos . . . . .	2
1.3 Estrutura do Texto . . . . .	5
<b>2 Fundamentações</b>	<b>7</b>
2.1 Segmentação . . . . .	7
2.1.1 Canny . . . . .	8
2.1.2 Segmentação de Imagens Coloridas . . . . .	11
2.1.3 Comentários sobre as técnicas de Segmentação . . . . .	15
2.2 Extração de imagens . . . . .	17
2.3 Classificação de Imagens . . . . .	18

---

2.3.1	Spatioqram . . . . .	19
2.3.2	Similaridade entre spatiograms . . . . .	20
2.4	Conclusão . . . . .	22
<b>3</b>	<b>Metodologias</b>	<b>24</b>
3.1	Segmentação . . . . .	25
3.1.1	Obtenção das Cores que compõe as Classes . . . . .	28
3.1.2	Estudo das distancias de Cores . . . . .	30
3.1.3	Mapa de bits . . . . .	34
3.1.4	Comentários . . . . .	36
3.2	Extração de Imagens . . . . .	37
3.2.1	Chain Code ou Código da Cadeia . . . . .	37
3.2.2	Técnica de Extração de imagens . . . . .	37
3.3	Classificação . . . . .	82
3.4	Conclusão . . . . .	87
<b>4</b>	<b>Resultados</b>	<b>88</b>
4.1	Segmentação . . . . .	88
4.1.1	Oversegmentation . . . . .	88
4.1.2	Marcação das classes de cores de Pixels. . . . .	89
4.1.3	Cores Armazenadas . . . . .	92
4.1.4	Estudo da distância de cores . . . . .	94
4.1.5	Mapa de Bits das regiões de Classes de cores . . . . .	100
4.1.6	Mapa de bits das Bordas das células . . . . .	102
4.2	Extração . . . . .	104
4.3	Classificação de Patologias . . . . .	106
4.4	Comentários . . . . .	114

---

<b>5</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>115</b>
5.1	Conclusão da Segmentação . . . . .	115
5.2	Conclusão da Extração . . . . .	119
5.3	Conclusão da Classificação das Patologias . . . . .	119
5.4	Conclusões Finais . . . . .	120
5.5	Trabalhos Futuros . . . . .	121
	<b>Referências Bibliográficas</b>	<b>122</b>

# Lista de Figuras

1.1	Esquema de funcionamento do SADAH. . . . .	3
1.2	Diagrama de Contexto do SADAH, pode ser visto na figura a interação do sistema com o paciente e com o médico. . . . .	4
1.3	Diagrama de Fluxo de dados da Classificação de Patologias. . . . .	5
2.1	Um exemplo da aplicação da supressão não máxima. . . . .	10
2.2	(a) imagem original (b) imagem resultante de Canny. . . . .	11
2.3	(a) Recorte do Leucócito da Figura 2.2, (b) figura resultante de Canny do recorte da Figura da letra(a). . . . .	11
2.4	Valores utilizados na segmentação de acordo com a classe escolhida. .	14
2.5	Imagem resultante da técnica de Segmentação Colorida. . . . .	15
2.6	(a) imagem resultante de Canny, (b) imagem resultante da segmentação colorida . . . . .	16
2.7	Recorte do Leucócito da Figura 2(a),(b) o mesmo recorte passado pela técnica de Canny, (c) o mesmo recorte passado pela técnica Comaniciu. 16	
2.8	(a) Imagem original (b) imagem com o efeito de borramento . . . . .	17
2.9	“chain codes” ou código da cadeia. . . . .	17
3.1	Passos da classificação. . . . .	24
3.2	Modelo de cor RGB. . . . .	26
3.3	Mapa de Bits. . . . .	27

---

3.4	Estrutura das tabelas de cores de Fundo e de Leucócitos e Plaquetas.	29
3.5	Diagonal do Cubo Tridimensional de cor no formato RGB. . . . .	31
3.6	Exemplo da criação do mapa de bits com as bordas das células. . . .	36
3.7	(a) seqüência de Gonzalez[1], (b) seqüência utilizada neste trabalho. .	37
3.8	Circulo adotado. . . . .	38
3.9	Bordas com espessuras “1” e “2” pixels . . . . .	39
3.10	Código dos Vizinhos. . . . .	39
3.11	Caminhamento pela borda. . . . .	41
3.12	Exemplo de correção da borda. . . . .	44
3.13	Configuração utilizada para identificação do pixel interno da borda. .	45
3.14	Exemplo da identificação do pixel interno da borda. . . . .	46
3.15	Tiras da imagem. . . . .	47
3.16	Vetor de tiras. . . . .	48
3.17	Exemplo do que os nós ou tiras armazenam. . . . .	48
3.18	Exemplo do preenchimento das variáveis $y_i$ e $y_f$ . . . . .	49
3.19	Exemplo do funcionamento do vetor de tiras. . . . .	50
3.20	Configuração dos pixels normais. . . . .	51
3.21	Configuração dos pixels de Linha. . . . .	51
3.22	Exemplo da identificação da linha continua na borda. . . . .	52
3.23	Exemplo de borda no 1° Quadrante. . . . .	53
3.24	Códigos do vizinho do Pixel Brotoeja. . . . .	53
3.25	Exemplo da inserção de pixels com códigos 12 e 16 nos quadrantes. .	54
3.26	Exemplo de Formas geométricas fechadas. . . . .	55
3.27	Identificação do Primeiro quadrante. . . . .	55
3.28	Exemplo de inserção de pixels normais no primeiro quadrante. . . . .	57
3.29	Exemplo de inserção do pixel de linha. . . . .	59
3.30	Exemplo da finalização da tira S2. . . . .	60

---

3.31	Exemplo de inserção do pixel de linha com código 11. . . . .	61
3.32	Exemplo de inserção do pixel de linha com código final 10. . . . .	62
3.33	Exemplo de inserção do pixel brotoeja. . . . .	62
3.34	Identificação do 2º quadrante (a) e do 3º quadrante (b). . . . .	63
3.35	Exemplo da inserção dos pixels normais no 2º e no 3º quadrantes. . .	64
3.36	Exemplo de inserção de linha que começa com o código 10 e finaliza com 11. . . . .	65
3.37	Exemplo de inserção de linha que começa com o código 10 e finaliza com 13. . . . .	66
3.38	Exemplo de inserção da linha continua que começa com o código 5 e termina com 11. . . . .	67
3.39	Exemplo de inserção da linha continua que começa com o código 5 e termina com 13. . . . .	68
3.40	Exemplo de inserção do pixel brotoeja no inicio da lista de tiras. . . .	69
3.41	Exemplo de inserção do pixel brotoeja no final da lista de tiras. . . .	69
3.42	Identificação do 4º quadrante. . . . .	70
3.43	Exemplo da inserção de tiras no quarto quadrante. . . . .	71
3.44	Exemplo da inserção do pixel inicial da linha no quarto quadrante. . .	74
3.45	Exemplo da inserção do pixel inicial da linha no quarto quadrante com código 10. . . . .	75
3.46	Exemplo da inserção do pixel inicial da linha no quarto quadrante com o código 11. . . . .	76
3.47	Exemplo da inserção do pixel inicial da linha no quarto quadrante que termina com código 10. . . . .	77
3.48	Exemplo de inserção do pixel brotoeja no quarto quadrante. . . . .	77
3.49	Imagem base para o cálculo das novas posições x e y. . . . .	78
3.50	Vetor do Histograma. . . . .	79

---

3.51	Exemplo do funcionamento do vetor do histograma. . . . .	80
3.52	Exemplo do Vetor que armazena o spatiogram. . . . .	81
3.53	Leucócitos selecionados para os testes de similaridade. . . . .	82
3.54	Graus de similaridades do Leucócito 21 da tabela Resultado2. . . . .	85
3.55	Consulta com os resultados dos cálculos de similaridade do Leucócito 250, ordenados pela coluna (JM). . . . .	87
4.1	Imagem de sangue normal. . . . .	89
4.2	Imagem resultante da oversegmentation da Figura 4.1. . . . .	89
4.3	Imagem original de sangue que apresenta uma Leucemia Aguda. . . . .	90
4.4	Imagem resultante da oversegmentation da Figura 4.3. . . . .	90
4.5	Imagem do sangue que apresenta a patologia de Desordem Limfopro- liferativo. . . . .	91
4.6	Imagem resultante da Oversegmentation da Figura 4.5. . . . .	91
4.7	Imagem da Figura 4.2 com todas as células marcadas de preto. . . . .	92
4.8	Imagem da Figura 4.2 com os pixels de fundo e de hemácias pintadas de preto. . . . .	92
4.9	Estruturas das tabelas, (a) estrutura da tabela da cor de fundo e (b) da de Leucócitos e Plaquetas. . . . .	93
4.10	Dados das tabelas de cores de Fundo e de Leucócitos e Plaquetas. . . . .	94
4.11	Imagem da Figura 4.4 com os pixels de fundo marcados pela distancia 4. . . . .	95
4.12	Imagem da Figura 4.4 com os pixels de fundo marcados pela distancia 5. . . . .	95
4.13	Imagem da Figura 4.4 com os pixels de fundo marcados pela distancia 10. . . . .	96

---

4.14 Imagem da Figura 4.6 com os pixels de fundo marcados pela distancia	
4. . . . .	96
4.15 Imagem da Figura 4.6 com os pixels de fundo marcados pela distancia	
5. . . . .	97
4.16 Imagem da Figura 4.6 com os pixels de fundo marcados pela distancia	
10. . . . .	97
4.17 Exemplos de imagens resultantes da marcação utilizando as distâncias selecionadas 4 (a), 5(b) e 10(c). . . . .	98
4.18 Imagem da Figura 4.6 com os pixels de Leucócitos e Plaquetas mar- cados pela distância euclidiana 4. . . . .	100
4.19 Imagem da Figura 4.6 com os pixels de Leucócitos e Plaquetas mar- cados pela distância euclidiana 5. . . . .	100
4.20 Imagem da Figura 4.6 com os pixels de Leucócitos e Plaquetas mar- cados pela distância euclidiana 10. . . . .	101
4.21 Imagens de Leucócitos e Plaquetas da Figura 4.1 . . . . .	102
4.22 Imagem com as hemácias da Figura 4.1 . . . . .	102
4.23 Imagem com os Leucócitos e Plaquetas da Figura 4.3 . . . . .	103
4.24 Imagem com as hemácias da Figura 4.3 . . . . .	103
4.25 Imagem com os Leucócitos e Plaquetas da Figura 4.5 . . . . .	104
4.26 Imagem com as hemácias da Figura 4.5 . . . . .	104
4.27 Imagem com as bordas das células da Figura 4.1 . . . . .	105
4.28 Imagem com as bordas das células da figura 4.3 . . . . .	105
4.29 Imagem com as bordas das células da figura 4.6 . . . . .	106
4.30 Imagem com as hemácias das células da Figura 4.1 . . . . .	107
4.31 Imagem com os Leucócitos e Plaquetas da Figura 4.1 . . . . .	108
4.32 Imagem que apresenta as hemácias da Figura 4.3 . . . . .	108
4.33 Imagem que apresenta os Leucócitos e Plaquetas da Figura 4.3 . . . .	108

---

4.34	Imagem que apresenta as hemácias da Figura 4.5 . . . . .	109
4.35	Imagem que apresenta os Leucócitos e Plaquetas da Figura 4.5 . . . . .	109
4.36	Consulta dos Resultados dos cálculos de similaridade feitos para o Leucócito 230, na coluna 'Celula.ID' da primeira linha. . . . .	110
4.37	Consulta dos Resultados dos cálculos de similaridade feitos para o Leucócito 21, na coluna 'Celula.ID' da primeira linha. . . . .	110
4.38	Tabela com o resultado da contagem. . . . .	111
4.39	Exemplo em se encontra os 4 Leucócitos da mesma patologia nas 4 linhas seguintes da primeira linha. . . . .	111
4.40	Exemplo da Consulta ordenada pelos resultados de Bhattacharyya. . . . .	112
4.41	Tabela com códigos dos Leucócitos que apareciam três linhas seguin- tes após a primeira linha da consulta. . . . .	113
5.1	Imagem padrão utilizada neste trabalho. . . . .	116
5.2	Bordas resultantes da técnica Canny a partir da Figura 1. . . . .	116
5.3	Bordas resultantes da segmentação de imagens coloridas geradas a partir da Figura 1. . . . .	117
5.4	Bordas geradas neste trabalho a partir da Figura 1. . . . .	118

# Lista de Tabelas

3.1	Classificação de Patologias do Sangue . . . . .	29
3.2	Resultados das distancias selecionadas aleatoriamente para as cores de fundo. . . . .	31
3.3	Tabela de associação das células da Figura 3.53 com as patologias. . .	83
3.4	Tabela com os códigos dados aos Leucócitos da Figura 50. . . . .	84

# Lista de Acrônimos

CBIR: *Content-based Image Retrieval*

MVE: *minimum volume ellipsoid*

MER: *Modelo de Entidade Relacionamento*

PACS: *Picture Archiving and Communication Systems*

# Capítulo 1

## Introdução

### 1.1 Considerações Iniciais

Este trabalho aborda os problemas da segmentação, extração e classificação automática de patologias do sangue por meio de imagens digitais. Estes problemas estão inseridos em que sistemas realizam o auxílio ao diagnóstico automático por imagem. Suas melhorias, poderão contribuir de maneira significativa para a diminuição do tempo de resposta dos exames bem como o aumento da precisão dos resultados dos mesmos. A maioria dos diagnósticos ainda são atualmente realizados manualmente, requerendo a presença de um especialista e uma grande quantidade de amostras com patologias previamente identificadas. Com a utilização de imagens em um sistema de apoio ao diagnóstico, a identificação de padrões de forma, cor ou textura pode ser obtida mais rapidamente, facilitando o aprofundamento no descobrimento de novas hemopatologias. Por outro lado, a precisão dos resultados se torna maior. A ciência médica tem se beneficiado muito com a evolução da computação. Essa contribuição tem sido significativa em várias frentes, tais como o controle de informações de pacientes, pesquisa de medicina à distância, simulação de procedimentos médicos e mais recentemente no diagnóstico por imagem.

No diagnóstico por imagens, obtêm-se imagens por vários meios, tais como ressonância magnética, tomografia computadorizada, ecocardiograma, ultra-sonografia e outros. As imagens podem ser aplicadas a vários exames, como o de densitometria óssea, do coração, da mama, do sangue entre outros.

Nas próximas seções serão mostradas as motivações que levaram ao desenvolvimento deste trabalho, a descrição dos seus objetivos bem como a estrutura do texto.

## 1.2 Motivação e Objetivos

Ainda hoje, em vários hospitais, os especialistas ainda realizam o exame da patologia do sangue utilizando o microscópio. Essa forma de obtenção de diagnóstico apresenta alguns problemas, tais como: a conservação das amostras de sangue em ambientes e condições apropriadas; o controle destas amostras; o tempo que o especialista precisa dedicar para o estudo; e a identificação e obtenção do diagnóstico preciso da patologia, o que leva a uma certa demora nos exames. Esses problemas, podem ser consideravelmente reduzidos com o uso de um sistema que classifique as patologias por meio de imagens do sangue. A utilização de imagens juntamente com o gerenciamento das informações dos pacientes, dará maior agilidade e eficiência aos resultados dos exames e fornecerá um apoio ao diagnóstico médico. Portanto, o objetivo geral deste trabalho é propor melhorias nas técnicas existentes que irão auxiliar o desenvolvimento de parte do sistema de auxílio automático ao diagnóstico de patologias do sangue. Como objetivos específicos, destacam-se:

- O desenvolvimento do algoritmo para segmentação através da melhoria da técnica por Comaniciu [2].

- O desenvolvimento de um algoritmo de extração proposto com baseado no chain codes [1].

- O desenvolvimento do algoritmo de classificação de imagens através da melhoria da técnica de Spatiogram [3] utilizando as das técnicas de Bhattacharyya [4], de Mahalanobis [5], de Jeffries-Matusita [6] para obter resultados mais no calculo de similaridade entre imagens de leucócitos.

As técnicas de processamento de imagens têm sido de fundamental importância no apoio ao diagnóstico por imagem, mais precisamente nos exames obtidos por novos meios de obtenção de imagens médicas tais como a radiografia, ressonância magnética, tomografia computadorizada, ultrassonografia, entre outros. Isso tem propiciado a criação de sistemas que auxiliam o diagnóstico médico em muitos hospitais. Um Sistema de Apoio ao Diagnóstico Automático de Hemopatologias (SADAH) seria útil não somente para dar apoio ao diagnóstico por imagem, mas também para gerenciar os dados dos pacientes, auxiliar nas pesquisas médicas e disponibilizar consultas pela internet. Uma idéia do funcionamento deste sistema, é mostrado na Figura 1.1.

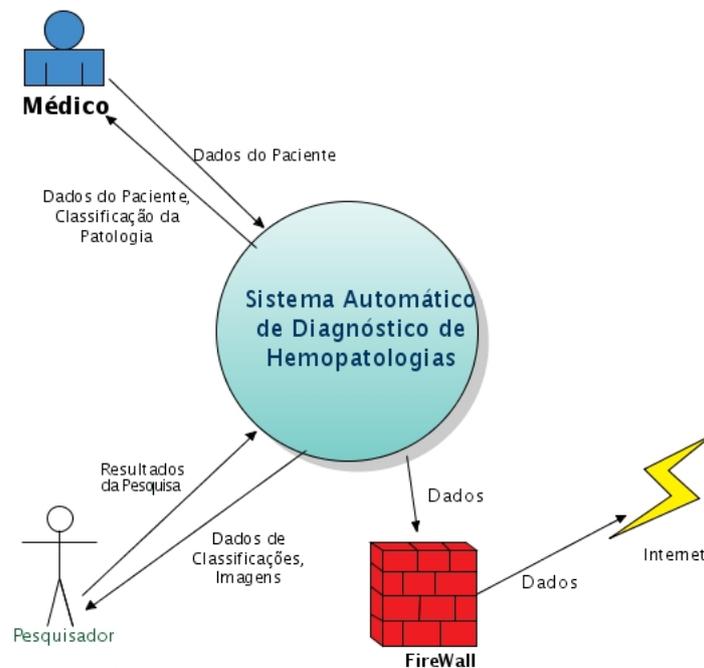


Fig. 1.1: Esquema de funcionamento do SADAH.

Na Figura 1.1 pode ser observada a interfaceamento do SADAH com o médico, com o pesquisador e o a disponibilização dos dados na internet. Internamente ao SADAH encontra-se um SGBD com todas as informações necessárias ao apoio ao diagnóstico.

Dentre os vários módulos existentes no SADAH, pode-se citar: o cadastro de pacientes, Consulta de dados, SGBD, Cadastro de Diagnostico e a classificação de imagens. A representação destes módulos no diagrama de contexto é apresentada na Figura 2. O módulo desenvolvido neste trabalho é o de classificação de imagens que aparece no canto superior direito da Figura 1.2. Os outros módulos estão fora do escopo deste trabalho.

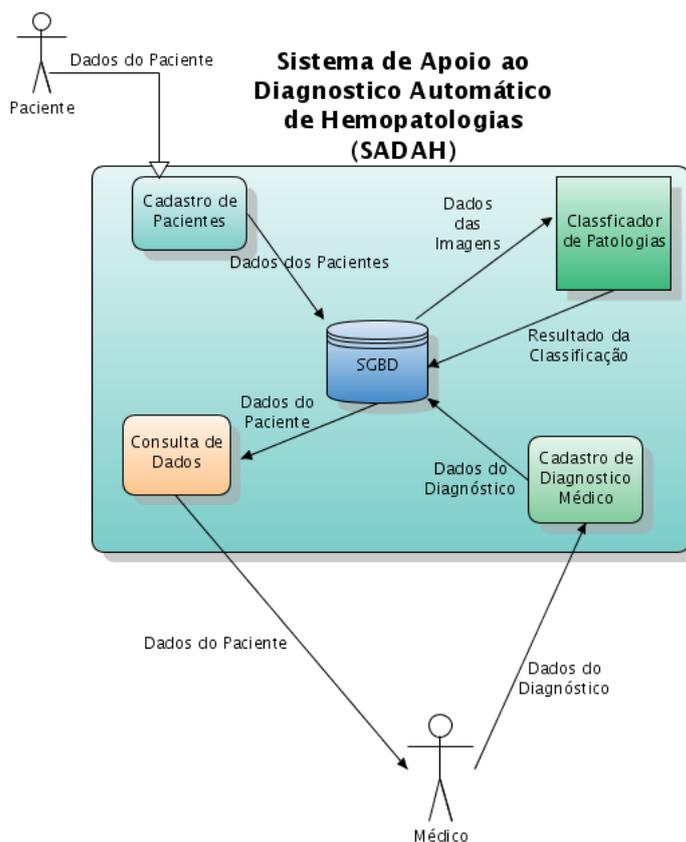


Fig. 1.2: Diagrama de Contexto do SADAH, pode ser visto na figura a interação do sistema com o paciente e com o médico.

O detalhamento do módulo de classificação de patologias é mostrado na Figura

1.3, onde pode ser observado os submódulos de segmentação, de extração e classificação das imagens do sangue.

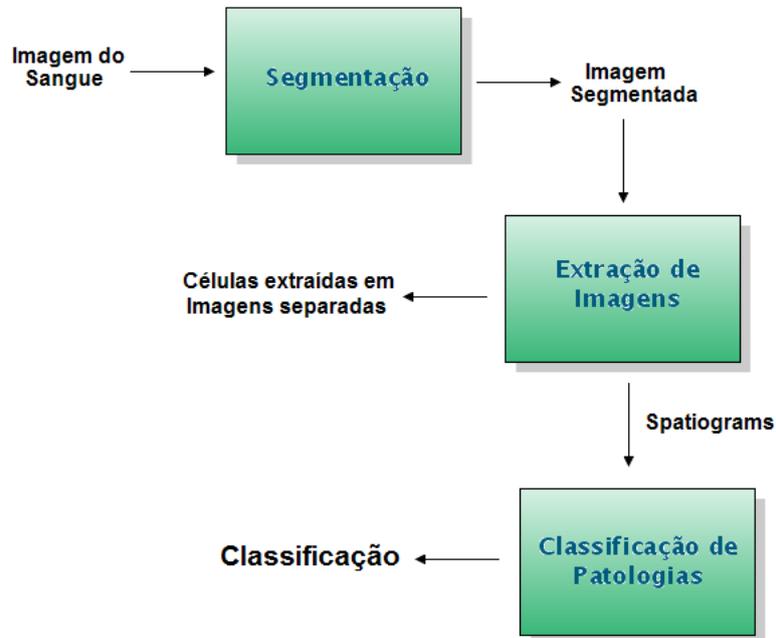


Fig. 1.3: Diagrama de Fluxo de dados da Classificação de Patologias.

O submódulo de Segmentação apresentado na Figura 1.3, cria uma imagem com as bordas das células a partir da imagem de entrada. O submódulo de Extração de Imagens extrai o spatioGRAM de cada célula gerando uma nova imagem para cada célula. O submódulo de Classificação de Patologias realiza a classificação a partir dos spatioGRAMs gerados pelo submódulo anterior.

## 1.3 Estrutura do Texto

A presente dissertação está organizada da seguinte forma:

No Capítulo 1 encontra-se a descrição geral do trabalho, bem como a motivação e os objetivos que foram seguidos.

No Capítulo 2 são apresentadas as fundamentações utilizadas para a realização do presente trabalho.

O Capítulo 3 é apresentada a metodologia que foi proposta e desenvolvida, bem como a melhoria de outros trabalhos.

O Capítulo 4 mostra os resultados obtidos no capítulo anterior.

No Capítulo 5 são apresentadas as conclusões dos resultados apresentados neste trabalho e análise dos resultados obtidos.

Por fim são mostradas as bibliografias utilizadas nesta dissertação.

# Capítulo 2

## Fundamentações

Neste capítulo serão apresentados os resumos de algumas técnicas pesquisadas, bem como as motivações para suas escolhas para a realização deste trabalho. Este capítulo está dividido em três partes: (i) Segmentação, (ii) Extração e (iii) Classificação.

### 2.1 Segmentação

Segundo Gonzalez [1], segmentação é a subdivisão da imagem em suas partes ou objetos constituintes, ou seja, é uma forma de encontrar as bordas dos objetos contidos na mesma, de maneira que eles possam ser extraídos e trabalhados individualmente.

As técnicas de segmentação descritas a seguir são algoritmos de detecção de bordas dos objetos da imagem.

Bordas são os limites que cada objeto apresenta em uma imagem, e estes limites, dependendo dos objetos da imagem, podem apresentar variações abruptas quando calculado o gradiente da imagem.

### 2.1.1 Canny

Criado em 1986 por John F. Canny [7], é considerado um algoritmo ótimo [8][9][10] para a detecção de bordas, baseado em um filtro de convolução gaussiana que suaviza os ruídos e localiza as bordas, além de ser considerado atualmente um algoritmo clássico.

Esta técnica trabalha somente com imagens em nível de cinza, mas pode ser aplicada a imagens coloridas depois que forem transformadas para imagens em nível de cinza.

Baseia-se nas seguintes regras:

- 1. Boa Detecção:** Localizar e marcar realmente as todas as bordas verdadeiramente existentes, isto é feito com base na maximização da razão sinal/ruído, pois as probabilidades de se encontrar bordas verdadeiras aumentam quando esta razão é grande;
- 2. Boa Localização:** A distância entre a borda detectada e a borda verdadeira deve ser mínima;
- 3. Boa Resposta:** Para cada borda deve haver somente uma resposta.

A técnica de Canny segue estes passos:

1. A imagem é suavizada pela convolução de um filtro gaussiano nas direções x e y, gerando duas novas imagens ( $I_x, I_y$ ), utilizando a equação (2.1):

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-x^2}{2\sigma^2}\right). \quad (2.1)$$

em que  $\sigma$  é o valor da variância de Gauss fornecido a função.

2. Convoluciona-se cada imagem gerada no passo anterior com um filtro que é uma derivada de 1ª ordem da função gaussiana utilizada no passo 1, em que  $I_x$  é convolucionado na direção  $y$  e  $I_y$  na direção  $x$  gerando  $I'_x$  e  $I'_y$ . A formula utilizada para a esta nova convolução esta mostrada na equação 2.2.

$$G'(x) = \frac{-x}{\sqrt{2\pi\sigma^4}} \exp\left(\frac{-x^2}{2\sigma^2}\right). \quad (2.2)$$

3. Calcula-se a magnitude e a orientação do gradiente da imagem. Isto pode ser feito com o uso das equações 2.3 e 2.3.

Magnitude:

$$M = \sqrt{\|I'_x\|^2 + \|I'_y\|^2}. \quad (2.3)$$

Orientação do Gradiente:

$$\theta = \arctan(I'_x, I'_y). \quad (2.4)$$

4. Aplica-se a supressão não máxima, que é uma técnica de afinamento de bordas. Isto é feito com base na vizinhança de 8 onde ao ser encontrado um pixel que tenha magnitude máxima, é verificado se os dois pixels vizinhos que estão na direção do gradiente, possuem magnitude máxima e se os pixels que se encontram na direção perpendicular do gradiente possuem magnitude zero . Se for verdade mantém-se o pixel, caso contrário zera-se o mesmo. A Figura 2.1 mostra um exemplo do que foi exposto.

Como pode ser observado na letra (a), existe uma borda que apresenta espessura a ser afinada, cujos pixels marcados apresentam magnitude máxima (os pixels não marcados possuem magnitude zero). A letra (b) mostra um pixel no centro da mascara 3x3 que possui gradiente máximo, em que a seta que aponta para o pixel superior esquerdo, indica a direção do gradiente. A outra seta indica a direção perpendicular ao gradiente. Pode ser verificado ainda que os pixels que estão na mesma

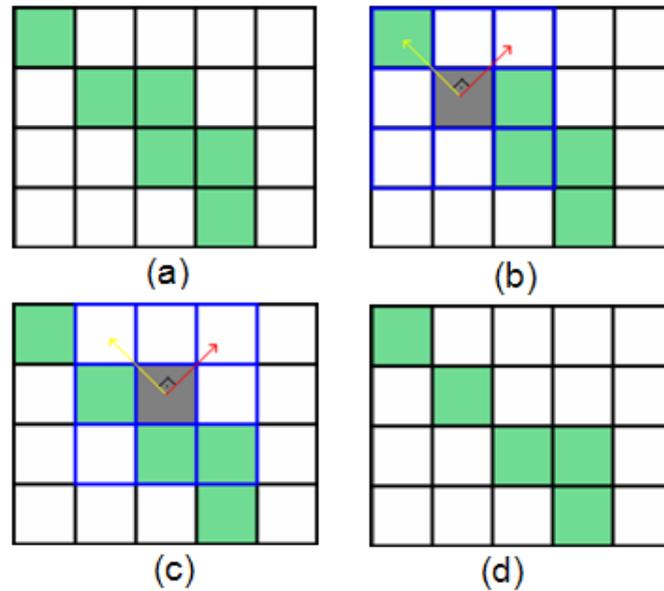


Fig. 2.1: Um exemplo da aplicação da supressão não máxima.

direção do gradiente, possuem magnitude máxima e os pixel que estão na direção perpendicular possuem magnitude zero, indicando que o pixel central é um pixel de borda, portanto será mantido. Na letra (c) pode ser verificado que os dois pixels na direção do gradiente não são máximos, portanto o pixel central não obedece a regra da supressão não máxima, sendo o mesmo descartado, como pode ser observado na letra (d).

5. Por último aplica-se uma técnica de limiarização por histerese para eliminar as bordas falsas e sinais espúrios, criando uma imagem binária. Esta técnica é realizada utilizando-se dois limiares  $\tau_1$  e  $\tau_2$  em que  $\tau_1 \cong 2\tau_2$ . Os valores acima de  $\tau_1$  e abaixo de  $\tau_2$  são marcados como 1 e os valor entre  $\tau_1$  e  $\tau_2$  são marcados como 0.

O resultado deste algoritmo pode ser visualizados na Figura 2.2(b), obtida pela técnica de Canny. A Figura 2.2(a) é a imagem original.

A técnica de Canny realizou a detecção de borda para a segmentação das células de sangue, mas apresentou o problema de junção de células diferentes em um mesmo

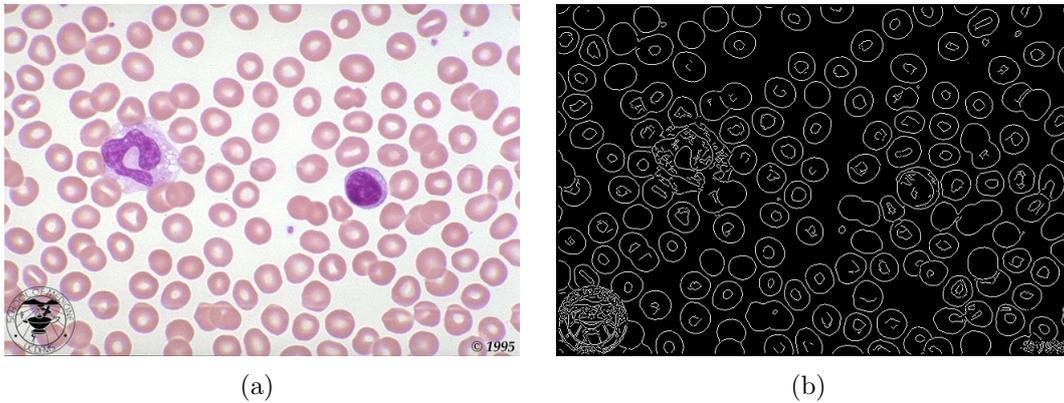


Fig. 2.2: (a) imagem original (b) imagem resultante de Canny.

segmento como pode ser observado na Figura 2.3. Este problema será abordado com maiores detalhes mais a frente.

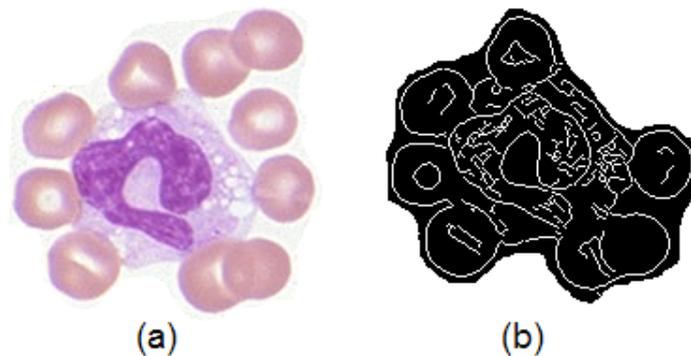


Fig. 2.3: (a) Recorte do Leucócito da Figura 2.2, (b) figura resultante de Canny do recorte da Figura da letra(a).

### 2.1.2 Segmentação de Imagens Coloridas

A técnica de Segmentação de imagens Coloridas proposta por Comaniciu [2] é baseado na técnica do deslocamento médio através de um procedimento simples que não envolve a estimação dos parâmetros estatísticos para a determinação da densidade do gradiente. Esta técnica trabalha diretamente com imagens coloridas, sem precisar converter-las para imagens em tons de cinza.

A seguir, serão apresentados os resumos das técnicas do deslocamento médio e do algoritmo utilizado na técnica de segmentação de imagens coloridas.

### Técnica do Deslocamento Médio

A técnica do deslocamento médio foi proposta por Fukunaga [11] e generalizada por Cheng [12]. Esta técnica busca a maior densidade do gradiente. Segundo Comaniciu [2], esta técnica é uma ferramenta extremamente versátil para a análise do espaço de características e fornece soluções confiáveis para diversas tarefas em visão computacional. Segundo Martinkauppi [13], a técnica do deslocamento médio é baseada na seguinte na equação:

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right). \quad (2.5)$$

Onde:

$x$  : é um ponto;

$n$  : é o número de pontos;

$K$  : é o kernel de Epanechnikov;

$d$  : é dimensão do espaço;

$h$  : é tamanho da janela.

O kernel de Epanechnikov é da forma:

$$K_E(x) = \begin{cases} \frac{1}{2}C_d^{-1} (d + 2) (1 - \|x\|^2), & \text{se } \|x\| < 1 \\ 0, & \text{caso contrário} \end{cases} \quad (2.6)$$

Onde:  $C_d$  é o volume da esfera d-dimensional.

A seguir é apresentado um resumo do algoritmo desta técnica segundo Comaniciu [2].

1. Escolhe-se o raio  $\mathbf{r}$  da janela da busca;

2. Escolhe-se a posição inicial da janela;
3. Computa-se o vetor médio do deslocamento e desloca-se a janela de busca por essa quantidade;
4. Repete-se os passos de 1 a 3 até convergir.

### Algoritmo para a Segmentação de Imagens Coloridas

Será visto agora um resumo deste algoritmo.

As imagens resultantes da técnica de segmentação de imagens coloridas, apresentam regiões de cores homogêneas. O número destas regiões aparecem em uma escala continua de acordo com a escolha de um dos métodos apresentados por Comaniciu [2] e descritos a seguir.

a) *Undersegmentation*: é uma segmentação que trabalha com as cores mais significantes da imagem, tem uma larga margem de tolerância para definir a homogeneidade e corresponde a baixa resolução.

b) *Oversegmentation*: é uma segmentação melhor que a anterior. É indicada para o reconhecimento de objetos e corresponde a resolução intermediária. É com esta parte que iremos trabalhar nesta dissertação.

c) *Quatization*: é a segmentação que mais se aproxima da imagem original, pois a imagem resultante apresenta características visuais muito próximas a da original. considera-se que todas as cores da imagem são importantes. É indicada para consulta baseada em conteúdo.

A seguir é apresentado o resumo do algoritmo de Segmentação de Imagens Coloridas.

1. *Definição dos parâmetros de segmentação*: é escolhida uma das três classes de segmentação e esta escolha implica na utilização dos parâmetros de tamanho da janela ( $r$ ), do número de cores significantes ( $N_{min}$ ) e do menor número de pixels

contínuo ( $N_{con}$ ) que estão na Tabela da Figura 2.4. O valor de ( $r$ ) é proporcional ao valor global da matriz de variância.

Classes de Segmentação	Parâmetros		
	$r$	$N_{min}$	$N_{con}$
Undersegmentation	$0.4\sigma$	400	10
Oversegmentation	$0.3\sigma$	100	10
Quatization	$0.2\sigma$	50	0

Fig. 2.4: Valores utilizados na segmentação de acordo com a classe escolhida.

2. *Definição da janela de Busca:* São escolhidos aleatoriamente 25 pixels candidatos, que são considerados como a posição inicial da janela de busca onde, para cada pixel é computada a média da vizinhança de 8 e são mapeados no vetor de características. Destes pixels candidatos, são selecionados os que possuem pixels vizinhos que pertençam a regiões homogêneas;

3. *Aplicação da técnica de deslocamento Médio:* aplica-se esta técnica até que o deslocamento seja menor que 0.1.;

4. *Remoção das Características Detectadas:* todos os pixels que pertencem a regiões finais das janelas são cortados, bem como os valores mapeados destes pixels no vetor de características Estes pixels são re-utilizados no passo 7 para que futuros erros sejam corrigidos;

5. *Interação:* repete-se os passos de 2 a 4 até que o número de janelas de buscas não exceda  $N_{min}$ .

6. *Determinação da característica inicial da paleta:* São consideradas cores significantes somente os pixels de cor que pertençam pelo menos a um componente (janela) conectado;

7. *Determinação da característica final da paleta:* o raio das janelas de características é multiplicado por  $\sqrt[3]{2}$  para que os pixels que foram removidos sejam recolocados em uma janela. Se um pixel não puder ser colocado em alguma janela, a cor ligada a este pixel será inserida no final da paleta;

8. *Pós-processamento*: São removidos todos os componentes (ou janelas) que possuem tamanho menor que  $N_{con}$ . Este passo, depende do objetivo da segmentação.

Na figura 2.5 é mostrado o resultado da detecção de bordas realizada pela técnica de segmentação de imagens coloridas.

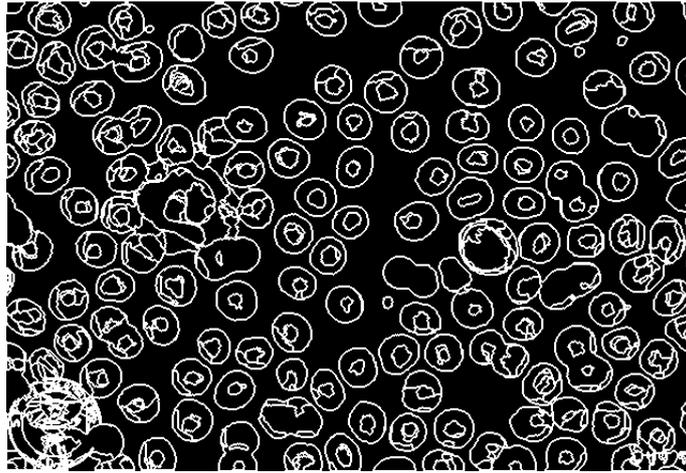


Fig. 2.5: Imagem resultante da técnica de Segmentação Colorida.

### 2.1.3 Comentários sobre as técnicas de Segmentação

Os resultados das técnicas de Canny e a de Segmentação de Imagens coloridas obtiveram resultados bem próximos. Isto é observado na parte selecionada na Figura 2.6, letras (a) e (b). Pode ser observado no detalhe da Figura 2.7 que ambas apresentaram o problema de colocarem células diferentes em um mesmo segmento.

Na letra (a) da figura 2.7 é apresentado o recorte das células ligadas ao leucócito da imagem original (Figura 2(a)) e em (b) e (c), o resultados das técnicas de Canny e de Segmentação de Imagens Coloridas aplicada a este mesmo recorte. Diferentes células unidas em um mesmo segmento causam muitas dificuldades quando se pretende identificar quais células estão neste segmento. Uma destas dificuldades, decorre do comportamento das cores de cada célula, o que impede que elas sejam separadas de forma automática por métodos mais simples, além de impedir que sejam

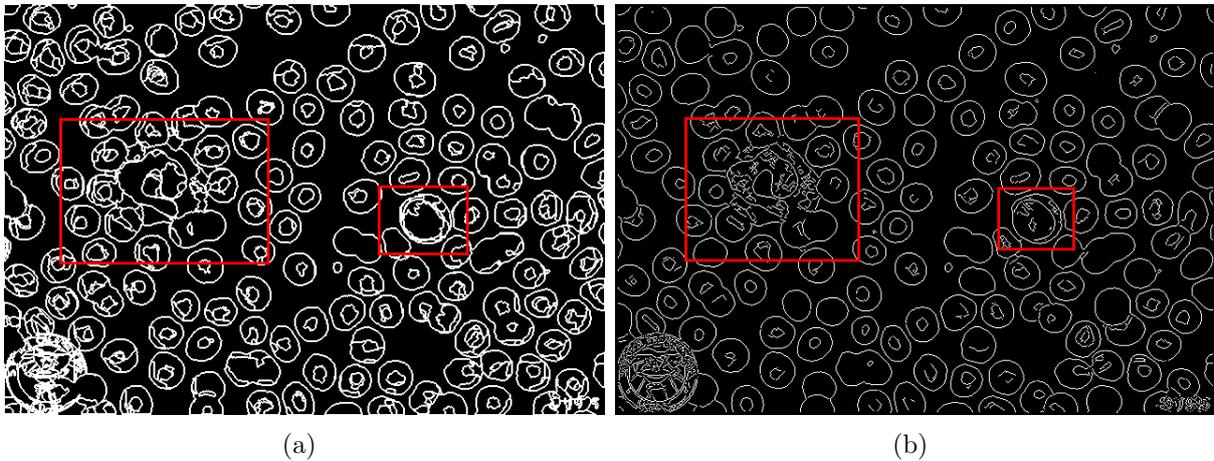


Fig. 2.6: (a) imagem resultante de Canny, (b) imagem resultante da segmentação colorida

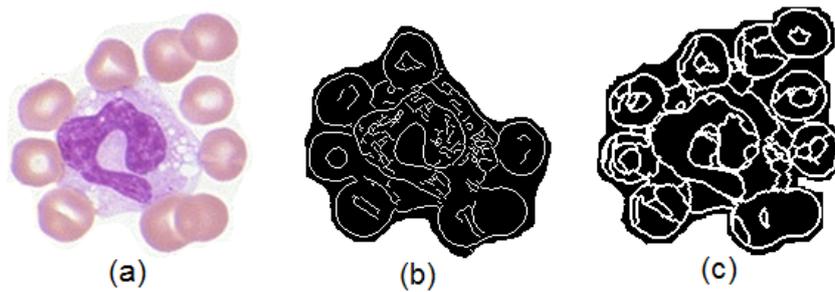


Fig. 2.7: Recorte do Leucócito da Figura 2(a),(b) o mesmo recorte passado pela técnica de Canny, (c) o mesmo recorte passado pela técnica Comaniciu.

realizados trabalhos futuros com as células separadamente. Por este motivo, neste trabalho, não se utilizou as técnicas de Canny nem de Segmentação de Imagens coloridas. No entanto, elas serão utilizadas para comparação dos resultados obtidos com a técnica melhorada de segmentação desenvolvida e descrita neste trabalho.

A técnica de Segmentação de Imagens coloridas cria uma imagem dividida em clusters. Cada cluster, apresenta cores homogêneas, dando um efeito de borramento na imagem como pode ser observado na Figura 2.8(b). A partir do resultado obtido na Figura 2.8(b) é realizada a segmentação proposta neste trabalho que será explicado no próximo capítulo.

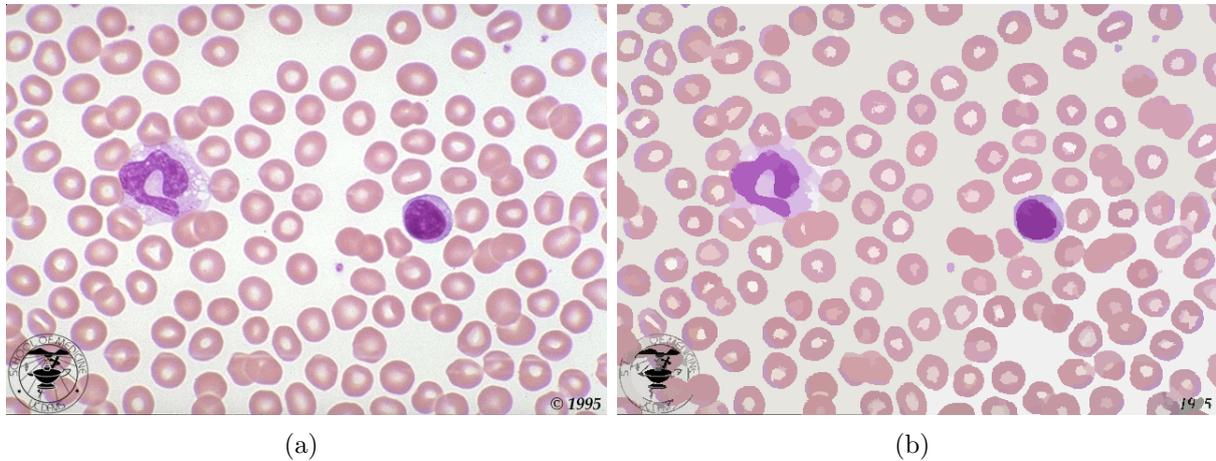


Fig. 2.8: (a) Imagem original (b) imagem com o efeito de borramento

## 2.2 Extração de imagens

A extração de imagens neste trabalho tem o objetivo de armazenar as células contidas na imagem de forma separada, gerando assim novas imagens.

A técnica de extração de imagens foi proposta e desenvolvida com base no “chain codes” Gonzalez [1]. Esta técnica cria uma seqüência numérica para representar as fronteiras da imagem, ao se percorrer a borda da mesma, utilizando uma numeração seqüencial, dada a vizinhança de 8 do pixel. A numeração dada a vizinhança de 8 do pixel, tem por objetivo enumerar a direção dos pixels vizinhos, para que seja possível criar a seqüência numérica.

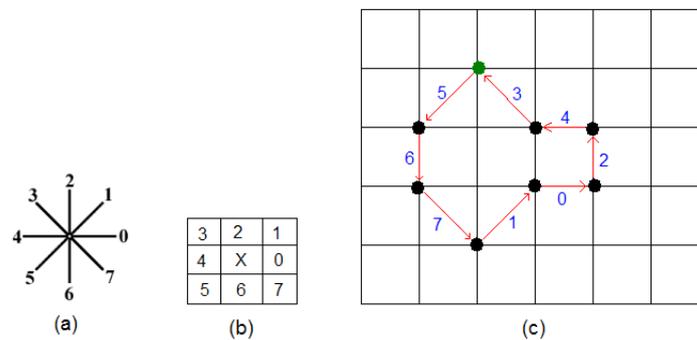


Fig. 2.9: “chain codes” ou código da cadeia.

Na Figura 2.9(a), tem-se a representação da numeração seqüencial dada por

Gonzalez [1], na letra (b), tem-se a mesma representação mas no formato tabular. Na letra (c), um exemplo da criação do código da cadeia, em que o pixel inicial foi marcado (pixel superior). A seqüência obtida pela numeração de Gonzalez [1] é: 5,6,7,1,0,2,4,3.

## 2.3 Classificação de Imagens

Segundo Spring [14], classificação é o processo de extração de informação de imagens para reconhecer padrões e objetos homogêneos ou não. Classificação de imagens, é uma maneira de atribuir uma classe ou categoria à imagem.

O reconhecimento de padrões, segundo Marques de Sá [15], extrai e seleciona características das imagens para a classificação utilizando alguma das seguintes abordagens: estatística [16][17]; sintática [18][19][20]; neural [21][22][23] e difusa [24][25][26]. Todas estas abordagens utilizam vetores de características para obter índices que serão utilizados para classificar a imagem no processo final.

A classificação realizada neste trabalho foi feito pela medição de similaridade entre imagens e a abordagem escolhida a abordagem estatística. Baseada nesta escolha, optou-se por pesquisar a aplicação de uma técnica chamada de spatiogram para realizar a classificação, apesar dos trabalhos já citados, por ser uma técnica estatística que obtém um vetor de características que representam a imagem de forma mais robusta, sem precisar da aplicação de técnicas de quantização [27][28] nem de morfologia matemática [29][30].

Vários trabalhos foram feitos para medir a similaridade entre imagens para recuperação de imagens baseada em conteúdo (CBIR) que foram baseados em cor [31][28], forma [29][32], textura [30][33], wavelets [34]. Mas o spatiogram se mostrou uma ferramenta mais simples para a realização deste trabalho.

### 2.3.1 Spatiogram

O spatiogram é uma técnica nova desenvolvida por Birchfield [3] em 2005, que armazena a distribuição espacial das cores presentes na imagem. Esta técnica é chamada de histograma de segunda ordem, pois além de armazenar o histograma da imagem é armazenado também o vetor de média e a matriz de covariância de cada cor, obtendo assim uma distribuição espacial das cores na imagem de forma estatística.

O spatiogram é apresentado pela equação 2.7.

$$h^2(b) = \langle \eta_b, \mu_b, \Sigma_b \rangle. \quad (2.7)$$

Onde:  $h^2$ : é o histograma de segunda ordem ou spatiogram.

$b$ : indica a posição do histograma.

$\eta_b$ : é a quantidade de pixels que tem a determinada posição indicada por  $b$ .

$\mu_b$ : um vetor de média  $(\mu_x, \mu_y)$ , onde  $\mu_x$  é a média da distribuição de cada posição do histograma no eixo x e  $\mu_y$  a média no eixo y conforme equação 2.8.

$$\mu_b = \frac{1}{\eta_b} \sum_{j=1}^n \lambda_j. \quad (2.8)$$

$\Sigma_b$ : a matriz de covariância da distribuição das cores dos pixels na imagem em x em y conforme equação 2.9.

$$\Sigma_b = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}. \quad (2.9)$$

Em que  $\sigma_{11}, \sigma_{12}, \sigma_{21}$  e  $\sigma_{22}$  são descritos conforme as equações de 2.10 a 2.12.

$$\sigma_{11} = \frac{\sum_{j=0}^n (x_i - \mu_x)^2}{\eta_b - 1}. \quad (2.10)$$

$$\sigma_{12} = \sigma_{21} = \frac{\sum_{j=0}^n (x_i - \mu_x)(y_i - \mu_y)}{\eta_b - 1}. \quad (2.11)$$

$$\sigma_{22} = \frac{\sum_{j=0}^n (y_i - \mu_y)^2}{\eta_b - 1}. \quad (2.12)$$

Sendo que :  $x_i$  : é uma coordenada do eixo x.  $y_i$  : é uma coordenada do eixo y.

O vetor de média e a matriz de covariância são calculados para cada ponto do histograma.

Os cálculos da matriz de covariância e do vetor de média das coordenadas em x e em y de cada ponto ou cor do histograma tornam esta técnica mais robusta por obter de forma estatística um vetor de característica que fornece a distribuição espacial das cores na imagem.

A distribuição espacial das cores permite a representação da imagem com uma riqueza maior de detalhes de uma maneira mais simples para a realização deste trabalho.

Devido a esta característica do spatiogram é que se optou por esta técnica para a realização da classificação de imagens.

### 2.3.2 Similaridade entre spatiograms

Birchfield [3] propôs um método para calcular a medida de similaridade entre spatiograms, que é calculado com base no vetor de média e na matriz de covariância. Este método computa a similaridade entre spatiograms com base como a soma de pesos entre dois histogramas utilizando a equação 2.13.

$$\rho(h, h') = \sum_{b=1}^B \Psi_b \rho_n(n_b, n'_b). \quad (2.13)$$

onde  $\Psi_b$  é uma distribuição multivariada gaussiana conforme a equação 2.14.

$$Psi_b = \eta \exp \left[ \frac{-1}{2} (\mu_b - \mu'_b)^T \sum_b^{\hat{-1}} (\mu_b - \mu'_b) \right]. \quad (2.14)$$

sendo:

$\mu_b$  e  $\mu'_b$  : os vetores de média.

$\sum_b^{\hat{-1}}$  : a soma dos determinantes das matrizes inversas de covariância.

$\eta$  : uma constante de normalização gaussiana conforme equação 2.15.

$$\eta = \frac{1}{2\pi \left| \sum_b^{\hat{-1}} \right|^2}. \quad (2.15)$$

$\rho_n(n_b, n'_b)$  de acordo com Birchfield [3], pode assumir três formas, que foram baseadas na interseção de histograma [35], 2.16, no coeficiente de Bhattacharyya [36], 2.17 e na equação 2.18 que foram propostas pelo mesmo.

$$\rho_n(n_b, n'_b) = \frac{\min(n_b, n'_b)}{\sum_{j=1}^B n_j}. \quad (2.16)$$

$$\rho_n(n_b, n'_b) = \frac{\sqrt{n_b n'_b}}{\sqrt{\left( \sum_{j=1}^B n_j \right) \left( \sum_{j=1}^B n'_j \right)}}. \quad (2.17)$$

$$\rho_n(n_b, n'_b) = \sqrt{n_b n'_b}. \quad (2.18)$$

Estes métodos utilizam todos os dados do spatiogram.

Apesar dos métodos propostos para medir a similaridade entre spatiograms, optou-se também por utilizar outros métodos que pudessem medir a distância entre matrizes de covariância e vetores de média. Os métodos escolhidos são os seguintes: Mahalanobis [5][37], Bhattacharyya [5][37] e Jeffries-Matusita [6][38][39].

Todos estes métodos estatísticos medem a probabilidade entre classes de padrões utilizando o vetor de média e matriz de covariância. Neste trabalho, serão utilizados

também para medir a similaridade entre distribuições de cores da imagem.

A seguir, são mostradas as equações utilizadas para calcular a similaridade de cada método.

Mahalanobis:

$$\rho_n(n_b, n'_b) = \sqrt{(\mu_b - \mu'_b)^T \sum_b^{\hat{-1}} (\mu_b - \mu'_b)}. \quad (2.19)$$

Bhattacharyya:

$$\rho_n(n_b, n'_b) = \frac{1}{8} (\mu_b - \mu'_b)^T \sum_b^{\hat{-1}} (\mu_b - \mu'_b) + \frac{1}{2} \ln \frac{\left| \frac{\Sigma_b + \Sigma_{b'}}{2} \right|}{\sqrt{|\Sigma_b| |\Sigma_{b'}|}}. \quad (2.20)$$

Jeffries-Matusita:

$$\rho_n(n_b, n'_b) = \sqrt{2(1 - e^{-\alpha})}. \quad (2.21)$$

onde  $\alpha$  é o resultado da distribuição de Bhattacharyya.

Todas as técnicas que medem as similaridades proposta nesta subseção, medem de forma estatística a distância entre distribuições de cores para cada banda do formato RGB.

## 2.4 Conclusão

Neste capítulo foram mostradas as fundamentações técnicas que serão utilizadas ao longo deste trabalho, bem como detalhados os motivos do seu uso neste trabalho.

A segmentação realizada é feita com base na melhoria da técnica de segmentação colorida. A técnica de Canny [7] será usada para comparar os resultados por ser considerada uma técnica ótima para a segmentação de imagens.

A classificação será realizada pela medida de similaridade entre imagens obtida por técnicas estatísticas para verificação de similaridade entre spatiograms, definidas neste capítulo. O spatiogram é uma técnica recente, que obtém a distribuição

espacial das cores da imagem de forma estatística. Devido a esta característica o spatioqram foi escolhido para realizar a classificação das patologias.

O desenvolvimento da extração proposta neste trabalho será mostrado no Capítulo de Metodologias.

# Capítulo 3

## Metodologias

Dada uma imagem digital do sangue, não é possível classificar a possível patologia existente na imagem de forma direta. A classificação é feita analisando-se as células de forma individual, nos seguintes passos: (i) segmentação, (ii) extração e (iii) classificação, estes passos são mostrados na Figura 3.1.

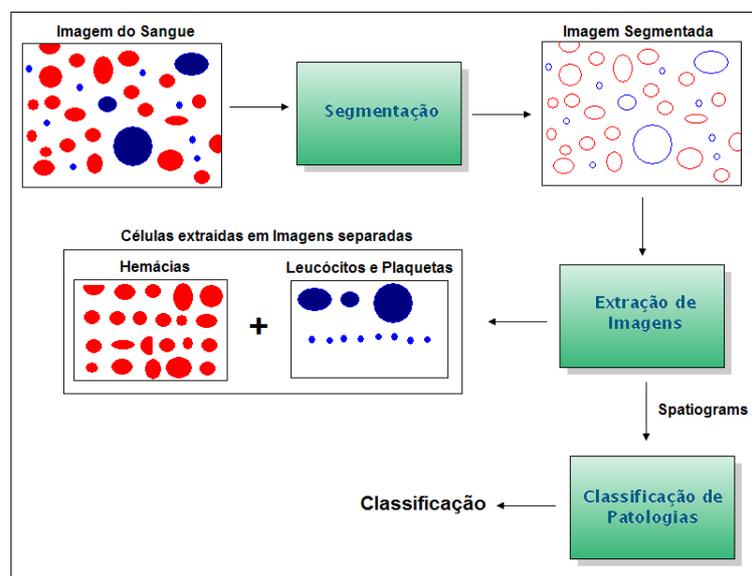


Fig. 3.1: Passos da classificação.

Neste capítulo será mostrado como foi feito o desenvolvimento de cada passo mostrado na Figura 3.1. Foi utilizada a ferramenta gimp foi utilizada para marcar

de preto as imagens do sangue que foram escolhidas como padrão para a obtenção das cores. Como Sistema Gerenciador de Banco de Dados, foi escolhida a ferramenta Mysql para o armazenamento e controle das tabelas utilizadas. Para visualização dos dados, foi utilizado o MysqlNavigator. Todo o trabalho realizado nesta dissertação foi feito em linguagem C, na plataforma Linux utilizando-se o gcc como compilador.

## 3.1 Segmentação

Segmentação é um trabalho imprescindível para fazer reconhecimento de objetos em imagens. Diversos trabalhos foram desenvolvidos para este objetivo. Alguns destes foram baseados em operadores morfológicos [40][41][42][43][44], redes neurais [45][46], lógica Fuzzy [47][48][49] e máquinas de aprendizagem [50][51][52] que em geral exigem alta complexidade computacional e uma boa estrutura de hardware e software.

A realização da segmentação em imagens digitais do sangue é difícil, pois, apresenta uma grande variação de cores, o que ocasiona a geração de muitas incertezas no momento de segmentar as células. A técnica proposta por Comaniciu [2] cria uma imagem dividida em cluster com cores homogêneas. Ao se homogeneizar estas cores, o número de cores presentes na imagem é diminuído sem perder a definição dos limites das células.

A diminuição destas cores leva a diminuição das incertezas na segmentação. Com a imagem do sangue apresentando cores homogêneas é possível dividir esta imagem em classes de cores.

Neste trabalho, foi considerado que uma imagem do sangue é composta por três tipos de classes de cores. A primeira é formada pelas cores de fundo, a segunda pelas cores dos leucócitos e plaquetas por serem muito próximas e a terceira pelas cores das hemácias.

O armazenamento das cores das classes de fundo, de Plaquetas e Leucócitos a partir de imagens escolhidas como padrão, permite a identificação dos pixels correspondentes a cada classe, utilizando a distância euclidiana entre as cores no modelo RGB.

Em geral, as cores de uma imagem digital estão no modelo de RGB, em que cada pixel é uma tripla (Red,Green,Blue), onde cada cor varia de 0 a 255, isto faz com que tenhamos um cubo tridimensional de cores, como pode ser visto na Figura 3.2.

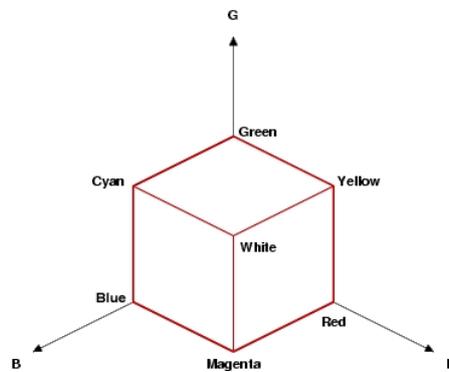


Fig. 3.2: Modelo de cor RGB.

Dados dois pixels  $p_1$  e  $p_2$  com posições  $(R_1, G_1, B_1)$  e  $(R_2, G_2, B_2)$  respectivamente, é possível saber a distância entre estes pixels pela distância euclidiana na equação 3.1.

$$d(p_1, p_2) = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2}. \quad (3.1)$$

Quanto menor é esta distância mais próximo estarão às cores destes pixels.

Este método possibilita descobrir se um determinado pixel apresenta cor próxima a de uma classe de cor ou não.

A obtenção de cores das classes de Fundo, Leucócitos e Plaquetas a partir de imagens referenciadas como padrão, permitem a identificação dos pixels pertencentes a cada classe nas outras imagens usando distância euclidiana.

Das células contidas nas imagens do sangue, as hemácias são as que apresentam

maior variação de cor com relação às outras células e o fundo. Este é o motivo de não se coletar as classes de cores das hemácias.

Com essa identificação, é possível criar um mapa de bits que informe exatamente os pixels de fundo, hemácias e leucócitos e plaquetas nas outras imagens.

Os pixels de hemácias são identificados no mapa de bits por exclusão, pois se um pixel não for de fundo ou de leucócitos e plaquetas, conseqüentemente será um pixel de hemácia.

O uso das classes de cores padrões através da distância euclidiana entre as cores, permite a criação de um primeiro mapa de bits com as regiões de cores de cada classe. A partir deste mapa, pretende-se criar um segundo mapa de bits com as bordas das células.

Na Figura 3.3(a) é mostrado um recorte de uma hemácia, a letra (b) mostra o mapa de bits da letra (a), e a letra (c) mostra o mapa de bits com as bordas geradas a partir do mapa de bits da letra (b).

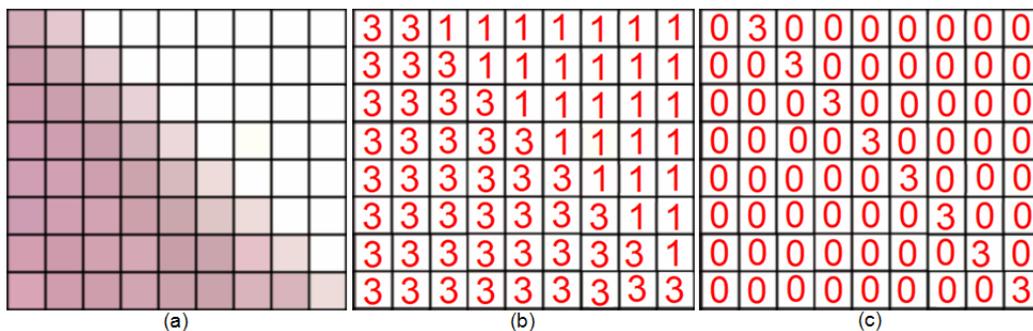


Fig. 3.3: Mapa de Bits.

Na seqüência, será mostrado como foi feita a obtenção de cores das classes de fundo, leucócitos e plaquetas, o estudo de distância de cores e a criação do mapa de bits e o procedimento feito para obter as bordas das células.

### 3.1.1 Obtenção das Cores que compõe as Classes

As cores são obtidas da seguinte forma:

- a) Coletam-se as imagens do sangue que tenham uma boa resolução;
- b) Separam-se as imagens por patologias e aplica-se a Oversegmentation de [2] em todas as imagens selecionadas;
- c) Escolhe-se de uma a três imagens por patologia para serem utilizadas como padrão;
- d) Para cada uma das imagens escolhidas no passo anterior, criam-se duas cópias destas, sendo que na primeira, marcam-se os pixels de todas as células com a cor preta e na segunda, marcam-se de preto somente as hemácias e o fundo;
- e) Nas imagens em que foram marcados os pixels de preto, armazena-se as cores que não forem pretas em um banco de dados, separando as cores de fundo e as cores de Plaquetas e Leucócitos;
- f) Repetem-se os passos (d) e (e) para cada imagem selecionada como amostra.

Para a realização do passo (a), foram coletadas 43 imagens a partir do site da Universidade Davis da Califórnia [53] e foram divididas em 8 patologias, em seguida, foi aplicada a Oversegmentation em todas elas como no passo (b). Das imagens coletadas, foram selecionadas 13 imagens como padrão para obtenção das classes de cores de fundo e de Leucócitos e Plaquetas, passo (c). Foram selecionadas 7 patologias do site [53], sendo que o sangue normal também foi considerado como uma patologia para que houvesse uma só classificação para o sangue. Os nomes destas patologias e a quantidade de imagens selecionadas por patologias são mostrados na Tabela 3.1.

Foi utilizada a ferramenta gimp para marcar os pixels das imagens de sangue de preto conforme o passo (d).

IT	PATOLOGIAS	QTDE
1	Normal	1
2	Anemias Microcíticas	2
3	Desordem MyeloProliferativo	2
4	Anemia Normocíticas	1
5	Leucemias Agudas	1
6	Anemias Hemolíticas	2
7	Anemias Macroscíticas	2
8	Desordem Limfoproliferativo	2

Tab. 3.1: Classificação de Patologias do Sangue

Foi desenvolvido um programa em linguagem C, para armazenar as cores das imagens que não foram marcadas de preto no banco de dados em tabelas separadas. A estrutura destas tabelas encontra-se na Figura 3.4, sendo que na letra (a) é mostrada a estrutura da tabela das cores de fundo e na letra (b), a estrutura da tabela de cores de Leucócitos e Plaquetas. A esta tabela, foi dada o nome de ‘Cor\_GB’, ou Cor dos Glóbulos Brancos, pois as células de Leucócitos e Plaquetas também são chamadas de Glóbulos Brancos.

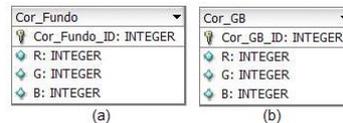


Fig. 3.4: Estrutura das tabelas de cores de Fundo e de Leucócitos e Plaquetas.

Os passos (d) e (e) foram repetidos para todas as imagens selecionadas como padrão.

Foram obtidos destas imagens padrão, 91 cores de fundo e 631 cores de Leucócitos e Plaquetas. Para efeito de teste, foram contadas as cores de hemácias e o valor desta contagem resultou em 6199 cores.

### 3.1.2 Estudo das distancias de Cores

Com as cores de Fundo e de Leucócitos e Plaquetas em um banco de dados, foi feito um estudo para descobrir qual a maior distância euclidiana entre duas cores que possa ser usada como limite máximo. Com a distância euclidiana entre uma cor conhecida e uma cor desconhecida, é preciso saber qual o maior valor que esta distância pode ter para saber se estas cores são próximas ou não.

Foram selecionadas pequenas distâncias ( $\leq 12$ ) com objetivo de se obter a proximidade das cores neste estudo, pois a maior distância euclidiana de cores no formato RGB, está entre a cor branca (255,255,255) e a cor preta (0,0,0) em que esta distância vale  $255\sqrt{3}$  ou 441,672..., isto vale exatamente o valor da maior diagonal (D) do cubo tridimensional de cor no formato RGB, como pode ser observado na Figura 3.5, em que D é dado pela equação:

$$D = \sqrt{(d^2 + (255)^2)}. \quad (3.2)$$

E  $d$  vale:

$$d = \sqrt{(255)^2 + (255)^2}. \quad (3.3)$$

Logo:

$$D = \sqrt{(255)^2 + (255)^2 + (255)^2} = \sqrt{3(255)^2} = 255\sqrt{3}. \quad (3.4)$$

Que é exatamente a distância entre as cores preta e branca.

Este estudo foi realizado marcando-se de preto os pixels das imagens restantes (30 imagens) que apresentaram uma distância euclidiana dentro de um dos valores pré-estabelecidos. É feita a marcação dos pixels nas imagens para que se possa ter um meio visual para análise das mesmas.

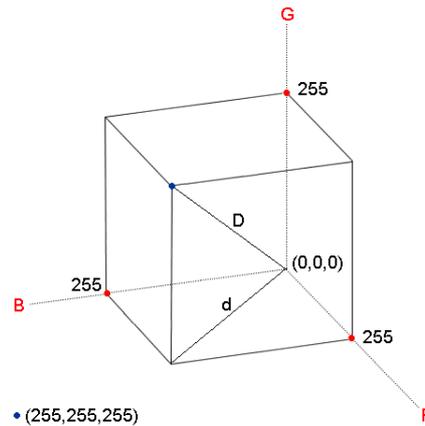


Fig. 3.5: Diagonal do Cubo Tridimensional de cor no formato RGB.

Como as cores são de duas classes distintas, foi preciso realizar um estudo para cada uma das classes de cores existentes.

### Estudo das distancias de Cores para a classe de cores de fundo

Foram marcados os pixels de fundo das imagens restantes em que a sua distância euclidiana fosse menor ou igual a um dos seguintes valores: 3, 4, 5, 7, 10 ou 12. Estes valores foram escolhidos aleatoriamente, sendo que cada valor foi testado com todas as imagens restantes.

A cada teste realizado, foi feita uma avaliação subjetiva nas imagens e foram contadas as imagens em que todos os pixels de fundo fossem marcados de preto. O resultado desta contagem esta mostrada na Tabela 3.2.

D.	N.I.C.	P.I.C.
3	12	40%
<b>4</b>	<b>15</b>	<b>50%</b>
5	22	73,3%
7	24	80%
10	28	93,3%
12	30	100%

Tab. 3.2: Resultados das distancias selecionadas aleatoriamente para as cores de fundo.

Onde, D., N.I.C. e P.I.C. significam respectivamente:

**Distância:** é a distância selecionada aleatoriamente;

**Número de Imagens Contadas:** é a quantidade de imagens que apresentaram uma grande quantidade pixels com cores próximas as que foram armazenadas;

**Porcentagem das Imagens Contadas:** é o percentual das imagens contadas com relação ao número total de imagens.

A distância de número 4 foi a que apresentou melhores resultados por não deformar muito os outros objetos da imagem como as hemácias.

### **Estudo das distâncias de Cores para a classe de cores de Leucócitos e Plaquetas**

Foram marcados os pixels de Leucócitos e Plaquetas das imagens restantes em que a sua distância euclidiana fosse menor ou igual a um dos seguintes valores: 1.8, 2, 3, 4, 5, 6, 7 ou 10. Estes valores foram escolhidos aleatoriamente, sendo que cada um destes foram testados com todas as imagens restantes.

Das 30 imagens restantes, só foi possível utilizar 12 por que as outras imagens não possuíam Leucócitos e algumas delas não apresentavam Plaquetas.

Neste estudo, optou-se por fazer uma análise subjetiva qualitativa ao invés da quantitativa, por que, à medida que a distância utilizada crescia, mais pixels das hemácias estavam sendo marcados como pertencentes à classe de cores de Leucócitos e Plaquetas, conseqüentemente isto levou a escolha de uma distância que apresentasse a menor distorção aceitável.

Os testes indicaram que a melhor distância euclidiana para a classe de cores de Leucócitos e Plaquetas foi a 5, pois foi a distância que atendeu as exigências anteriores.

Mesmo que a distância 5 tenha apresentado bons resultados, ainda assim há o problema de se ter pixels de hemácias marcadas de Preto. Para resolver o caso de

pixels de hemácias marcados como pixels de Leucócitos ou Plaquetas foi proposto o seguinte algoritmo:

- a) Percorre-se o mapa de bits da esquerda para a direita e de cima para baixo.
- b) Ao se encontrar cada pixel que tenha o código 2, utiliza-se o algoritmo flood-fill para contar os pixels pertencentes a região da classe de Leucócito ou Plaqueta, marcando-se estes pixels com o código 9. A posição do pixel encontrado é armazenado em uma lista simplesmente encadeada, onde é armazenado também o número de pixels de Fundo e de Hemácias que estão ao redor da região da célula de Leucócito ou Plaqueta. A contagem dos pixels de Fundo e Hemácias é feita pelo número destes pixels que são encontrados na vizinhança de 4 dos pixels de Leucócitos e Plaquetas.
- c) Percorre-se a lista simplesmente encadeada para verificar qual o nó da lista que apresenta a maior quantidade de pixels da classe de Leucócitos e Plaquetas.
- d) Descarta-se todos os nós que tiverem uma quantidade de pixels superior a 70% da maior quantidade de pixels da classe de Leucócitos e Plaquetas. Isto é feito é remarcando-se as regiões dos pixels com o código 2, utilizando a função floodfill a partir da posição do pixel inicial.
- e) Para cada um dos nós que restaram na lista, identifica-se qual destes nós possuem pixels de Hemácias ou de Leucócitos e Plaquetas realizando a verificação das suas quantidades de pixels de Fundo e de Hemácias. Se não houver nenhum pixel de hemácias nestes nós, então seus pixels são de Plaquetas ou Leucócitos. Se houver pixels de hemácias e não houver pixels de fundo nestes nós, então seus pixels são de hemácias. Se houver pixels de fundo e de hemácias ao mesmo tempo faz-se algumas considerações: (i) Se a quantidade de pixels de hemácias for maior que a quantidade de pixels de fundo, então os pixels deste nó que deveriam ser de Leucócitos e Plaquetas, na verdade são de hemácias; (ii) Se a quantidade de pixels de hemácias for menor que a quantidade de pixels de fundo em até 30%, então os pixels deste nó que deveriam ser de Leucócitos e Plaquetas, na verdade são de hemácias;

(iii) Se a quantidade de pixels de fundo for superior que a taxa anterior, então os pixels deste nó, são realmente de Leucócitos e Plaquetas.

### 3.1.3 Mapa de bits

O mapa de bits referidos neste trabalho é uma matriz onde cada posição pode armazenar um valor inteiro de 0 a 255. A matriz tem o mesmo tamanho da imagem.

Cada ponto do mapa de bits representa um pixel da imagem e será preenchido de acordo com a numeração dada a cada classe de cores. Para os pixels de fundo será atribuído o código 1, para os Leucócitos e Plaquetas será atribuído 2 e para as Hemácias o código 3.

Neste trabalho, serão criados dois mapas de bits. O primeiro, conterà as regiões dos pixels pertencentes às classes de cores de Fundo, de Leucócitos e Plaquetas e de Hemácias, este primeiro, será usado como base para a criação do segundo mapa.

O segundo, conterà somente as bordas das regiões de pixels pertencentes às classes de cores de Leucócitos e Plaquetas e de Hemácias, sendo este o mapa de bits utilizado no resto deste trabalho.

A seguir, são mostrados os passos para a criação dos mapas de bits com as Regiões das cores e em seguida, será apresentado como é criado o segundo mapa de bits, que armazena as bordas das células.

#### Mapa de bits com regiões dos pixels pertencentes às classes de cores

a) Compara-se cada cor dos pixels da imagem com as cores da classe de Leucócitos e plaquetas utilizando-se a distância euclidiana e onde houver um pixel que tenha uma cor com distância menor ou igual a 5, sua posição no mapa de bits recebe o código 2, indicando que temos um pixel de Leucócito ou plaqueta;

b) Compara-se cada cor dos pixels da imagem com as cores da classe de fundo utilizando-se a distância euclidiana e onde houver um pixel que tenha uma cor

com distancia menor ou igual a 4, sua posição no mapa de bits recebe o código 1, indicando que temos um pixel de fundo. Após a execução deste passo, é aplicado o algoritmo descrito na Subseção 3.1.2;

c) Marca-se com o código 3 onde for 0 no mapa de bits.

Para criar o mapa de bits da imagem com as regiões das classes de cores, primeiro selecionar a classe que se quer marcar no mapa de bits, em seguida, é preciso ler cada pixel da nova imagem em seguida percorre-se a tabela da classe de cor selecionada até que se encontre uma cor em que a distância euclidiana entre esta cor e a cor do pixel lido seja menor ou igual a distância estabelecida para a classe de cor selecionada.

### **Mapa de bits com as bordas das regiões de pixels pertencentes as classes de cores de Leucócitos e plaquetas e de Hemácias**

Com um mapa de bits em que se sabe exatamente onde estão os pixels das classes de cores de hemácias, Plaquetas e Leucócitos. Para obter as bordas, é criado um segundo mapa de bits, inicialmente com zero em todas as posições, onde só são marcadas as posições que no mapa de bits anterior possuem a numeração de uma das classes de cores citadas e que possua na sua vizinhança de 4 ao menos uma posição que tenha o número 1.

O encontro de uma posição no mapa de bits anterior que tenha na sua vizinhança de 4, uma posição que tenha a numeração da cor de fundo, indica que temos um pixel de borda e os pixels que não obedecem a esta regra são descartados.

O código marcado no novo mapa de bits é o número da classe de cores encontrado.

Na Figura 3.6 é mostrada um exemplo de como é criado o segundo mapa de bits com as bordas das células. A letra (a) mostra a identificação do primeiro pixel de hemácia em que existe ao menos um pixel de fundo na sua vizinhança de 4. A letra (b), mostra a primeira marcação feita com o código do pixel de hemácia no mapa de bits da borda. A letra (c), mostra o encontro do segundo pixel de borda da hemácia

e a letra (d), mostra a marcação do código da borda da hemácia no novo mapa de bits. As letras (e) e (f) mostram a finalização da criação do mapa de bits.

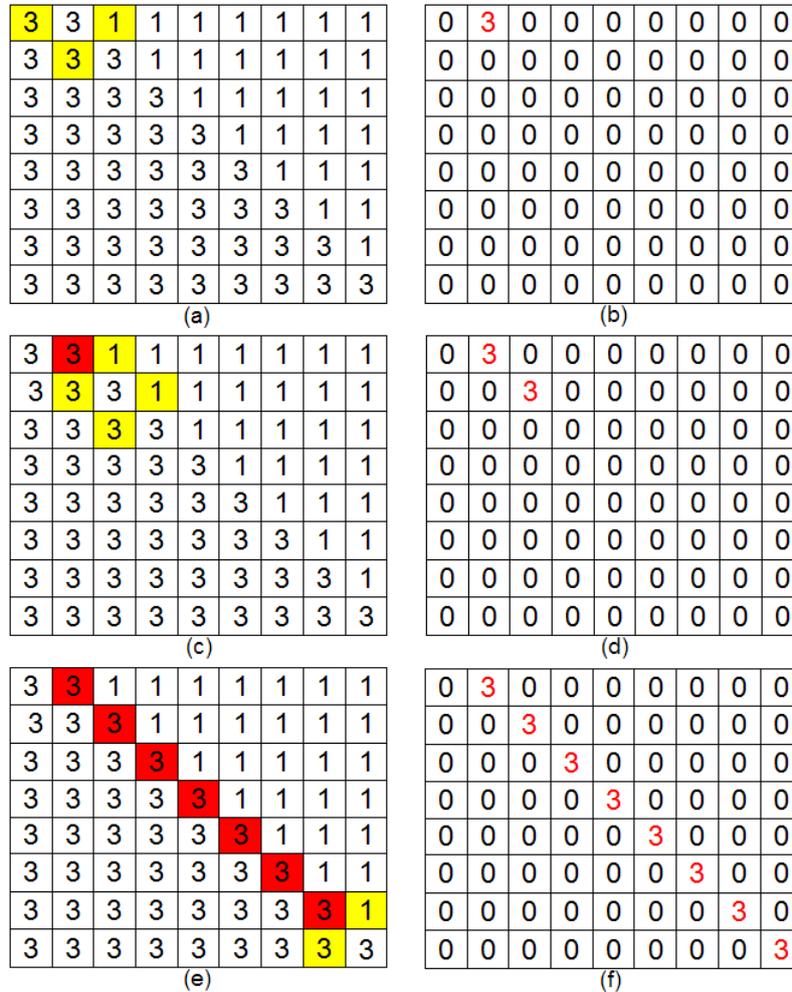


Fig. 3.6: Exemplo da criação do mapa de bits com as bordas das células.

### 3.1.4 Comentários

Foram mostrados nesta seção os passos para a realização da segmentação utilizados neste trabalho. Os resultados serão mostrados na seção do Capítulo de Resultados. Para os casos em que os pixels das Classes de cores de fundo e de Leucócitos e Plaquetas não foram marcados nas outras imagens, é necessário o recolhimento das cores destas classes para melhorar o resultado da segmentação, conforme os passos

descritos na Subseção 3.1.1.

## 3.2 Extração de Imagens

Em pesquisas realizadas, não foram encontradas informações suficientes a respeito de técnicas específicas para a extração de imagens a partir dos segmentos existentes na mesma, motivo pelo qual levou à elaboração de uma proposta para a execução da mesma. Os passos para a execução da técnica proposta serão detalhados nos subitens seguintes.

### 3.2.1 Chain Code ou Código da Cadeia

Segundo Gonzalez [1], o código da cadeia segue a seqüência numérica descrita conforme na Figura 7(a), no entanto, neste trabalho, optou-se por utilizar a seqüência conforme a Figura 7(b), devido ao fato de se considerar que todos os objetos na imagem são circulares e que o primeiro pixel nas bordas da imagem se encontram na parte superior esquerdo e que o percurso de detecção da borda segue o sentido anti-horário.

3	2	1	0	7	6
4	X	0	1	X	5
5	6	7	2	3	4
<b>(a)</b>			<b>(b)</b>		

Fig. 3.7: (a) seqüência de Gonzalez[1], (b) seqüência utilizada neste trabalho.

### 3.2.2 Técnica de Extração de imagens

Após a detecção das bordas da imagem (segmentação), tem-se um mapa de bits com as bordas das células presentes na mesma. Isto permite que as células sejam extraídas da imagem de forma eficaz.

Para a extração de um objeto de uma imagem, percorre-se cada pixel da imagem, da esquerda para a direita e de cima para baixo até encontrar o primeiro pixel do objeto.

A partir deste momento é iniciado o processo de extração do mesmo da imagem. Este procedimento é realizado com base nos seguintes passos:

1. Mapeamento das bordas das células;
2. Correção da Borda que apresenta problemas;
3. Delimitação de tiras da borda;
4. Realização da Extração.

Para cada objeto existente na imagem segmentada são executados os passos acima descritos, com as seguintes considerações:

a) Os objetos das imagens segmentadas são consideradas circulares e geometricamente fechadas (em que dado um ponto à borda começa e termina no mesmo ponto) e são divididos em quadrantes conforme Figura 3.8;

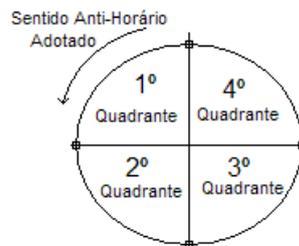


Fig. 3.8: Circulo adotado.

b) As bordas que possuem espessura de tamanho 1 (um pixel), onde cada pixel possui somente dois pixels vizinhos na sua vizinhança de 8 são consideradas bordas ótimas. As bordas que não obedecerem esta regra serão corrigidas para se transformarem em bordas ótimas. As Figuras 9(a) e 9(b) mostram um exemplo de borda ótima e borda não ótima. Os detalhes destas correções serão abordados posteriormente.

c) Baseado na quantidade de vizinhos que um pixel pode ter, de acordo com

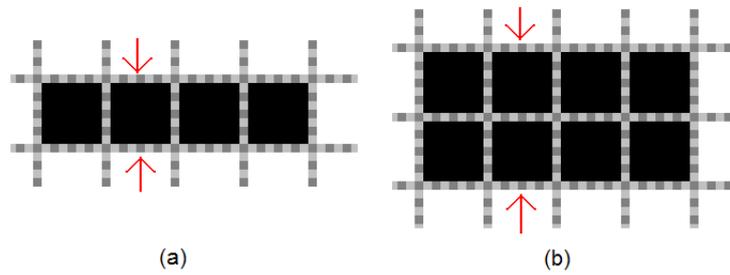


Fig. 3.9: Bordas com espessuras “1” e “2” pixels

a regra da letra b, foram desenvolvidos códigos para o posicionamento destes vizinhos com base na localização dos mesmos dentro do “chain codes”. Por exemplo, para determinar se um pixel tem o código 1, é necessário verificar se estes vizinhos se encontram nas posições 0 e 4 da numeração estabelecida na Figura 3.7(b). Se estiverem, então o pixel terá código 1. A Figura 3.10 mostra os códigos das configurações do posicionamento dos pixels na vizinhança de 8. Para cada código foi dado o nome de “código do vizinho”. Devido a geração de incertezas no momento do mapeamento da borda, não foram utilizadas outras distribuições de posicionamento dos pixels.

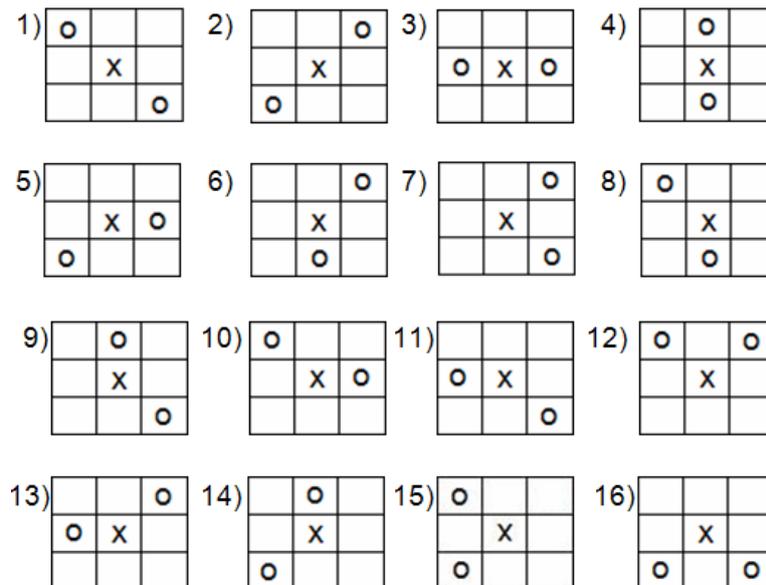


Fig. 3.10: Código dos Vizinhos.

d) Para que se tenha a garantia de que o primeiro pixel encontrado na célula a ser retirada seja o pixel superior, a imagem será percorrida da esquerda para direita e de cima para baixo. Para cada célula extraída será gerada uma nova imagem.

e) Todo o trabalho realizado com as bordas é desenvolvido em cima do mapa de bits. Onde for borda, o código é diferente de 0, cada ponto do mapa de bits é referenciado como se fosse um pixel da imagem.

### Mapeamento das bordas das células

O mapeamento das bordas de uma célula é realizado pela criação de uma lista simplesmente encadeada com as posições  $x$  e  $y$  de cada pixel na borda, utilizando os códigos criados anteriormente. Esta lista é temporária, pois será descartada quando o vetor de tiras for criado. Isto será abordado mais a frente. A seguir, será visto o algoritmo utilizado para criar a lista simplesmente encadeada com base na movimentação feita na borda de cada célula. A Figura 3.11(a) mostra um exemplo de borda.

Algoritmo para criação da lista simplesmente encadeada.

1. Quando é encontrado o primeiro pixel da borda, é verificado se o código do vizinho do pixel é 5 ou 16. Se for qualquer um dos dois, sabe-se que o próximo pixel a ser pesquisado é o pixel que se encontra na posição 2 da Figura 7(b) ou o pixel que se encontra na parte inferior esquerda. Em seguida, armazena-se a posição do primeiro pixel encontrado na lista junto ao seu código do vizinho. A Figura 11(b) mostra um exemplo da identificação do primeiro pixel de borda. É possível observar que o pixel marcado em preto no centro da máscara 3x3 tem código 5 (comparando-se a janela 3x3 da Figura 11(b) com a distribuição do código 5 da Figura 3.10). Observando-se a máscara 3x3 da Figura 11(b), pode-se notar que o próximo pixel a ser pesquisado será o pixel que se encontra na parte inferior esquerda.

2. Ao se pesquisar o segundo pixel detectado pelo passo anterior, verifica-se

o código do vizinho do mesmo e a posição do último pixel armazenado na lista simplesmente encadeada dentro da mascara 3x3. Isto é feito para achar a posição do próximo pixel a ser pesquisado. Pode-se ter uma idéia disto na Figura 3.11(c) onde o pixel atual esta em vermelho na janela 3x3. O pixel anterior está em preto e o próximo pixel a ser visitado esta em azul.

3. A cada novo pixel visitado verifica-se o código do vizinho e a posição do último pixel armazenado na lista, pois o próximo pixel a ser visitado será o pixel que sobrar na janela 3x3. Repete-se este passo até que se encontre o primeiro pixel novamente que esta armazenado na lista simplesmente encadeada. Pode se ver a repetição deste passo nas letras (d), (e) e (f).

4. A cada pixel visitado, é verificado o valor da posição do pixel em x e em y para encontrar o maior e o menor valor em x e em y. Estes valores são armazenados nas variáveis: maiorx, menorx, maiory e menory. Para que o cálculo da altura e da largura da célula possa ser calculada após a borda ser percorrida.

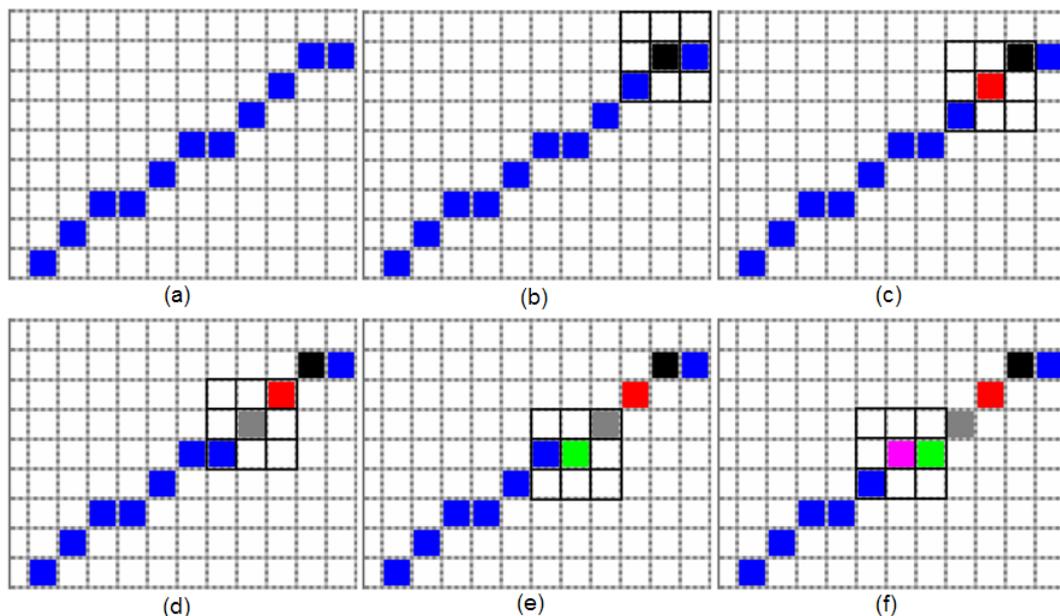


Fig. 3.11: Caminhamento pela borda.

É com base na observação dos pixels de borda na imagem que se concluiu que

sempre o primeiro pixel da borda terá código 5 ou 16. As bordas que apresentarem códigos diferentes logo no primeiro pixel, serão corrigidas também.

Na Figura 3.11(b), na mascara 3x3, aparecem dois pixels em azul. Ao se encontrar o primeiro pixel de borda que tenha código 5 ou 16, sempre se irá considerar o pixel que aparece na posição inferior esquerda da mascara como o segundo pixel a ser visitado. O pixel que sobrar, do primeiro pixel de borda, Figura 3.11(b), será ignorado inicialmente. Isto é feito para que a movimentação da borda seja feita no sentido anti-horário. O pixel ignorado inicialmente será visitado quando a movimentação voltar ao mesmo ponto inicial da borda.

### Correção das Bordas que apresentam problemas

Quando é encontrada uma borda que não obedeça à regra da borda ótima, é preciso corrigi-la para que obedeça a esta regra.

Apesar da segmentação gerar bordas ótimas, existem casos em que há bordas com problemas de distribuição de pixels diferentes das que já foram estabelecidas na Figura 3.10, ou bordas com espessura de tamanho maior que 1. Para resolver este problema, optou-se por marcar os pixels mais internos da borda para que fosse gerada uma nova borda. Os pixels restantes são chamados de aresta por ficarem de fora da nova borda e serão apagados após a criação da nova borda. O número de pixels descartados é pequeno se comparado com o total de pixels da célula a serem extraídos.

A correção da borda da célula é feita no mapa de bits.

Em um mapa de bits com as bordas das células, não há como saber se um determinado bit do mapa é interno ou não. Pois, as bordas das hemácias têm código 3 e as bordas das células de Leucócitos e Plaquetas tem código 2 e o resto dos bits têm código 0 tanto internamente como externamente às bordas.

Para a realização da correção é utilizada a técnica floodfill [54]. Esta técnica é

um algoritmo recursivo que é usado para preencher áreas em imagens.

A técnica para correção de bordas é feita com base nos seguintes passos:

1. Utiliza-se o algoritmo floodfill para marcar todos os pixels de borda no mapa de bits com um código -1. Isto é feito a partir do primeiro pixel de borda encontrado, até que se encontre um pixel de borda interno. Após o encontro do primeiro pixel interno, os pixels da borda marcados com o código -1 são remarcados com o código 1 utilizando a mesma função;

2. Encontrado o primeiro pixel interno da célula no mapa de bits, utiliza-se novamente a técnica de floodfill para preencher a parte interna da borda no mapa com um novo código de fundo (4). Ao se encontrar um pixel interno no mapa em que na sua vizinhança de 4 exista ao menos um pixel de borda, o mesmo será marcado com um novo código de borda (5). A cada pixel de borda marcado é verificado se o mesmo possui o menor valor em x e o menor valor relativo em y para armazená-los como a posição do novo pixel inicial da nova borda. O pixel inicial de toda e qualquer borda sempre terá o menor valor em x e o menor valor relativo em y, pois a imagem é percorrida conforme descrito na Seção 3.2 letra (b). Esta propriedade garante o menor valor em x, mas não em y.

3. Após a nova borda ser criada no mapa, utiliza-se o algoritmo floodfill para preencher a parte interna da borda novamente com o código '0'.

4. Percorre-se a nova borda no mapa para apagar os pixels de aresta, utilizando-se o algoritmo floodfill para marcar estes pixels com o código '0'.

É possível ter uma compreensão melhor deste algoritmo, observando-se os passos na Figura 3.12. Nesta figura, é apresentada apenas uma representação de como funciona o algoritmo de correção de bordas no mapa de bits.

Pode-se verificar nas letras da Figura 3.12 que:

- (a) A Imagem em que a borda da célula não possui pixels com a configuração igual ao que já foi pré-definida na Figura 3.10;

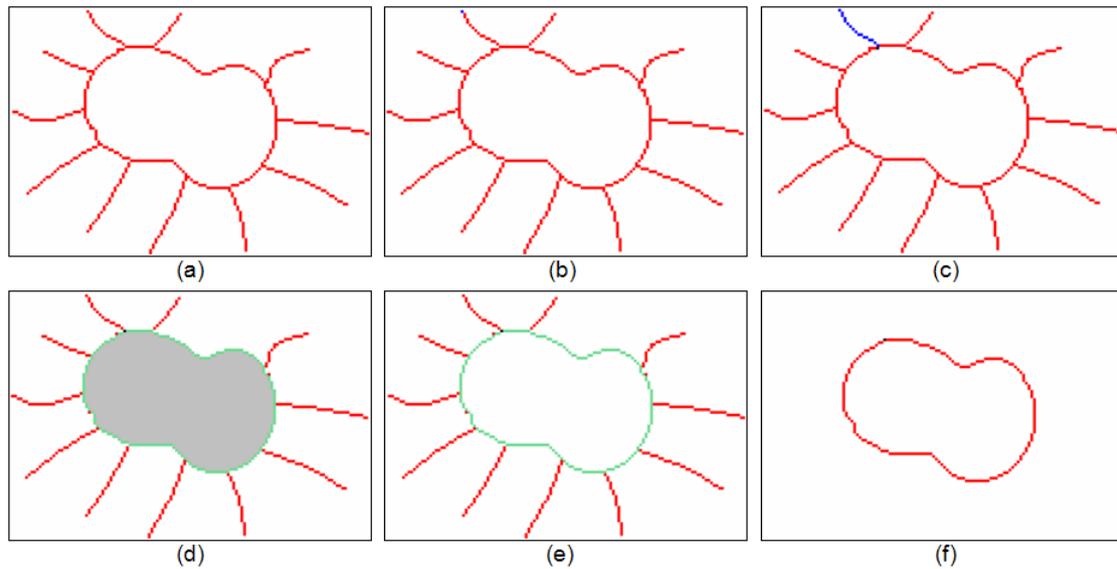


Fig. 3.12: Exemplo de correção da borda.

(b) O pixel inicial de borda, marcado no mapa de bits com o código -1, este pixel está em azul;

(c) Marcação em azul no mapa de bits, dos pixels de borda com o código -1 até que se encontre o primeiro pixel interno. O primeiro pixel inicial da parte interna está em preto. Após se achar o pixel interno da borda, remarcam-se os pixels que têm o código -1 com o código 1 (até aqui foi realizado o passo 1);

(d) Os pixels de borda que foram marcados com o código (5) no mapa de bits estão em verde claro. Os pixels de fundo da parte interna da borda são marcados com o código (4). Estes pixels aparecem em cinza. É calculada também a posição do pixel inicial da nova borda (até aqui foi realizado o passo 2);

(e) Os pixels da parte interna da borda são remarcados com o código 0 (passo 3). Estes pixels foram marcados de branco novamente;

(f) Agora, ao ser apagada as arestas, uma nova borda é gerada. Isto é feito, marcando-se com o código 0 no mapa de bits os pixels de aresta, como indicado no passo 4. Esta nova borda pode agora ser percorrida para preencher a lista de borda.

Para identificar um pixel interno da borda de uma célula, é preciso verificar a

vizinhança de 8 do pixel atual e verificar se ocorre uma das configurações definidas na Figura 3.13. Estas configurações são utilizadas somente para a correção das bordas. Cada numeração dada, é um código utilizado para identificação do pixel interno.

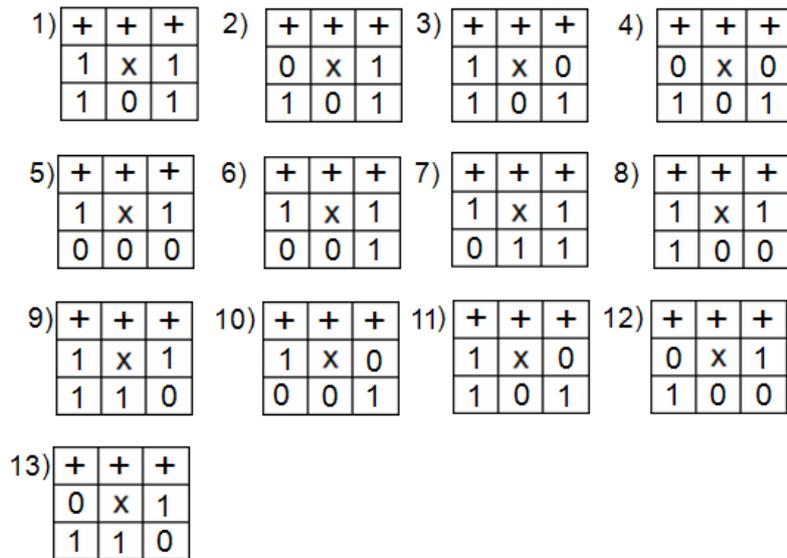


Fig. 3.13: Configuração utilizada para identificação do pixel interno da borda.

As configurações da Figura 3.13 foram criadas com base na observação da distribuição dos pixels de borda nas imagens segmentadas. Outras possíveis configurações foram rejeitadas devido ao aumento das incertezas bem como por não ocorrerem na prática, apesar de serem possíveis teoricamente.

O sinal '+' na Figura 3.13 indica que o valor do código deste pixel é irrelevante. O código 1 indica pixel de borda e o código 0 indica pixel de fundo. Todo número 0 na parte inferior da mascara 3x3 já é considerado pixel interno. Onde tiver somente um zero, a posição deste pixel será escolhida e onde houver dois ou mais zeros, será escolhido o pixel que se encontrar na parte inferior do centro da mascara.

À medida que a função floodfill vai cobrindo a borda da célula, a posição do pixel pesquisado vai descendo. Isto garante que o pixel interno da borda esteja na parte inferior da mascara 3x3. A distribuição dos pixels de borda e de fundo nas posições

de 1 a 5 do “chain code” Figura 3.7(b) possibilita a identificação do pixel interno de forma segura. Isto faz com que o código que esteja na parte superior da máscara seja irrelevante na identificação.

Na Figura 3.14(a), (b) e (c) pode ser observado um exemplo da aplicação da configuração da Figura 3.13. Os pixels que se encontram em azul são os pixels cobertos pela função floodfill. Todos os pixels em branco são de fundo, os em vermelho, são de borda, os em preto são os pixels pesquisados atualmente na máscara 3x3 e os pixels em cinza, são os pixels selecionados como pixel interno da borda.

Na Figura 3.14(a), pode ser observado que a configuração encontrada é a de número 12 da Figura 3.13. Na Figura 3.14(b) pode a configuração de número 5 e na Figura 3.14(c) a configuração de número 12. Em todas as letras da Figura 3.14, nota-se que o valor da parte superior da máscara 3x3 são realmente irrelevantes.

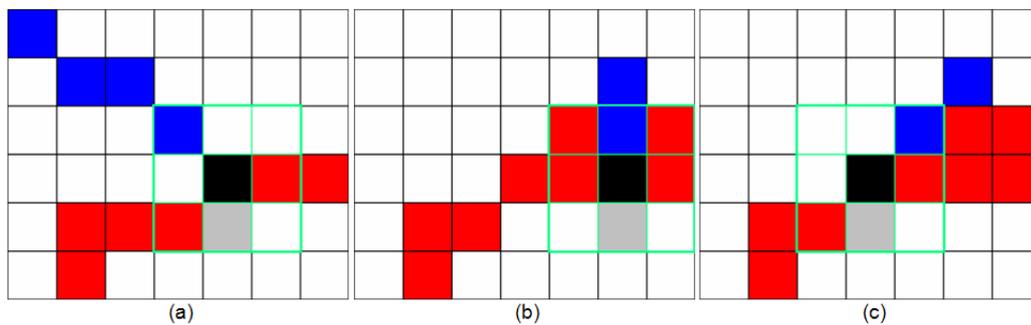


Fig. 3.14: Exemplo da identificação do pixel interno da borda.

### Delimitação das tiras da borda

Com os pixels de borda armazenados em uma lista simplesmente encadeada, foi proposta uma solução para realizar o armazenamento das tiras de uma célula em um vetor de tiras para facilitar sua extração. Uma tira é um segmento de reta delimitado pela borda. Cada linha da célula pode ter uma ou mais tiras, pois a mesma é dividida em tiras para permitir uma extração limpa e eficiente da célula.

Esta solução foi proposta para não serem extraídos outros elementos da imagem, tais como cores de outras células próximas ou cores de fundo visto que esta última, em geral, possui ruídos que podem atrapalhar o reconhecimento da célula posteriormente.

A Figura 3.15 mostra um exemplo para uma melhor compreensão a respeito das tiras da célula. Cada tira é identificada por  $T_n$ , onde  $T$  é de tira e  $n$  a numeração dada a cada tira. Os índices ‘i’ e ‘f’ significam inicial e final. O vetor de tiras armazena para cada linha da célula: o número de tiras da linha, a posição em  $x$  da linha e uma lista simplesmente encadeada que contém o valor inicial e final de cada tira existente. Este vetor tem o mesmo tamanho da célula. Por exemplo, a tira  $T1$  aparece indicada por  $T1i$  e  $T1f$  respectivamente como o início e o fim da Tira 1.

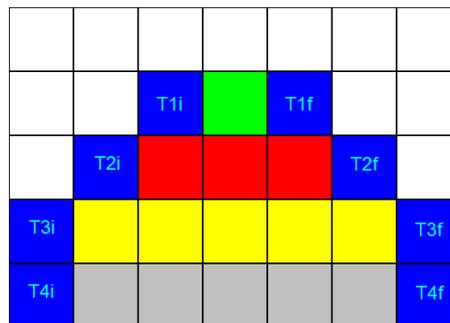


Fig. 3.15: Tiras da imagem.

Para uma melhor compreensão de como funciona este vetor de tiras, é mostrado um exemplo na Figura 3.16, onde ‘ $P$ ’ indica a posição do vetor, ‘ $N\_tiras$ ’ indica a quantidade de tiras, ‘ $Valor\_x$ ’ a posição em  $x$  da linha da célula, ‘ $No$ ’ o ponteiro da lista simplesmente encadeada e cada nó da lista armazena ‘ $yi^i$ ’ e ‘ $yi^f$ ’, que são as variáveis que armazenam o valor inicial e final das tiras no eixo  $y$  de cada tira.

O funcionamento de cada nó da lista de tiras é mostrado como um exemplo na Figura 3.17. Na letra (a), é mostrado que os nós da lista indicam o início e o fim das mesmas. Na letra (b), é mostrado o que é armazenado em cada nó. Por exemplo, a tira  $N1$  armazena o valor inicial e final da tira 1 em  $yi1$  e  $yf1$  respectivamente, e

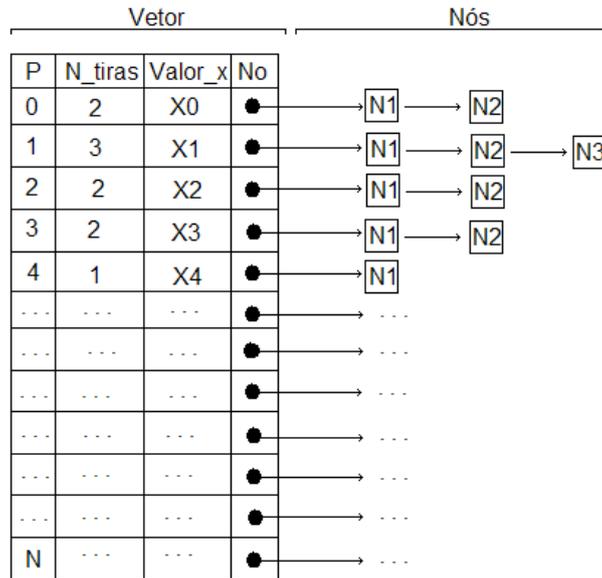


Fig. 3.16: Vetor de tiras.

assim por diante.

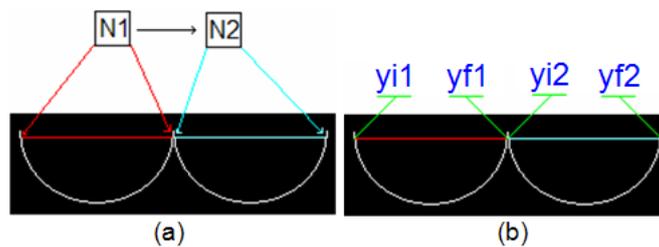


Fig. 3.17: Exemplo do que os nós ou tiras armazenam.

Neste trabalho definiu-se que em todas as imagens que apresentam a inserção de alguma tira, a seta que sai do nó para a direita refere-se ao preenchimento da variável 'yf' e a seta que sai do nó para a esquerda, refere-se ao preenchimento da variável 'yi'. Se não houver nenhuma seta em um destes lados, indica que uma destas variáveis foi preenchida com 0, como pode ser visto na Figura 3.18.

O objetivo do vetor de tiras é obter todas as tiras das linhas da célula que se quer extrair. Um exemplo mais detalhado é mostrado na Figura 3.19. Onde pode observado na letra (a), que na posição 0 do vetor, a lista armazena uma única tira que contém a tira da primeira linha da célula. A letra (b), mostra que a lista da

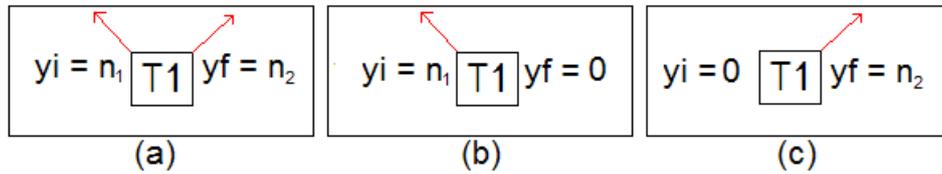


Fig. 3.18: Exemplo do preenchimento das variáveis  $y_i$  e  $y_f$ .

posição 1 armazena as tiras da segunda linha. A letra (c), mostra que as tiras da terceira linha são armazenadas pela sua respectiva lista de tiras na posição 2 do vetor, e as outras posições do vetor armazenam as tiras das linhas restantes da célula.

Assim, foram vistos dois tipos de listas simplesmente encadeadas. A primeira, contém a posição dos pixels de borda da célula a ser extraída, a segunda, a localização das tiras em cada linha da célula.

As listas do vetor de tiras são preenchidas à medida que os nós da primeira lista são lidos.

O algoritmo de preenchimento do vetor de tiras é descrito a seguir.

1. Lê-se um nó da lista de borda;
2. Se a posição em x do nó lido não estiver cadastrada no vetor, armazena-se este valor em uma nova posição do vetor. Em seguida, é inserido a primeira tira na lista simplesmente encadeada deste ponto do vetor com o valor de y armazenado em  $y_i$  da nova tira.
3. Se o valor em x do nó lido já estiver cadastrado no vetor, armazena-se o valor em y de acordo com a propriedade do quadrante em que estiver o pixel da borda. Estas propriedades serão mostradas mais a frente, bem como a inserção do valor em y do pixel atual na lista simplesmente encadeada.

4. Apaga-se o nó lido no passo 1.

5. Repetem-se os passos de 1 a 4 até que não haja mais nó na lista de borda.

Foi convencionado neste trabalho, que cada nó da lista de tiras será chamado

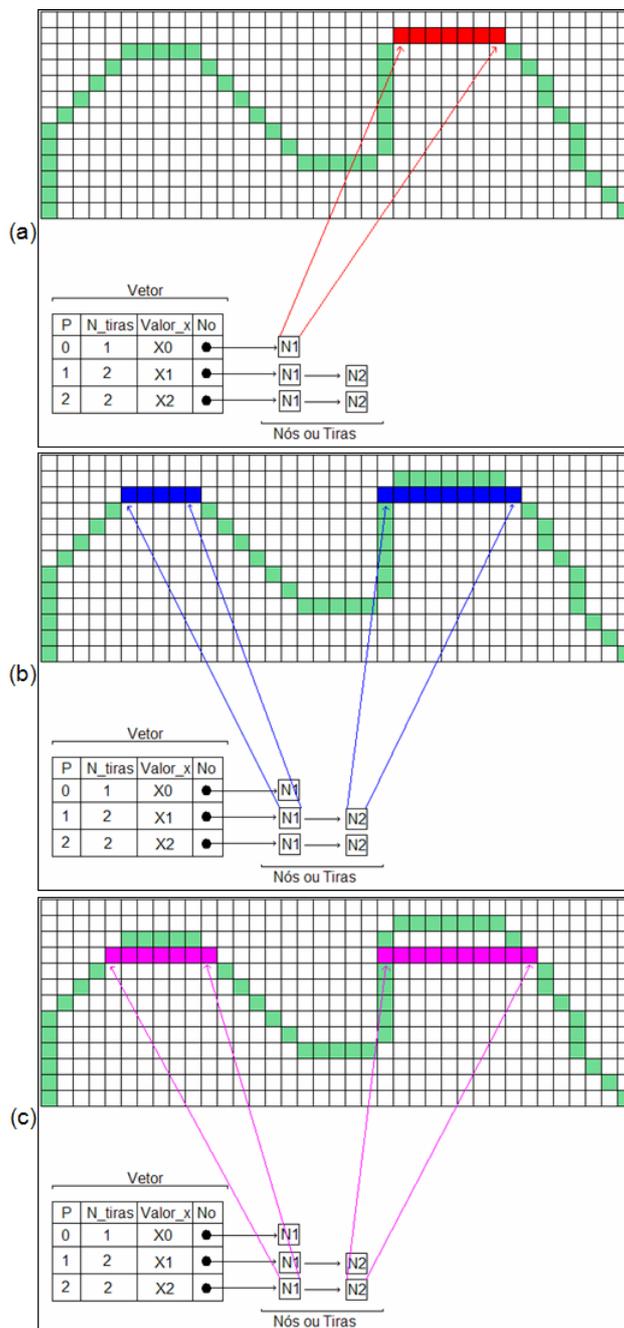


Fig. 3.19: Exemplo do funcionamento do vetor de tiras.

simplesmente de tira e não de nó da lista de tiras.

Os pixels de borda, de acordo com o seu código do vizinho, receberam uma classificação para facilitar a extração e isto será mostrado a seguir.

## Classificação dos Pixels

### 1. Pixels Normais.

Pixels normais são os pixels que podem ser usados como inicial e final de tira, pois possuem códigos: 1, 2, 4, 5, 6, 7, 8, 9, 14 e 15.

Os pixels que apresentam estes códigos têm a particularidade de delimitar apenas as tiras da linha da célula. São mais fáceis de manipular, pois os mesmos ou são pixels iniciais ou finais de tira. Na Figura 3.20 é mostrada a configuração da distribuição dos pixels normais.

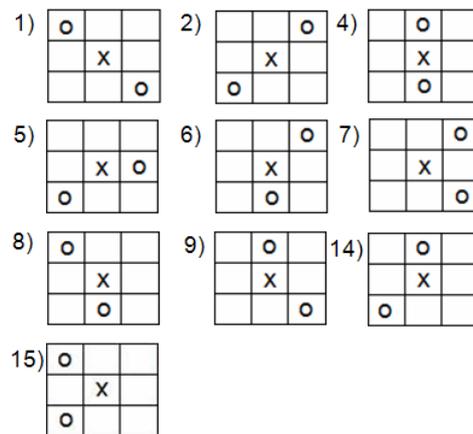


Fig. 3.20: Configuração dos pixels normais.

### 2. Pixel inicial de linha

São pixels que iniciam e finalizam uma linha contínua na borda. Têm código igual a: 5, 10, 11 e 13. Na Figura 3.21 é mostrada a configuração destes pixels.

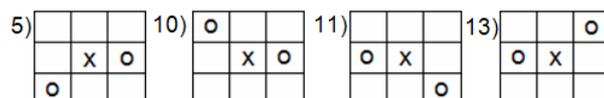


Fig. 3.21: Configuração dos pixels de Linha.

A particularidade de uma linha contínua na borda, é que os limites desta devem ser armazenados no vetor de tiras. Quando uma linha começa com pixels que tenha

código 5 ou 10, esta linha sempre termina com pixels que tenha código 11 ou 13 e vice-versa. Isto pode ser observado na Figura 3.22. Na letra (a), o código do vizinho da parte central da mascara 3x3 em vermelho é 10, esta linha é contínua na borda e termina com o código 11 como pode observado na mascara 3x3 em azul. Se forem comparadas as distribuições dos pixels dentro das mascaras 3x3 nas outras letras, com a distribuição dos pixels que foi definido na Figura 3.21, poderá ser observado a confirmação o que foi escrito.

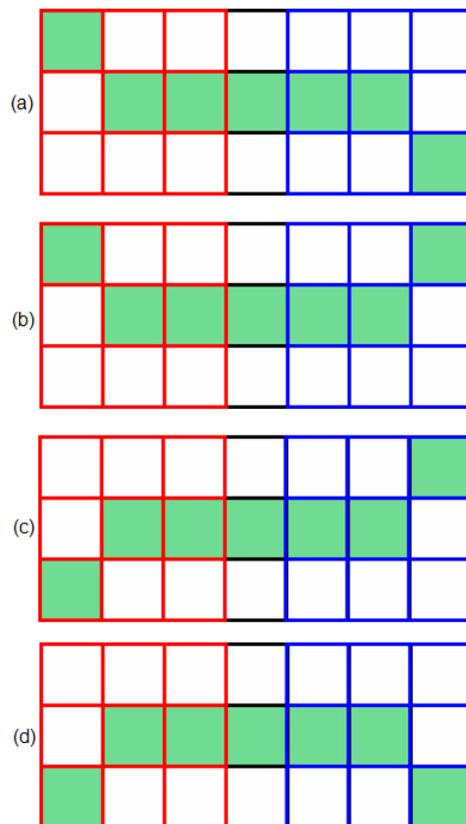


Fig. 3.22: Exemplo da identificação da linha continua na borda.

Na Figura 3.23, mostra um exemplo de borda do 1º Quadrante, onde é possível ver o pixel inicial de linha com código 11 em amarelo, o pixel inicial de linha com código 13 em rosa, e também o pixel brotoeja em vermelho. O conceito de pixel brotoeja será explicado a seguir.

### 3. Pixel brotoeja.

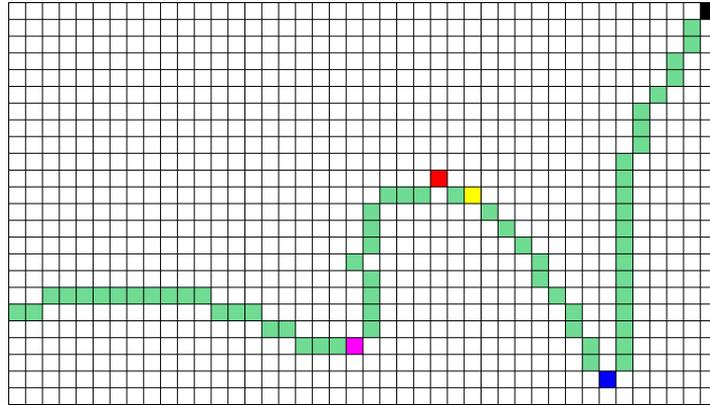


Fig. 3.23: Exemplo de borda no 1º Quadrante.

Este nome “brotoeja” deve-se ao fato que este pixel sempre está sozinho e fora de uma linha na parte externa do objeto, pode se observar isto no pixel que esta em vermelho na Figura 3.23.

A classificação “pixel brotoeja” depende do quadrante onde se encontram estes pixels, pois sempre possuem códigos: 12 ou 16, como se pode ver na Figura 3.24.

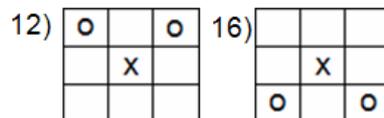


Fig. 3.24: Códigos do vizinho do Pixel Brotoeja.

No primeiro e no quarto quadrante, só recebem esta classificação os pixels que tem o código 16. No segundo e no terceiro quadrante somente os pixels que tiverem o código 12.

Quando ocorrer o caso de se encontrar pixels com código 12 no primeiro ou no quarto quadrante, ou encontrar pixels com código 16 no segundo ou no terceiro quadrante, estes pixels serão desconsiderados, pois, se estes pixels forem considerados nesta situação, vai ocorrer o caso da divisão de uma tira em duas novas tiras, sendo que a mesma poderia ser armazenada em uma única. A Figura 3.25 clareia melhor estes casos, onde na letra (a), pode ser observado que a tira armazena a informação de todo o segmento da linha da célula na parte superior. Se for considerado o pixel

de código 12 no primeiro ou no quarto quadrante, ocorreria o que se vê na letra (b), em que se têm duas tiras armazenando o mesmo segmento de linha. Já na letra (c), pode-se ver que existe uma única tira armazenando todo o segmento de linha da célula na parte inferior. Se for considerado o código 16, no segundo ou no terceiro quadrante, ocorreria o que se vê na letra (d) em que há duas tiras armazenando o mesmo segmento da linha.

Estes casos serão evitados neste trabalho.

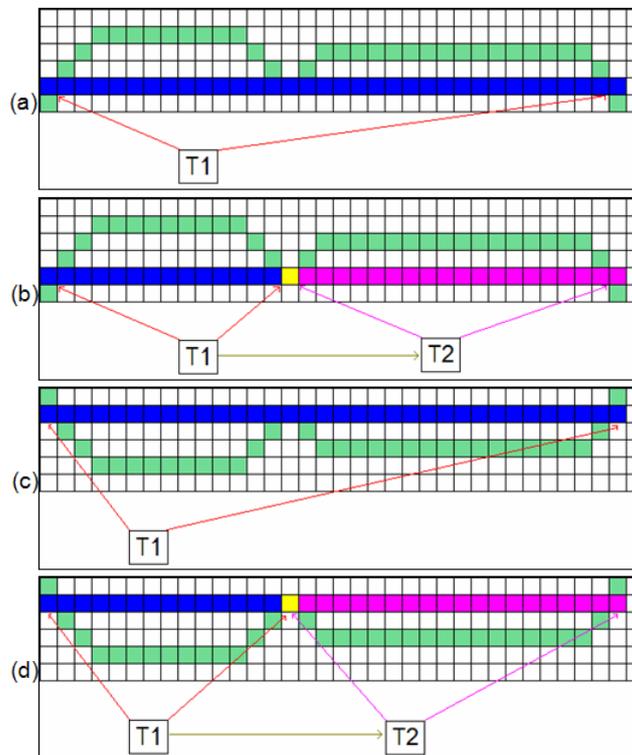


Fig. 3.25: Exemplo da inserção de pixels com códigos 12 e 16 nos quadrantes.

## Quadrantes

Em cada quadrante há particularidades próprias que parecem deixar brechas na lista de tiras que possam atrapalhar a extração. Isto não ocorre por que toda borda é considerada uma forma geométrica fechada. Este conceito, garante que nenhuma

tira ficará incompleta depois de se percorrer toda a borda.

O conceito adotado neste trabalho de forma geométrica fechada é: toda borda é fechada quando dado um pixel, é possível voltar ao mesmo ponto ao se percorrer a borda. Sendo que cada pixel obedece a distribuição estabelecida na Figura 3.10, não importando a forma que tenha a borda. Um exemplo disto pode ser visto nas letras da Figura 3.26.

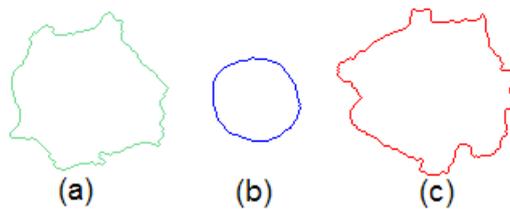


Fig. 3.26: Exemplo de Formas geométricas fechadas.

Observa-se que nas formas geométricas da Figura 3.26 é possível voltar ao mesmo ponto a partir de qualquer pixel da borda utilizando a configuração da Figura 3.10. A seguir é descrito a propriedade de cada quadrante.

### 1º Quadrante

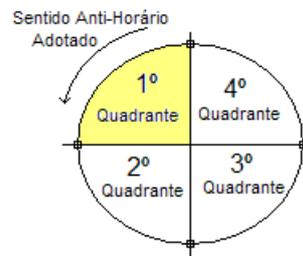


Fig. 3.27: Identificação do Primeiro quadrante.

Este quadrante tem a particularidade de que é preciso inserir novas tiras no início da lista, como se fosse a inclusão de um nó em uma pilha. Isto ocorre quando o valor

em  $y$  é menor que o valor em  $y_i$  da tira inicial da lista. A Figura 3.28 mostra um exemplo da inserção de tiras de uma linha 'm' da célula na lista de tiras na posição 'N' do vetor. A borda é percorrida a partir do pixel inicial e a marcação dos pixels em vermelho, representa a movimentação feita na borda da célula. Na letra (a), é mostrado a inserção do primeiro pixel da linha 'm', em que é armazenado o valor de  $y$  do pixel atual em  $y_i$  da tira (N1) na lista de tiras que se encontra na posição 'N' do vetor. Na letra (b), é mostrado a inserção do segundo pixel de borda da linha 'm', em que é armazenado o valor em  $y$  do segundo pixel em  $y_f$  da tira (N2). O valor em  $y$  do segundo pixel é menor que  $y_i$  da tira (N1), por isto, foi inserido em  $y_f$  da nova tira (N2). Na letra (c), é mostrado o encontro do terceiro pixel da linha 'm', em que o valor de  $y_i$  da tira (N2) está zerado. Neste caso, é inserido o valor de  $y$  em  $y_i$  da tira (N2). Na letra (d), é mostrado o encontro do pixel normal que fecha a tira (N1).

De acordo com a classificação de cada pixel encontrado, são realizados os procedimentos descritos a seguir.

a) **Pixel normal:** ao se encontrar um pixel normal (em que seu valor em  $y$  antecede o valor em  $y_i$  da tira inicial) com  $y_i$  diferente de zero, é preciso inserir uma nova tira antes da primeira, armazenando o valor de  $y$  em  $y_f$  da nova tira, com  $y_i$  zerado. Mas, se  $y_i$  da tira inicial tiver valor zero, isto quer dizer que o valor de  $y$  do pixel atual também deve ser armazenado em  $y_i$ .

A Figura 3.28 já mostrou um exemplo de inserção de pixels normais no vetor de tiras, pois todos os pixels utilizados nesta figura são normais.

b) **Pixel inicial de Linha:** no primeiro quadrante, uma linha contínua na borda sempre apresenta o código do vizinho 11 ou 13 quando começa e 5 ou 10 quando termina. Ao se encontrar um pixel inicial de linha e se na tira inicial da lista o valor de  $y_i$  não for zero, é preciso inserir uma nova tira no início da lista com o valor em  $y$  do pixel atual armazenado em  $y_f$ . Se for  $y_i$  zero, o valor em  $y$  é armazenado em  $y_i$

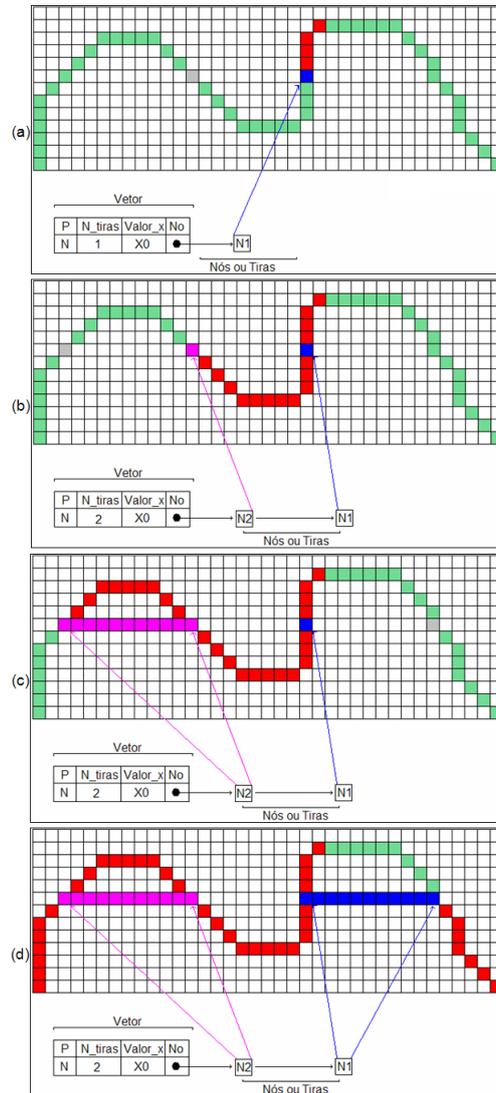


Fig. 3.28: Exemplo de inserção de pixels normais no primeiro quadrante.

da tira inicial da lista antes de se inserir a nova tira. Em seguida, percorre-se a lista da borda até que se encontre um pixel que tenha um código do vizinho 5 ou 10. Se for 5, armazena-se o valor de  $y$  em  $y_i$  da nova tira inserida. Se for 10, armazena-se o valor de  $y$  do pixel atual em  $y_i$  da tira inserida atualmente. Em seguida, insere-se uma nova tira no início da lista e armazena-se o valor de  $y$  do pixel atual em  $y_f$ .

A seguir, serão mostrados exemplos para os dois tipos de linhas existentes no primeiro quadrante. No exemplo 1, as linhas que começam com código 13 terminam com 5 ou 10. No exemplo 2, as linhas que começam com 11 e terminam com 5 ou

10.

Exemplo 1.

A Figura 3.29 mostra um exemplo de inserção do pixel inicial de linha que tem código do vizinho igual a 13. Nas letras (a) e (b), são feitas os armazenamentos dos pixels normais de uma linha da célula, sendo que na letra (b) ocorre também à identificação do primeiro pixel inicial de linha. Este pixel foi marcado em cinza. Na letra (c), o valor em  $y$  do pixel atual é armazenado em  $y_i$  da tira (S1) inserida no início da lista, pois, o mesmo apresenta '0' na variável  $y_i$ . Na letra (d), é inserido mais uma tira (S2) no início da lista e é armazenado o valor de  $y$  do pixel atual em  $y_f$  desta tira, em seguida, percorre-se a lista de bordas até se encontrar um pixel de final de linha com código 5 ou 10. Neste exemplo, o código do pixel achado foi 10, (marcado de cinza na letra (d)). Na letra (e), o valor de  $y$  do pixel final é armazenado em  $y_i$  da tira (S2). Insere-se outra tira (S3) no início da lista e armazena-se o valor de  $y$  do pixel final de linha em  $y_f$  da tira. Mas, se o código do pixel achado na letra (d) fosse o 5 ao invés do 10, simplesmente se armazenaria o valor de  $y$  deste pixel em  $y_i$  da última tira inserida na lista, como pode ser visto na Figura 3.30(a) e (b).

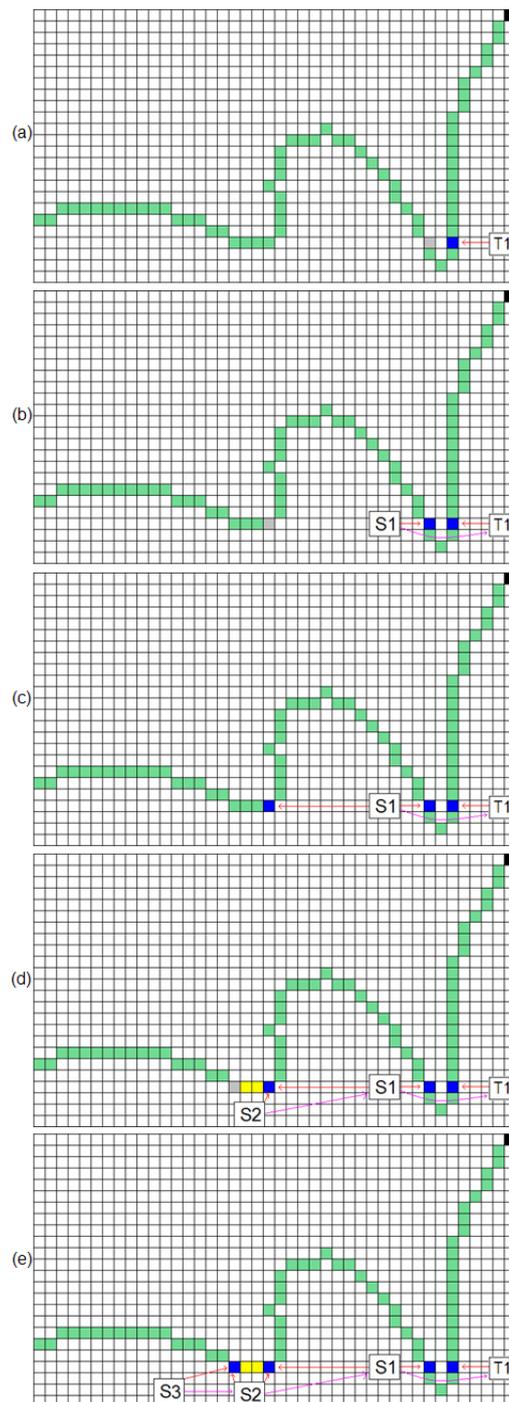


Fig. 3.29: Exemplo de inserção do pixel de linha.

Exemplo 2.

A Figura 3.31 mostra um exemplo de inserção do pixel inicial de linha que tem código do vizinho igual a 11. Na letra (a), é feito o armazenamento do pixel normal

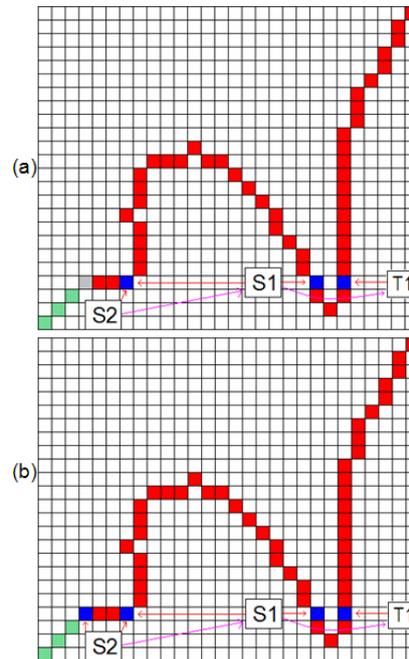


Fig. 3.30: Exemplo da finalização da tira S2.

da linha da célula, além da identificação do pixel inicial de linha (pixel foi marcado em cinza). Na letra (b), o valor de  $y$  do pixel atual é armazenado primeiramente em  $y_i$  da tira (T2) inserida no início da lista de tiras. Na letra (c), a borda é percorrida até que se encontre um pixel de final de linha com código 5 ou 10. Neste exemplo, o código do pixel achado foi o 5 (marcado em cinza). Na letra (d), o valor de  $y$  do pixel final encontrado, é armazenado em  $y_i$  da tira (T2). Se o código do pixel achado em (d) fosse o 10 ao invés do 5, como mostrado na Figura 3.32(a), então armazenaria-se o valor de  $y$  deste pixel em  $y_i$  da última tira inserida em seguida, inseria-se uma outra nova tira no início da lista e armazenaria-se o valor de  $y$  do pixel atual em  $y_f$ , como pode ser visto na Figura 3.32(b).

c) **Pixel Brotoeja**: seu código do vizinho é 16. Quando se vai inserir este pixel na lista de tiras, é preciso inserir uma tira no início da lista com o valor de  $y$  do pixel atual em  $y_i$  e em  $y_f$  da nova tira.

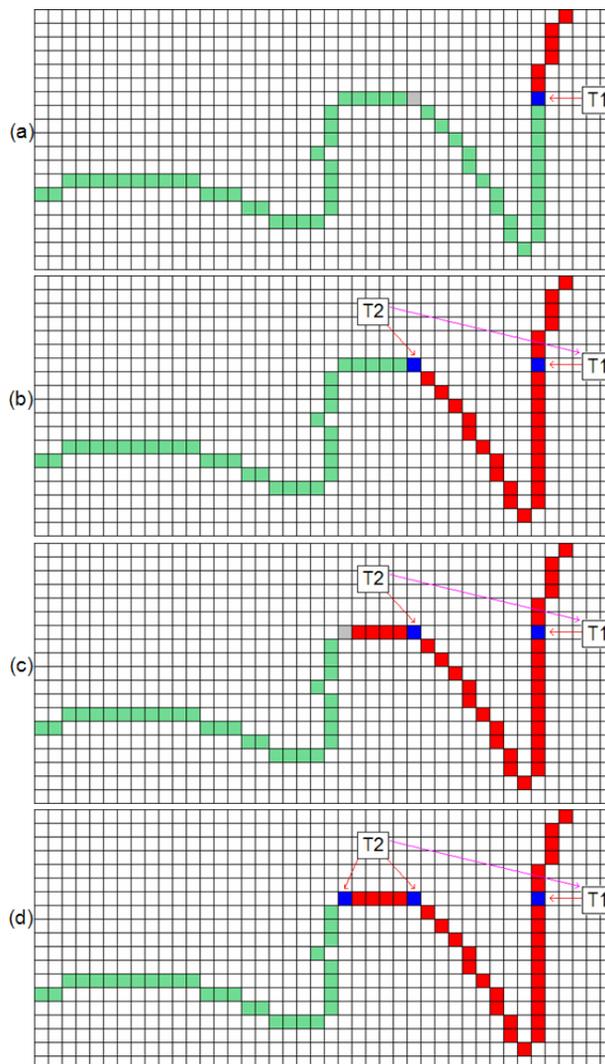


Fig. 3.31: Exemplo de inserção do pixel de linha com código 11.

A Figura 3.33 mostra um exemplo da inserção do pixel brotoeja na lista de tiras. A letra (a) mostra a inserção de um pixel normal na mesma linha em que ocorre à identificação do pixel brotoeja (marcado em cinza). Na letra (b), é feita a inserção da nova tira (S1) no início da lista com o valor de  $y$  do pixel atual armazenado em  $y_i$  e em  $y_f$ .

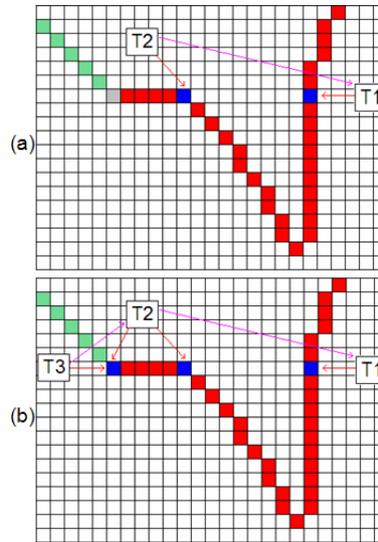


Fig. 3.32: Exemplo de inserção do pixel de linha com código final 10.

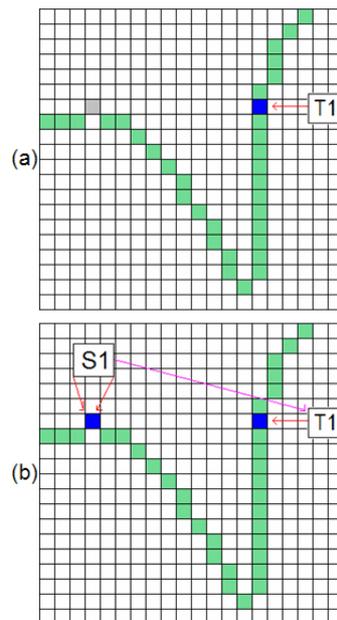


Fig. 3.33: Exemplo de inserção do pixel brotoeja.

## 2º e 3º Quadrantes

Estes quadrantes têm como características o fato de que ao se terminar de percorrê-los todas as tiras terem sido inseridas de forma completa, pois, as tiras são inseridas sempre no final da lista. Inserir uma tira de forma completa quer dizer que

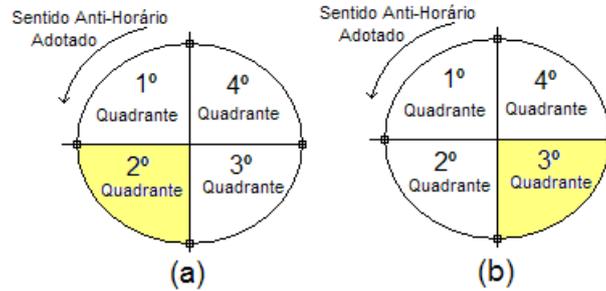


Fig. 3.34: Identificação do 2º quadrante (a) e do 3º quadrante (b).

não ocorre o caso de tiras que só serão finalizadas no quarto quadrante como ocorre com as tiras do primeiro quadrante.

Uma característica marcante do segundo e do terceiro quadrante é que o valor de  $y$  do pixel atual será sempre maior que qualquer valor armazenado em  $y_i$  ou  $y_f$  das tiras já inseridas.

Serão vistos a seguir, os procedimentos de inclusão dos pixels da borda na lista de tiras de acordo com as classificações dos pixels.

a) **Pixels normais:** são inseridos de forma seqüencial no final de cada lista de tiras.

A Figura 3.35 mostra um exemplo, onde em (a) é mostrado a identificação do pixel normal a ser inserido na lista (marcado em cinza). Em (b), é mostrado a inserção do primeiro pixel na lista, pois, o valor de  $y$  do pixel atual é armazenado em  $y_i$  da tira (T1). Nesta mesma figura, é mostrada a identificação do segundo pixel a ser inserido na lista (marcado em cinza). Em (c), é mostrado a inserção do segundo pixel na lista, pois, o valor de  $y$  do segundo pixel em  $y_f$  é armazenado da tira (T1). Nesta mesma figura é mostrada a identificação do terceiro pixel a ser inserido na lista. Em (d), é mostrado a inserção do terceiro pixel na lista de tiras, pois, o valor de  $y$  do pixel atual é armazenado em  $y_i$  da tira (S1), que foi inserida após a tira (T1) na lista. Nesta mesma figura, aparece a identificação do quarto pixel a ser inserido. Em (e), é mostrado a inserção do quarto pixel na lista de tiras, ao armazenar o valor de  $y$  do pixel atual em  $y_f$  da tira (S1).

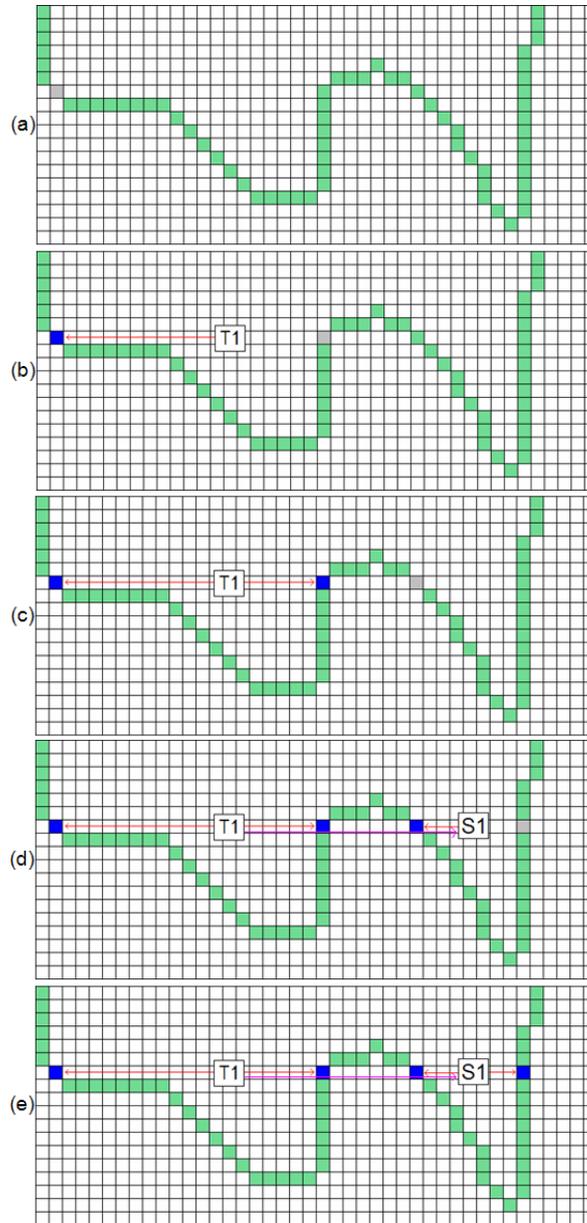


Fig. 3.35: Exemplo da inserção dos pixels normais no 2º e no 3º quadrantes.

b) **Pixel inicial de linha:** No segundo e no terceiro quadrante os pixels iniciais de linha começam com os códigos 5 ou 10 e termina com 11 ou 13.

A seguir, serão mostrados quatro exemplos de inserção de linhas.

Exemplo 1

A inserção da linha contínua que começa com o código 10 e termina com o código 11

é mostrada na Figura 3.36. Na letra (a), foi identificado o pixel de código 5 (marcado em cinza). Na letra (b), é mostrado que o pixel inicial de linha foi inserido na lista de tiras e que a borda foi percorrida até se encontrar um pixel com código 11 ou 13. Neste caso, foi achado um pixel com código 11. Na letra (c), o valor de  $y$  do pixel final de linha foi inserido em  $y_f$  da tira (T1) e acrescentou-se uma nova tira (S1) em que o valor de  $y$  foi armazenado em  $y_i$  da nova tira. Também, foi identificado um novo pixel a ser inserido na lista de tira (marcado em cinza). Na letra (d), a tira (S1) foi finalizada com o armazenamento do valor de  $y$  em  $y_f$ .

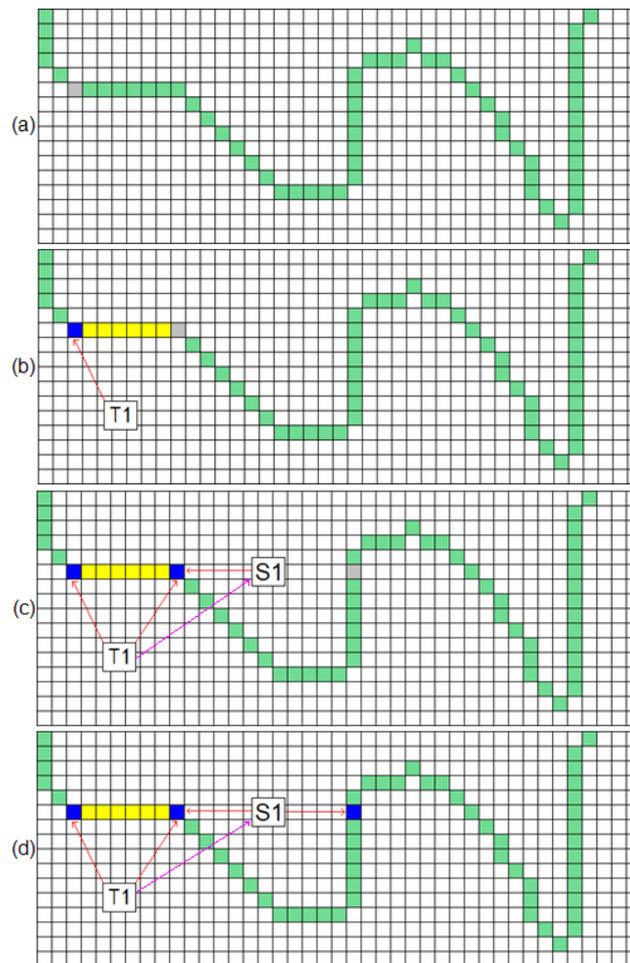


Fig. 3.36: Exemplo de inserção de linha que começa com o código 10 e finaliza com 11.

## Exemplo 2

Na Figura 3.37 é mostrado um exemplo da inserção da linha contínua que começa com o código 10 e termina com o código 13. Na letra (a), foi identificado o pixel inicial de linha com código 10 (marcado em cinza). Na letra (b), este pixel foi armazenado na lista de tiras. Isto foi feito armazenando-se o valor de  $y$  em  $y_i$  da tira (T1) armazenada no início da lista. Em seguida, a borda foi percorrida até se encontrar um pixel de final de linha com o código 11 ou 13. Neste caso, foi encontrado o pixel de código 13. Na letra (c), o pixel encontrado foi inserido na tira.

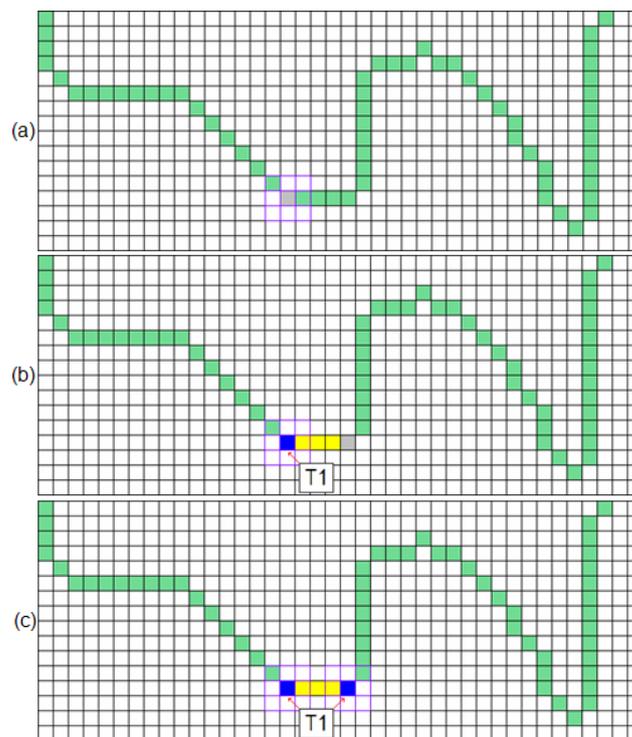


Fig. 3.37: Exemplo de inserção de linha que começa com o código 10 e finaliza com 13.

## Exemplo 3

Na Figura 3.38, é mostrado um exemplo de inserção da linha contínua que começa com o código 5 e termina com o código 11. Em (a), é mostrado a inserção de um pixel normal na lista de tiras bem como a identificação do próximo pixel (marcado

em cinza) a ser inserido na lista. Em (b), o valor de  $y$  é armazenado em  $y_f$  da tira (T1) e também em  $y_i$  da tira (T2). Em seguida, a borda foi percorrida até encontrar um pixel que tenha código do vizinho igual a 11 ou 13. Neste caso, foi achado o pixel com código 11 (marcado em cinza). Em (c), o valor de  $y$  é armazenado em  $y_f$  da tira (T2) bem como em  $y_i$  da nova tira (T3). Foi identificado também um novo pixel normal (marcado em cinza). Em (d), o valor de  $y$  do último pixel encontrado foi armazenado em  $y_f$  da tira (T3), finalizando assim a tira da linha da célula.

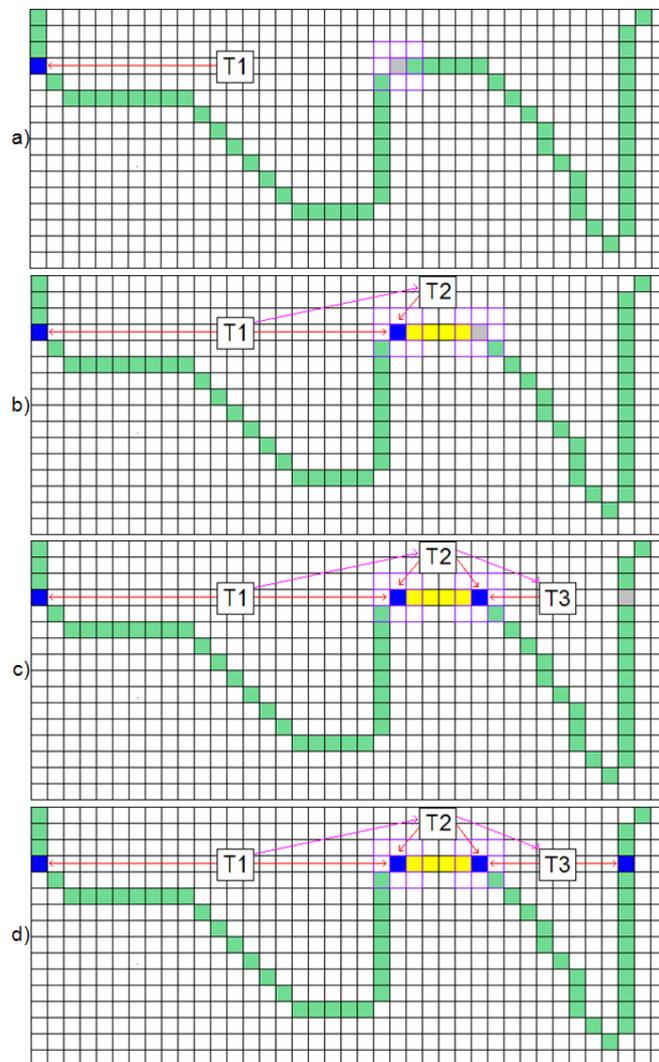


Fig. 3.38: Exemplo de inserção da linha contínua que começa com o código 5 e termina com 11.

## Exemplo 4

Na Figura 3.39, é mostrado um exemplo da inserção da linha continua que começa com o código 5 e termina com o código 13. Na letra (a), é armazenado o primeiro pixel normal da tira em (T1) e identificado o pixel inicial de linha com código 5 (marcado em cinza). Na letra (b), o valor de y do pixel identificado é armazenado em yf da tira (T1). A borda foi percorrida até encontrar um pixel que tenha código de vizinho igual a 11 ou 13. Neste caso, foi achado o pixel com código 13. Na letra (c), é inserida uma nova tira no final da lista em que o valor de yf da tira (T1) é armazenado em yi da tira (S1) e o valor de y do pixel atual é armazenado em yf da tira (S1).

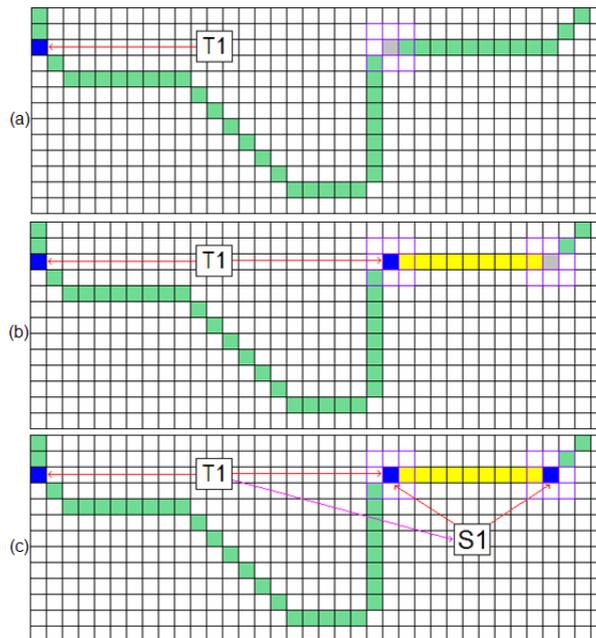


Fig. 3.39: Exemplo de inserção da linha continua que começa com o código 5 e termina com 13.

c) **Pixel brotoeja**: o código do vizinho deste pixel é 12 no segundo e no terceiro quadrante. A inserção deste pixel na lista de tiras ocorre em duas situações: a primeira, é quando este pixel vai iniciar a lista de tiras e a segunda, é quando vai

ser inserido no final da lista. Nas duas situações, a maneira de se inserir é a mesma. É inserida uma tira (seja no início ou no final da lista) em que o valor de  $y$  do pixel atual é armazenado tanto em  $y_i$  e em  $y_f$  da tira.

A Figura 3.40 mostra um exemplo da inserção de um pixel brotoeja, quando esta inserção inicia a lista de tiras, em que o valor de  $y$  do pixel é armazenado tanto em  $y_i$  com em  $y_f$ .

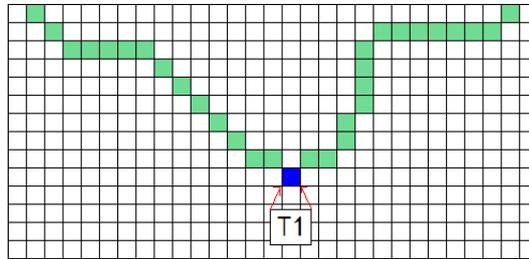


Fig. 3.40: Exemplo de inserção do pixel brotoeja no início da lista de tiras.

A Figura 3.41 mostra um exemplo de inserção do pixel brotoeja quando ele é inserido no final da lista de tiras. Em (a), é mostrada a inserção de pixels normais na tira (T1). Em (b), é feita a inserção do pixel brotoeja no final da lista de tiras.

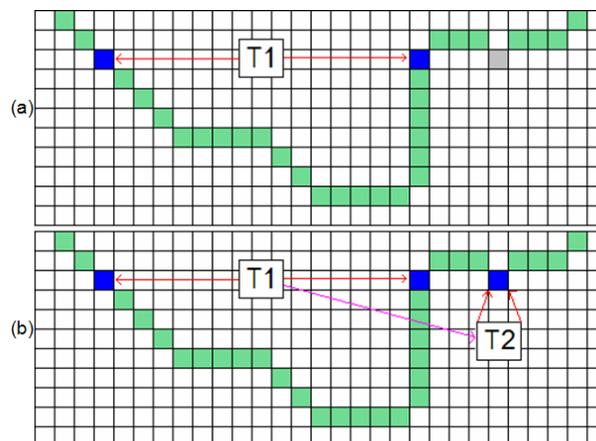


Fig. 3.41: Exemplo de inserção do pixel brotoeja no final da lista de tiras.

## 4º Quadrante

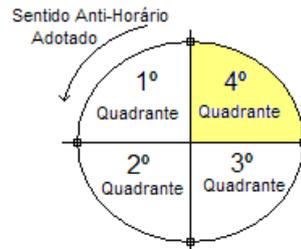


Fig. 3.42: Identificação do 4º quadrante.

O quarto quadrante tem a característica de armazenar os pixels de borda no meio de tiras já inserida na lista. Isto ocorre quando o valor em  $y$  de um pixel está entre os valores de  $y_i$  e  $y_f$  de uma tira.

A Figura 3.43 mostra um exemplo da inserção de tiras no quarto quadrante. Na letra (a), a borda é percorrida até encontrar o primeiro pixel que foi inserido na tira (T1). Em (b), a borda é percorrida até encontrar o pixel final relativo da tira (T1), pois, o mesmo não é o pixel final da primeira tira. Este pixel é inserido na tira (T1). Em (c), a borda continua a ser percorrida até encontrar o pixel que tenha o valor de  $y$  que fica entre  $y_i$  e  $y_f$  da tira (T1). Na letra (d), o problema é resolvido com a inserção de uma nova tira (T2). Nesta tira, é armazenado o valor de  $y_f$  da tira (T1) em  $y_f$  da mesma em seguida, o valor de  $y$  do pixel atual é armazenado em  $y_i$  da tira (T2). A variável  $y_f$  da tira (T1) é então zerada. Na letra (e), a borda foi percorrida até encontrar o pixel que finaliza a tira (T1), sendo que esta tira foi finalizada na letra (f).

Os pixels da borda são armazenados de acordo com as classificações dos mesmos. Isto é visto a seguir:

a) **Pixel normal**: quando o valor de  $y$  de um pixel normal está no meio dos valores de  $y_i$  e  $y_f$  de uma tira, é necessário inserir uma nova tira após esta, armazenando-se

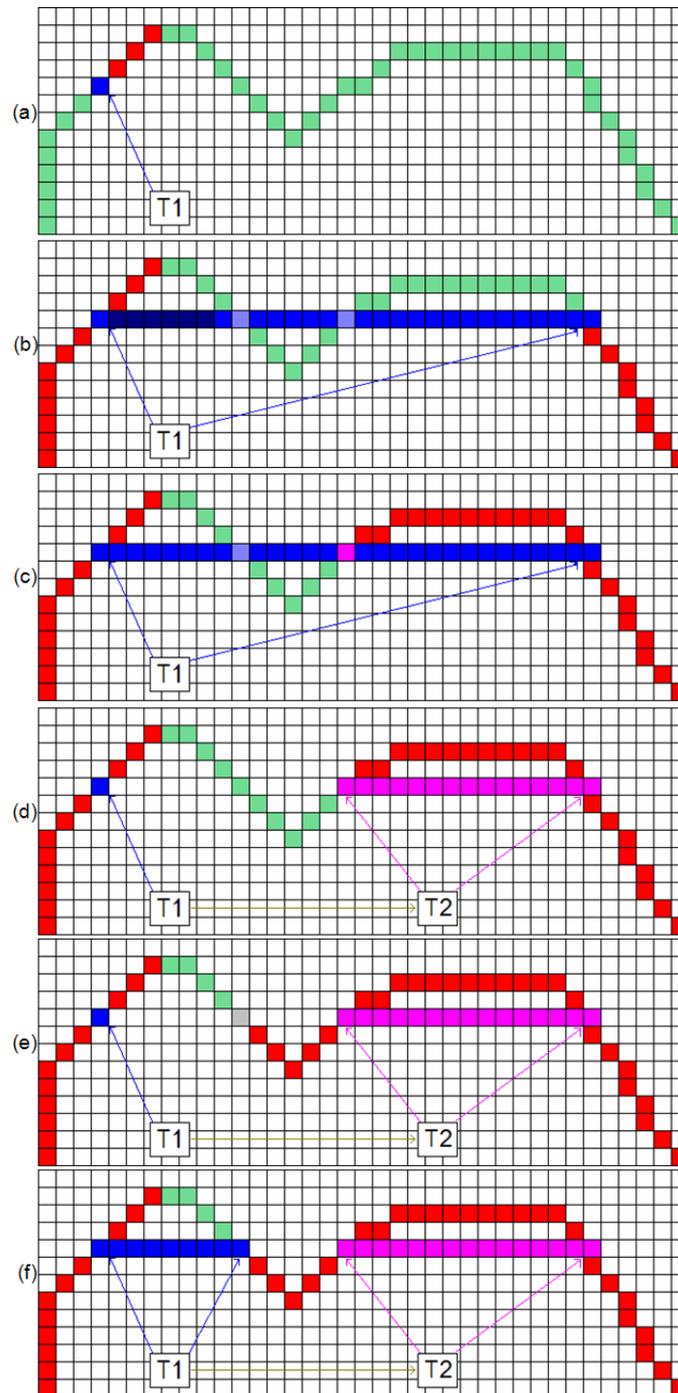


Fig. 3.43: Exemplo da inserção de tiras no quarto quadrante.

o valor de  $y_f$  da tira anterior em  $y_f$  da tira atual. Em seguida, armazena-se o valor de  $y$  do pixel atual em  $y_i$  da tira atual e zera-se o valor de  $y_f$  da tira anterior. Um exemplo disto esta na Figura 3.43.

b) **Pixel inicial de linha:** no quarto quadrante, uma linha contínua na borda, sempre apresenta o código do vizinho 11 ou 13 quando começa e terminam com o código 5 ou 10. A inserção do pixel inicial de linha na lista é parecida com a inserção feita no primeiro quadrante. O que muda é que, a inserção é feita no meio de alguma tira inserida na lista. A seguir, serão mostrados dois exemplos de inserção de linha no quarto quadrante: O primeiro, será a inserção de uma linha que começa com o código 13 e termina com 5 ou 10. O segundo, a inserção de uma linha que começa com o código 11 e termina com 5 ou 10.

Exemplo 1.

A Figura 3.44 mostra um exemplo da inserção do pixel inicial de linha no quarto quadrante. Na letra (a), a borda é percorrida até a identificação do primeiro pixel normal a ser inserido na lista (marcado em cinza). Na letra (b), o valor de  $y$  do pixel atual é armazenado em  $y_i$  da tira (T1) e a borda é percorrida até a identificação do segundo pixel normal a ser inserido na lista. Em (c), o valor de  $y$  do segundo pixel encontrado é armazenado em  $y_f$  da tira (T1) e a borda é percorrida até que o pixel inicial de linha seja encontrado. Este pixel possui o valor de  $y$  entre os valores de  $y_i$  e  $y_f$  da tira (T1). Em (d), foi inserida na lista uma tira (T2) após a tira (T1) em que o valor de  $y_f$  da tira (T1) foi copiado para  $y_f$  da tira (T2). Em seguida, zerou-se o valor de  $y_f$  da tira (T1). Em (e), foi armazenado o valor de  $y$  do pixel atual em  $y_i$  da tira (T2) e a borda foi percorrida até encontrar um pixel que tenha código 5 ou 10. Neste exemplo, foi achado o código 5 (marcado em cinza). Na letra (f), foi inserida uma nova tira (T3), após a tira (T1). O valor de  $y_i$  da tira (T2) foi armazenado em  $y_f$  da tira (T3). Em (g), o valor de  $y$  do pixel atual foi armazenado em  $y_i$  da tira (T3) e a borda foi percorrida até se identificar o pixel normal que finaliza a tira (T1). Na letra (h), a tira (T1) foi finalizada ao se armazenar o valor de  $y$  do pixel atual em  $y_f$  da tira (T1). Se na Figura 3.44(e), fosse encontrado um pixel com código 10 ao invés de 5, como pode ser verificado na Figura 3.45(a), seria

inserida uma nova tira (T3) na lista, e o valor em  $y_i$  da tira (T2) seria armazenado em  $y_f$  da tira (T3). O valor de  $y$  do pixel atual seria armazenado em  $y_f$  da tira (T1) e em  $y_i$  da tira (T2), finalizando assim, o armazenamento das tiras da linha da célula, isto pode ser visto na Figura 3.45(b).

Exemplo 2.

A Figura 3.46 mostra um exemplo da inserção do pixel inicial de linha no quarto quadrante. Na letra (a), a borda é percorrida até a identificação do primeiro pixel normal a ser inserido na lista (marcado em cinza). Na letra (b), o valor de  $y$  do pixel atual é armazenado em  $y_i$  da tira (T1) e a borda é percorrida até a identificação do segundo pixel normal a ser inserido na lista. Em (c), o valor de  $y$  do segundo pixel encontrado é armazenado em  $y_f$  da tira (T1) e a borda é percorrida até que o terceiro pixel normal seja encontrado. Este pixel possui o valor de  $y$  entre os valores de  $y_i$  e  $y_f$  da tira (T1). Em (d), foi inserida na lista uma tira (T2) após a tira (T1) em que o valor de  $y_f$  da tira (T1) foi copiado para  $y_f$  da tira (T2). Em seguida, zerou-se o valor de  $y_f$  da tira (T1). Em (e), foi armazenado o valor de  $y$  do pixel atual em  $y_i$  da tira (T2) e a borda foi percorrida até encontrar um pixel inicial de linha. Na letra (f), foi inserida uma nova tira (T3), após a tira (T1). O valor de  $y$  do pixel atual foi armazenado em  $y_f$  da tira (T3) e a borda foi percorrida até se achar o pixel que tenha código 5 ou 10. Neste exemplo, foi achado o código 5 (marcado em cinza). Em (g), o valor de  $y$  do pixel atual foi armazenado em  $y_i$  da tira (T3) e a borda foi percorrida até se identificar o pixel normal que finaliza a tira (T1). Na letra (h), a tira (T1) foi finalizada ao se armazenar o valor de  $y$  do pixel atual em  $y_f$  da tira (T1). Se na Figura 3.46(f), fosse encontrado um pixel com código 10 ao invés de 5, como pode ser verificado na Figura 3.47(a), o valor de  $y$  do pixel atual seria armazenado em  $y_f$  da tira (T1) e em  $y_i$  da tira (T3), finalizando assim, o armazenamento das tiras da linha da célula, isto pode ser visto na Figura 3.47(b).

c) **Pixel Brotoeja:** tem código do vizinho 16. Para cadastrar este pixel na lista,



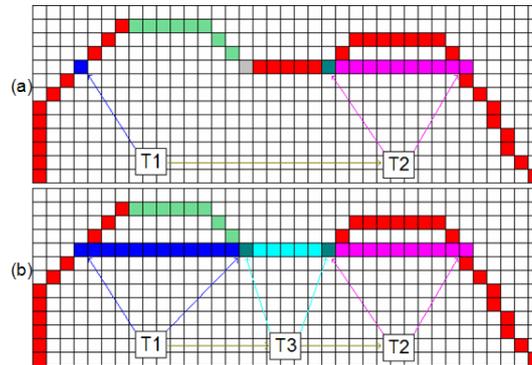


Fig. 3.45: Exemplo da inserção do pixel inicial da linha no quarto quadrante com código 10.

Na figura 3.48, é mostrado um exemplo de inserção do pixel brotoeja no quarto quadrante. Na letra (a), a borda foi percorrida até a identificação do primeiro pixel a ser inserido na lista. Na letra (b), o valor de y do pixel atual foi inserido em yi da tira (T1) e a borda foi percorrida até ser identificado o pixel brotoeja (marcado em cinza). Na letra (c), foi inserida uma segunda tira (T2) com o valor de y do pixel brotoeja armazenado em yi e em yf da tira (T2). A borda foi percorrida novamente até se identificar o pixel normal que finaliza a tira (T1). Em (d), a tira (T1) foi finalizada com o armazenamento do valor de y do pixel atual em yf da tira (T1).

### Realização da Extração

Nesta subseção, será abordada a realização da extração de imagens propriamente dita, pois, os passos anteriores criaram condições para que esta extração fosse feita.

Para a extração da célula, foi criada uma imagem em branco (matriz do tipo inteiro de 8 bits) com as mesmas dimensões (altura, largura) da célula a ser extraída. Em seguida, esta imagem é preenchida de forma segura, com os dados dos segmentos das linhas das células armazenados no vetor de tiras. A célula não é apagada da imagem original e sim sua respectiva borda no mapa de bits, evitando assim que a mesma não seja extraída novamente. Os cálculos da altura e da largura da célula

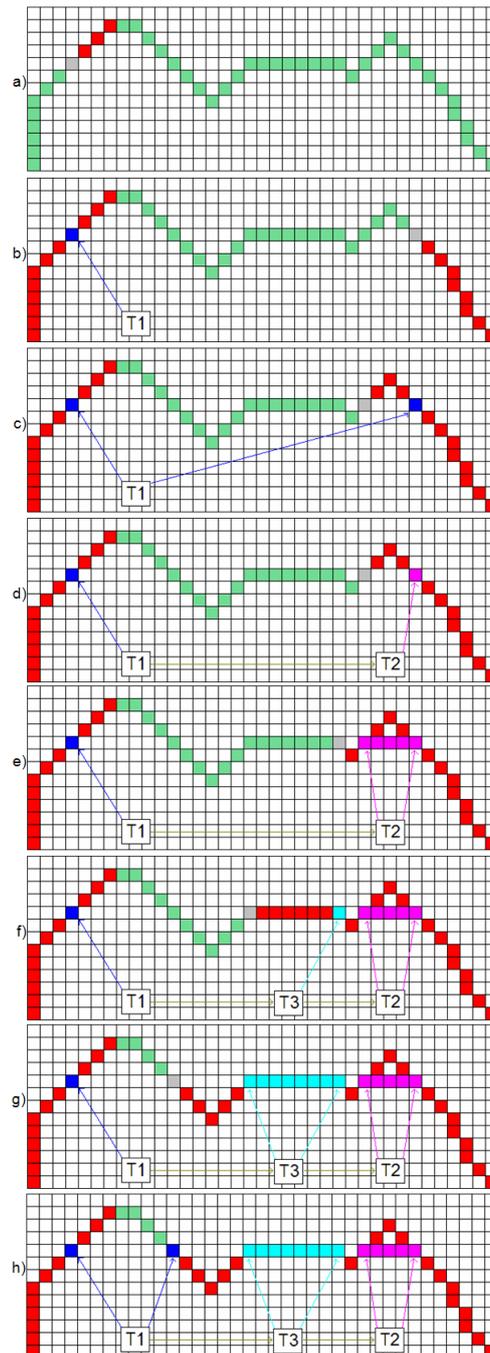


Fig. 3.46: Exemplo da inserção do pixel inicial da linha no quarto quadrante com o código 11.

são realizados com base nos valores das variáveis maiorx, menorx, maiory e menory.

O cálculo é realizado conforme a equações 3.5 e 3.6.

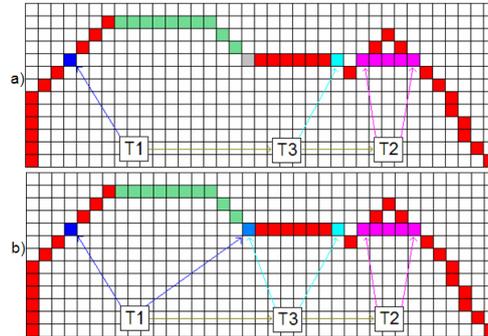


Fig. 3.47: Exemplo da inserção do pixel inicial da linha no quarto quadrante que termina com código 10.

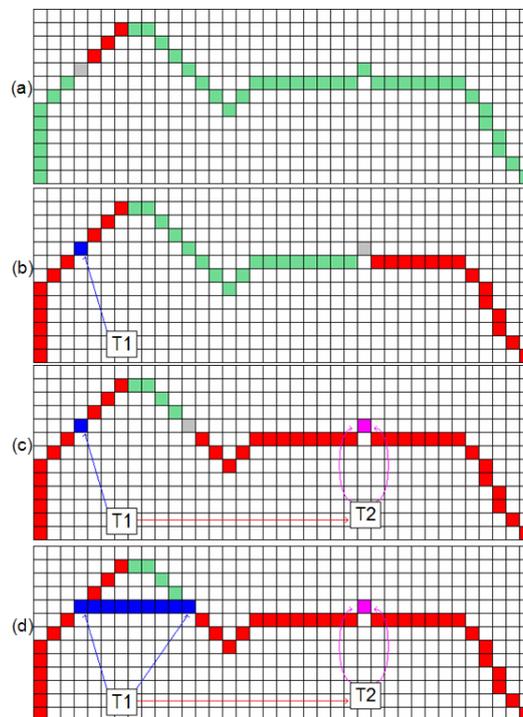


Fig. 3.48: Exemplo de inserção do pixel brotoeja no quarto quadrante.

$$Altura = (maior_y - menor_y) + 1; \quad (3.5)$$

$$Largura = (maior_x - menor_x) + 1; \quad (3.6)$$

A imagem criada tem as dimensões:  $Altura \times Largura \times 3$ . O tamanho desta imagem é três vezes superior ao tamanho da célula para que a mesma possa comportar

as componentes RGB da célula contida na imagem original.

Para o preenchimento da imagem em branco, é feita para a mesma, a cópia das cores de todos os pixels da célula, mantendo-se a mesma ordem dos pixels na nova imagem. Esta ordem é mantida, calculando-se a nova posição dos pixels na nova imagem de acordo com as equações 3.7 e 3.8.

$$x_n = (x' - menorx) + 1. \quad (3.7)$$

$$y_n = (y' - menory) + 1. \quad (3.8)$$

Onde,  $x_n$  e  $y_n$  são a nova posição do pixel nos eixos x e y da nova imagem respectivamente. e são a posição pixel da célula no eixo x e y respectivamente. Estas equações foram criadas com base na disposição dos pixels na imagem e pelo limites da célula dentro da imagem, como pode ser observado na Figura 3.49.

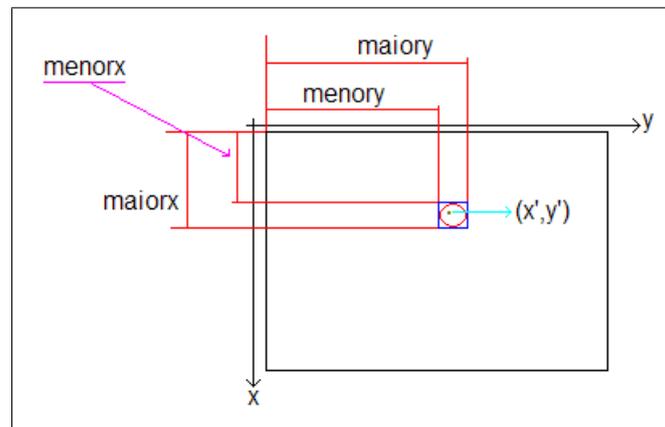


Fig. 3.49: Imagem base para o cálculo das novas posições x e y.

À medida que cada pixel é lido, para um nível de ocorrência de R, G e B, é armazenada a posição x e y correspondente e é calculado um histograma para cada componente. Este histograma bem como as coordenadas x e y serão posteriormente utilizados para a criação do spatioqram.

A Figura 3.50 mostra a estrutura do vetor que comporta o histograma e as

posições dos pixels. Pode se notar que a posição dos pixels é armazenada em uma lista simplesmente encadeada pela variável 'No'. A variável 'P' indica a posição do histograma, 'QTDE' armazena a quantidade de vezes que determinada cor indicada por 'P' foi contada. Este vetor é chamado de Vetor do Histograma. Cada nó, armazena somente a posição (x,y) de cada cor da célula.

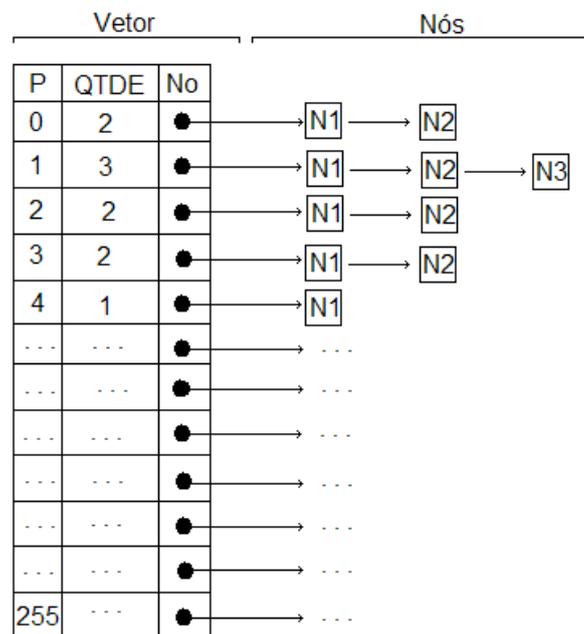


Fig. 3.50: Vetor do Histograma.

É mostrado a seguir, um exemplo para se ter uma melhor compreensão do que foi explicado.

Exemplo:

Dado um pixel  $P_1$  com coordenadas (45,81), tendo os valores RGB de (100,200,200), que ainda não foi computado no histograma, como pode ser observado nas letras (a), (b) e (c) da Figura 3.51. Ao se computar este pixel, as variáveis 'QTDE' de cada posição indicada pelo nível de cada componente RGB são incrementadas de 1, bem como, é inserido um nó na lista simplesmente encadeada de cada vetor, contendo o valor (45,81). O resultado é mostrado nas letras (d), (e) e (f) da Figura 3.51.

Calcula-se a matriz de covariância para cada nível das componentes RGB, utilizando-

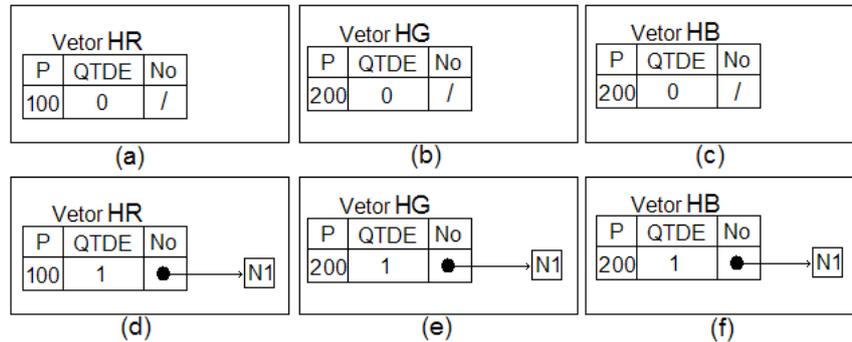


Fig. 3.51: Exemplo do funcionamento do vetor do histograma.

se o vetor de médias. O vetor de médias é obtido pela média das coordenadas em  $x$  e em  $y$ , que foram armazenados na lista simplesmente encadeada com o objetivo de verificar o quanto agrupado estão os pixels com as cores determinadas.

A diagonal principal da equação 2.9 da Subseção 2.3.1 do Capítulo 2 armazena a variância nos eixos  $x$  e  $y$  (posições  $[1,1]$  e  $[2,2]$ ) para cada nível do histograma. A diagonal secundária armazena a covariância entre os eixos  $x$  e  $y$  (posições  $[1,2]$  e  $[2,1]$ ) para cada nível do histograma. Os valores das posições  $[1,2]$  e  $[2,1]$  são iguais.

A Figura 3.52 mostra o resultado do spatiogram calculado a partir do histograma de uma célula, para uma das componentes RGB, onde 'P', indica a posição do histograma, 'QTDE' a quantidade de vezes que uma determinada cor foi contada, 'mx' a média dos valores no eixo  $x$ , 'my' a média dos valores no eixo  $y$ , 'var\_11' a variância das coordenadas do eixo  $x$  na posição  $[1,1]$ , 'var\_22' a variância das coordenadas do eixo  $y$  na posição  $[2,2]$  e 'covar' a covariância das coordenadas entre os eixos  $x$  e  $y$ .

Após a geração do spatiogram, para cada componente RGB de uma célula, os mesmos são armazenados em uma tabela do banco de dados.

Vetor Spatiogram

P	QTDE	mx	my	var_11	var_22	covar
0	2	11.5	34.25	49.67	14.92	26.5
1	3	15	35	128	18	48
2	2	16.2	30.8	238.7	37.7	17.55
3	2	18.5	36.67	24.3	1.06	2.8
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...
255	...	...	...	...	...	...

Fig. 3.52: Exemplo do Vetor que armazena o spatiogram.

### Comentários sobre a extração.

Foram mostrados os passos para a realização da extração de células da imagem de sangue. Para cada célula extraída, é gerada uma nova imagem da célula e são gerados spatiograms para cada uma dos componentes RGB da célula extraída.

### 3.3 Classificação

Devido ao fato de ter sido optado por imagens da internet para trabalhar nesta dissertação, não foi possível obter uma grande quantidade e nem variedade de imagens para cada patologia do sangue, além de as imagens apresentarem resoluções diferentes. Isto afetou de forma significativa os resultados da pesquisa quanto a precisão da classificação das patologias do sangue. Apesar disto, optou-se por testar as técnicas de similaridades descritas no Capítulo 2 para ver se estas funções de similaridade podem mostrar se uma imagem é similar a outra ou não.

Os testes foram realizados com a utilização de leucócitos, pois as hemácias possuem um alto grau de similaridade, apresentando muitos falsos positivos através dos métodos utilizados neste trabalho.

A Figura 3.53, mostra os leucócitos que foram selecionados a partir das 7 Figuras que apresentaram boas resoluções de imagem.

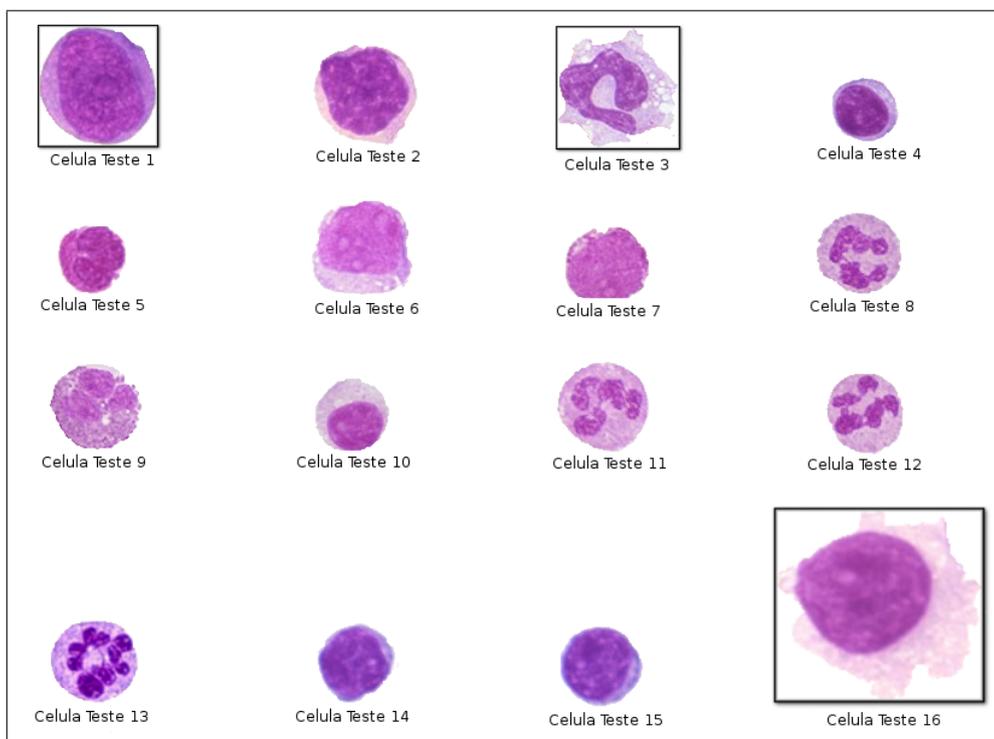


Fig. 3.53: Leucócitos selecionados para os testes de similaridade.

Para cada Leucócito da Figura 3.53, foram realizados os cálculos da medida do grau de similaridade. Primeiramente foi realizada a medida com o próprio Leucócito em seguida, a medida com os outros. Por exemplo, calculou-se o grau de similaridade da Célula Teste 1 com ela mesma e em seguida, foi feito o calculo da medida de similaridade da mesma com as outras células.

Como há 16 células na Figura 3.53, foram executados 16 testes para cada Leucócito.

As informações das células mostradas na Figura 3.53 foram colocadas na Tabela 3.3, para uma melhor compreensão. A coluna ‘Número das Células Teste’ indica o número associado a cada célula da Figura, por exemplo, os números 3 e 4 que aparecem na linha 1 indicam as Células Teste 3 e 4 da Figura 3.53. A coluna ‘Patologias’ indica o nome das patologias.

IT	Patologias	Número das Células Teste
1	Normal	3,4
2	Anemias Microcíticas	-
3	Desordem MyeloProliferativo	5, 6 e 7
4	Anemia Normocíticas	-
5	Leucemias Agudas	1, 2
6	Anemias Hemolíticas	-
7	Anemias Macroscíticas	8, 9, 10, 11, 12 e 13
8	Desordem Limfoproliferativo	14, 15 e 16

Tab. 3.3: Tabela de associação das células da Figura 3.53 com as patologias.

Quando foi calculado o spatiogram das células da Figura 3.53, foi atribuído um código para cada uma delas no banco de dados, estes códigos são mostrados na Tabela 3.4. Os mesmos foram criados a partir de uma numeração seqüencial dada a cada célula extraída das imagens.

Foram criadas duas tabelas para armazenar os resultados dados dos testes. A primeira (Resultado1) armazenou os dados dos testes utilizando as técnicas de similaridades propostas por Birchfield [3]. A segunda (Resultado2) armazenou os resultados das técnicas que medem a distância entre matrizes de covariância e veto-

Nome das Células	Código
Célula Teste 1	21
Célula Teste 2	31
Célula Teste 3	93
Célula Teste 4	115
Célula Teste 5	192
Célula Teste 6	199
Célula Teste 7	230
Célula Teste 8	250
Célula Teste 9	262
Célula Teste 10	270
Célula Teste 11	277
Célula Teste 12	279
Célula Teste 13	312
Célula Teste 14	373
Célula Teste 15	382
Célula Teste 16	447

Tab. 3.4: Tabela com os códigos dados aos Leucócitos da Figura 50.

res de média dos spatiograms.

As técnicas que medem a distância entre matrizes de covariância e vetores de média dos spatiograms como: Mahalanobis, Bhattacharyya e Jeffries-Matusita, apresentavam um grau de similiaridade zero em cada teste, nas três bandas R, G e B, quando a imagem era comparada consigo mesma. O que não ocorreu no resultado das técnicas de similaridades propostas por Birchfield [3]. A Figura 3.54 mostra os resultados do calculo do grau de similaridade para o Leucócito de número 21. Na primeira linha, pode ser observado que o resultado do cálculo para o leucócito 21 utilizando as técnicas de Bhattacharyya (Coluna B), Mahalanobis (M) e Jeffries-Matusita (JM). As outras linhas mostram o resultado do cálculo do grau de similaridade deste leucócito para os outros. Sendo que estes resultados estão ordenados pela coluna ‘ResBM.ID’.

Os resultados das técnicas propostas por Birchfield [3], mostraram-se muito confusos para se tomar qualquer decisão.

ResBM_ID	TesteBM_ID	Celula_ID	Banda	B	M	JM
<input checked="" type="checkbox"/>	1	21	R	0	0	0
<input type="checkbox"/>	7	93	R	1585370.25	7617.044	96.345
<input type="checkbox"/>	4	31	R	488931.625	4881.236	105.909
<input type="checkbox"/>	43	382	R	460067.625	4993.766	109.361
<input type="checkbox"/>	46	447	R	689960.938	4030.127	116.674
<input type="checkbox"/>	31	277	R	1155749.625	9353.474	117.429
<input type="checkbox"/>	37	312	R	308768.062	3872.294	120.153
<input type="checkbox"/>	25	262	R	772100.188	8163.068	124.54
<input type="checkbox"/>	19	230	R	1680685	9626.42	127.435
<input type="checkbox"/>	22	250	R	443898.906	5834.24	129.152
<input type="checkbox"/>	40	373	R	396078.031	4328.676	130.398
<input type="checkbox"/>	16	199	R	691038.562	5093.059	134.131
<input type="checkbox"/>	34	279	R	990726.062	8027.383	136.864
<input type="checkbox"/>	28	270	R	911816.938	9007.464	138.164
<input type="checkbox"/>	10	115	R	1390545.5	6656.327	141.537
<input type="checkbox"/>	13	192	R	619879.062	6221.566	144.801

Fig. 3.54: Graus de similaridades do Leucócito 21 da tabela Resultado2.

Ao se realizar a consulta no banco de dados da tabela Resultado2 ordenada pela consulta dos valores de uma determinada técnica, seja Bhattacharyya, Mahalanobis ou Jeffries-Matusita, para um Leucócito, observou-se que os valores dos graus de similaridade dos Leucócitos que pertencem a mesma patologia do Leucócito que apresenta grau zero, aparecem logo em seguida ao mesmo. A Figura 3.55 mostra os resultados do cálculo de similaridade do Leucócito 250 (ou Célula Teste 8 da Figura 3.53). As três linhas selecionadas são três Leucócitos que pertencem a mesma patologia do Leucócito 250, que esta na primeira linha da consulta. A consulta mostrada na Figura 3.55 foi ordenada pela coluna (JM).

Analisando-se todos os resultados dos testes, para cada uma das bandas no formato RGB, com os valores dos graus de similaridade ordenados por uma das técnicas, foi verificado que para os casos em que há muitos exemplos de Leucócitos para uma determinada patologia, as técnicas de Mahalanobis e Jeffries-Matusita chegavam a apresentar de 1 até 3 Leucócitos que pertencem a mesma patologia nas linhas subseqüentes à linha da consulta, como pode ser observado também na Figura 3.55.

As técnicas de Bhattacharyya, Mahalanobis e Jeffries-Matusita mostraram pelos

resultados, que é possível apresentar as medidas de similaridade calculadas a partir de uma dada imagem M1 para as outras imagens de forma ordenada e crescente, em que o valor zero indica que o resultado do cálculo de similaridade feita com a própria imagem, e que os outros valores indicam, o quanto uma imagem é similar a M1. Quanto maior os valores, menos similares serão as imagens.

A Figura 3.55 mostra os resultados dos cálculos de similaridade realizado pelas técnicas de Bhattacharyya, Mahalanobis e Jeffries-Matusita para o Leucócito 250 (Este valor se encontra na primeira linha na coluna ‘Celula.ID’) ou ‘Célula Teste 8’ da Figura 53. Os resultados estão ordenados em ordem crescente pelos resultados da técnica de Jeffries-Matusita. As três linhas seguintes à primeira, mostram os Leucócitos que possuem maior similaridade com o Leucócito da primeira linha. A última linha mostra o Leucócito menos similar ao Leucócito da primeira linha.

Os testes mostraram que, quanto maior o número de imagens exemplos possuir uma patologia, mais Leucócitos da mesma patologia apareceriam depois da primeira linha da consulta. Com base nisto, foi feito um estudo para avaliar qual das técnicas Bhattacharyya, Mahalanobis e Jeffries-Matusita apresentava o maior número de células de uma mesma patologia, nas três primeiras linhas da consulta após a primeira. Este estudo foi realizado somente para patologias que tenham pelo menos três imagens exemplos. Os resultados do estudo serão mostrados no Capítulo de Resultados.

Este estudo foi feito da seguinte maneira: fizeram-se as contagens das quantidades dos Leucócitos pertencentes a mesma patologia, que apareciam nas três primeiras linhas da consulta ordenada por algumas das técnicas de Bhattacharyya, Mahalanobis e Jeffries-Matusita, em cada uma das componentes RGB. Esta contagem só foi feita para as patologias de Desordem MyeloProliferativo, Anemias Macroclíticas e Desordem Linfoproliferativo.

Ao analisar os resultados, verificou-se que a técnica que mais apresentou Leucócitos

ResBM_ID	TesteBM_ID	Celula_ID	Banda	B	M	JM	
<input type="checkbox"/> 358	8	250	R	0	0	0	
<input type="checkbox"/> 367	8	277	R	51283.559	1459.221	61.097	
<input type="checkbox"/> 370	8	279	R	69224.977	1777.751	71.047	
<input type="checkbox"/> 361	8	262	R	174828.312	3752.681	74.326	
<input type="checkbox"/> 355	8	230	R	159997.844	4096.659	77.859	
<input type="checkbox"/> 364	8	270	R	120675.953	3155.195	86.192	
<input type="checkbox"/> 352	8	199	R	104490.586	3532.332	88.013	
<input type="checkbox"/> 373	8	312	R	35556.434	1862.902	97.528	
<input type="checkbox"/> 346	8	115	R	164005.125	3941.961	97.871	
<input type="checkbox"/> 379	8	382	R	67230.352	2533.447	98.815	
<input type="checkbox"/> 349	8	192	R	152234.203	3926.622	99.676	
<input type="checkbox"/> 376	8	373	R	47377.59	1979.718	104.429	
<input type="checkbox"/> 340	8	31	R	98789.656	3072.848	112.608	
<input type="checkbox"/> 343	8	93	R	161867.469	3186.611	121.977	
<input type="checkbox"/> 382	8	447	R	730463.688	4598.892	130.882	
<input type="checkbox"/> 337	8	21	R	1082176.75	8161.985	143.666	

Fig. 3.55: Consulta com os resultados dos cálculos de similaridade do Leucócito 250, ordenados pela coluna (JM).

da mesma patologia nas três linhas seguintes à primeira, foi a de Jeffries-Matusita. Foi possível também verificar pela associação dos resultados das três primeiras linhas que alguns dos Leucócitos da Figura 53, pertenciam a uma mesma classe ou patologia. Os resultados da associação das três primeiras linhas que mostram que alguns Leucócitos da Figura 53 pertencem a uma mesma classe ou patologia são apresentados no Capítulo de Resultados bem como, os resultados da associação das três primeiras linhas.

### 3.4 Conclusão

Neste capítulo, foram apresentadas as metodologias utilizadas para a realização da segmentação, extração e classificação das patologias. A segmentação descrita foi realizada em alguns passos. Já na classificação, não foi possível realizar uma pesquisa com maior profundidade pela falta de imagens exemplos. Os resultados dos passos feitos neste capítulo serão apresentados no Capítulo de Resultados.

# Capítulo 4

## Resultados

Neste capítulo serão apresentados os resultados dos passos de segmentação, extração e classificação de patologias abordadas no capítulo anterior.

### 4.1 Segmentação

#### 4.1.1 Oversegmentation

Serão mostradas algumas imagens resultantes da oversegmentation aplicada nas imagens originais. As Figuras 4.1 e 4.3 são consideradas imagens padrão para o armazenamento de cores para as classes de cores. As Figuras 4.2 e 4.4 são imagens resultantes da oversegmentation aplicado nas Figuras 4.1 e 4.3. A Figura 4.5 é uma imagem original que foi selecionada a partir das 30 imagens que não foram utilizadas como padrão. A Figura 4.6 é a imagem resultante da oversegmentation da Figura 4.5. As Figuras 4.1, 4.3 e 4.5 são utilizadas como Figuras padrão para se comparar com as imagens resultantes da oversegmentation, nas Figuras 4.2, 4.4 e 4.6. A oversegmentation cria um efeito de borramento observado nas figuras resultantes e isto vem da diminuição das cores presentes na imagem.



Fig. 4.1: Imagem de sangue normal.

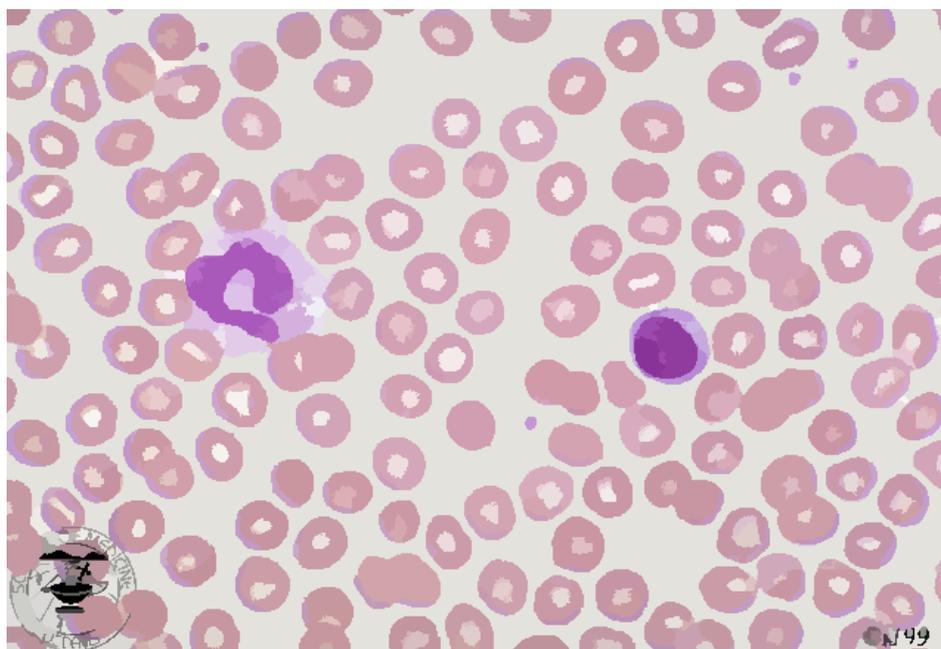


Fig. 4.2: Imagem resultante da oversegmentation da Figura 4.1.

#### 4.1.2 Marcação das classes de cores de Pixels.

É mostrado um exemplo de como são marcados de preto as imagens resultantes da oversegmentation através do uso da ferramenta gimp. As Figuras 4.7 e 4.8

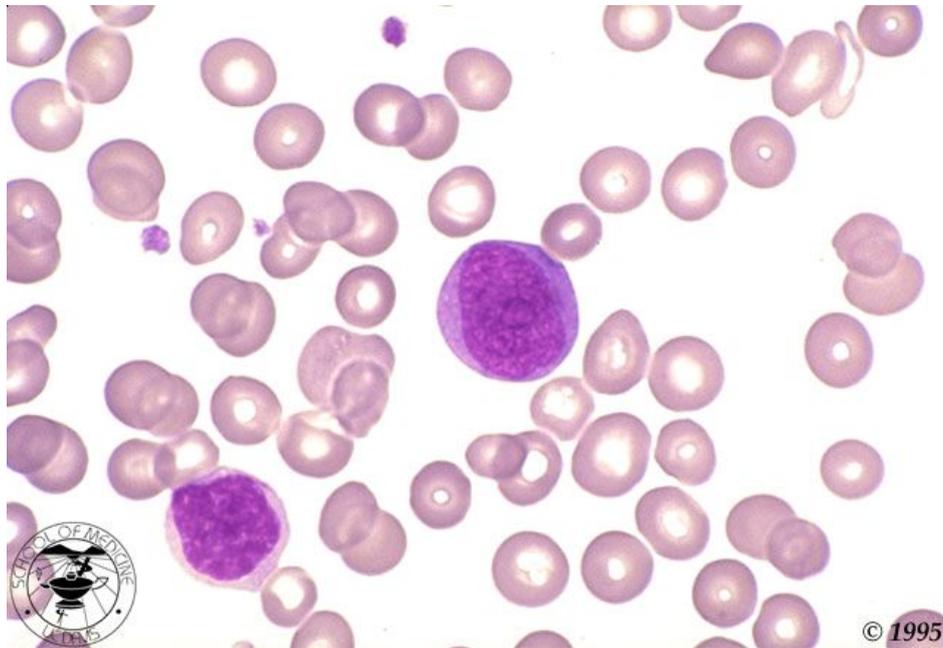


Fig. 4.3: Imagem original de sangue que apresenta uma Leucemia Aguda.

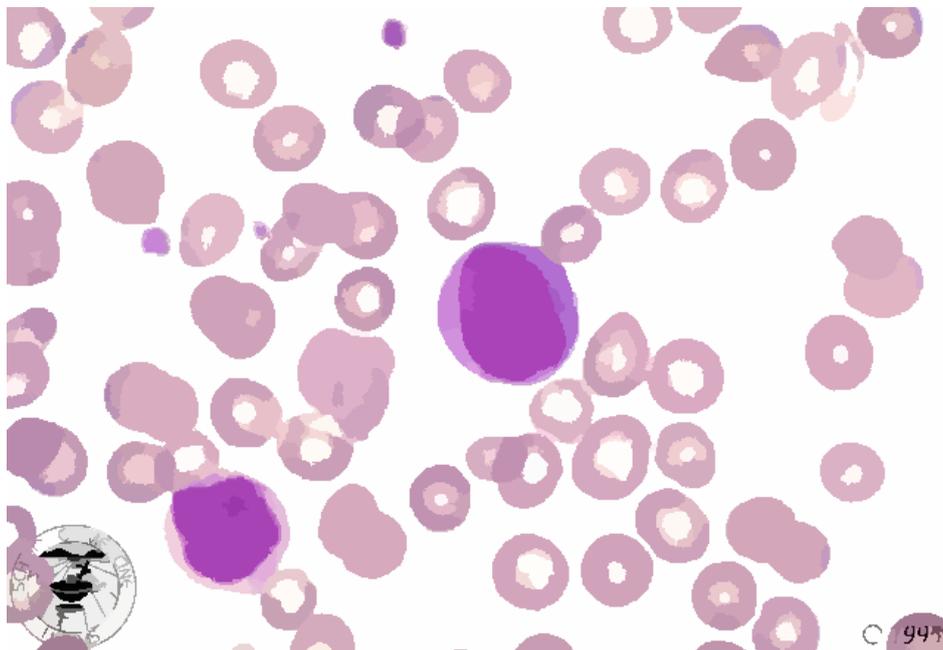


Fig. 4.4: Imagem resultante da oversegmentation da Figura 4.3.

foram utilizadas para a obtenção das cores das Classes de Fundo e de Leucócitos e Plaquetas. A Figura 4.7 mostra o resultado da Figura 4.2 quando os pixels das células foram marcados de preto.

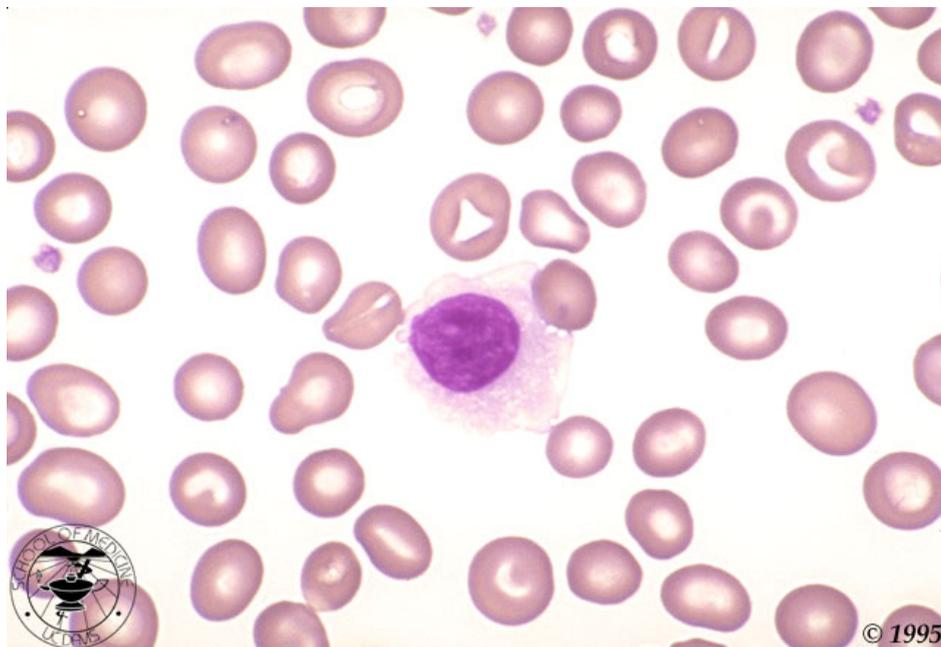


Fig. 4.5: Imagem do sangue que apresenta a patologia de Desordem Limfoproliferativo.

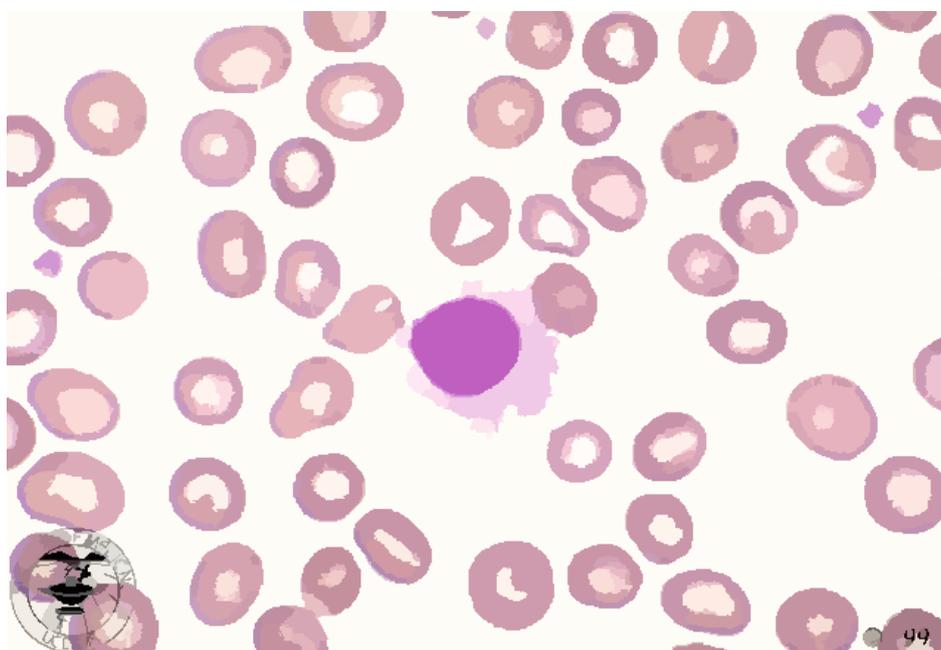


Fig. 4.6: Imagem resultante da Oversegmentation da Figura 4.5.

A Figura 4.8 mostra o resultado da Figura 4.2 quando todas as células de hemácias foram marcadas de preto junto com os pixels de fundo.

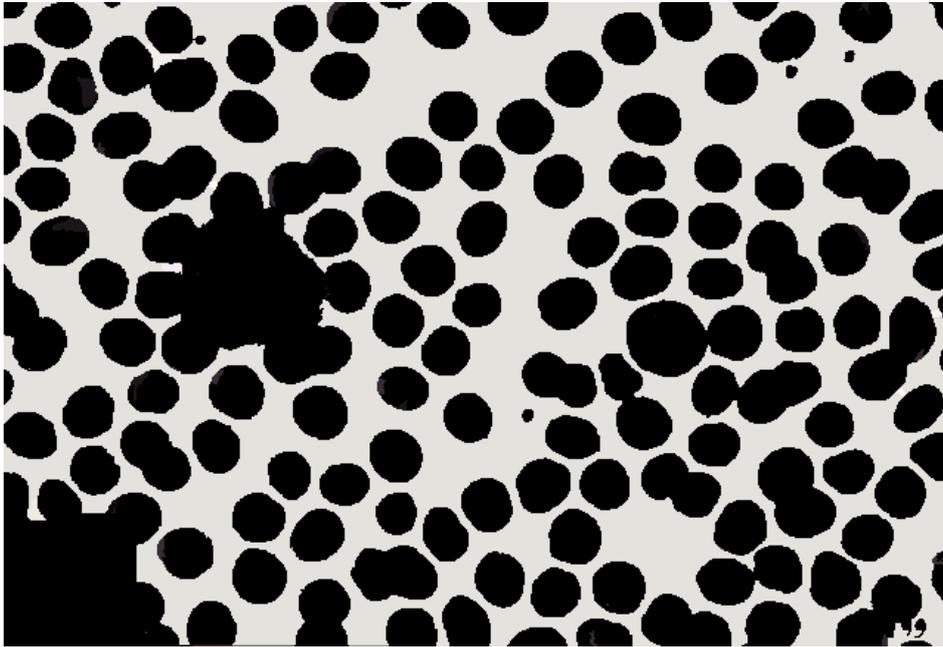


Fig. 4.7: Imagem da Figura 4.2 com todas as células marcadas de preto.

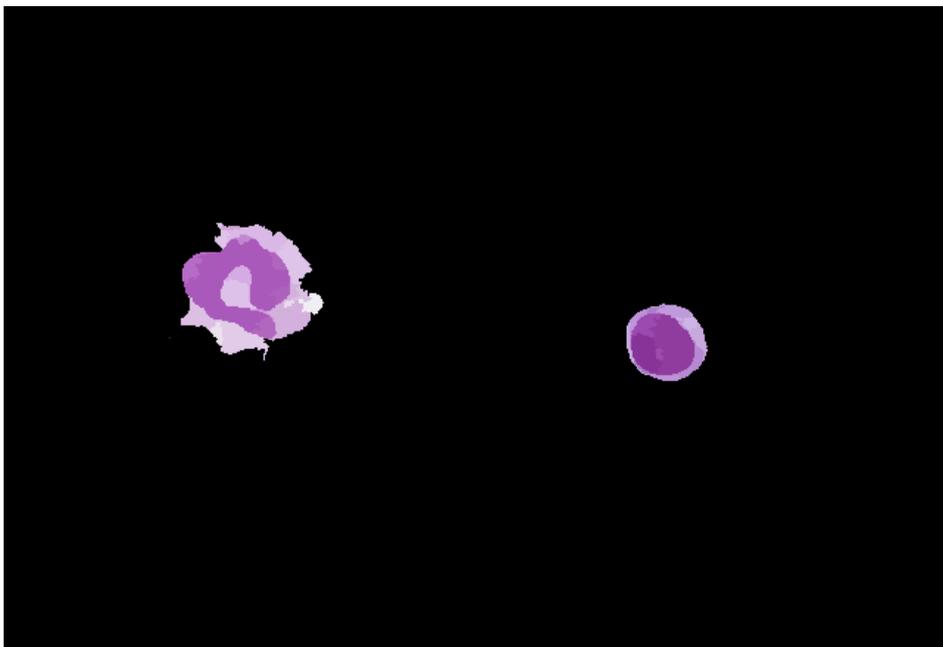


Fig. 4.8: Imagem da Figura 4.2 com os pixels de fundo e de hemácias pintadas de preto.

### 4.1.3 Cores Armazenadas

A seguir, será mostrado como os dados das cores que foram armazenadas no banco de dados. A estrutura das tabelas utilizadas no banco de dados é mostrada

na Figura 4.9 e os dados das respectivas tabelas são mostrados na Figura 4.10.

A estrutura mostrada na Figura 4.9(a), mostra a estrutura da tabela ‘Cor\_Fundo’, que armazena as cores dos pixels de fundo, onde ‘Cor\_Fundo.ID’ armazena o valor do campo chave da tabela e este valor é gerado automaticamente pelo banco de dados. Os campos ‘R’, ‘G’ e ‘B’, armazenam os valores dos componentes RGB. A Figura 4.10(a), mostra um exemplo das cores dos pixels que foram armazenados nesta tabela.

A estrutura mostrada na Figura 4.9(b), mostra a estrutura da tabela ‘Cor\_GB’, em que ‘GB’ vem de ‘Glóbulos Brancos’ e armazena as cores dos pixels de Leucócitos e Plaquetas. Onde ‘Cor\_GB.ID’ armazena o valor do campo chave da tabela e este valor é gerado automaticamente pelo banco de dados. Os campos ‘R’, ‘G’ e ‘B’, armazenam os valores dos componentes RGB. A Figura 4.10(b), mostra um exemplo das cores dos pixels de Leucócitos e Plaquetas que foram armazenados nesta tabela.

Cor_Fundo	Cor_GB
Cor_Fundo_ID: INTEGER	Cor_GB_ID: INTEGER
R: INTEGER	R: INTEGER
G: INTEGER	G: INTEGER
B: INTEGER	B: INTEGER

(a) (b)

Fig. 4.9: Estruturas das tabelas, (a) estrutura da tabela da cor de fundo e (b) da de Leucócitos e Plaquetas.

Cor_fundo_ID	R	G	B
<input type="checkbox"/> 1	254	253	253
<input type="checkbox"/> 2	251	238	245
<input type="checkbox"/> 3	252	239	244
<input type="checkbox"/> 4	232	230	224
<input type="checkbox"/> 5	248	235	235
<input type="checkbox"/> 6	250	239	238
<input type="checkbox"/> 7	253	246	245
<input type="checkbox"/> 8	244	229	225
<input type="checkbox"/> 9	254	247	244
<input type="checkbox"/> 10	238	232	227
<input type="checkbox"/> 11	248	247	241
<input type="checkbox"/> 12	249	238	248
<input type="checkbox"/> 13	251	240	242
<input type="checkbox"/> 14	231	210	208
<input type="checkbox"/> 15	240	225	220
<input type="checkbox"/> 16	223	209	203
<input type="checkbox"/> 17	230	215	208
<input type="checkbox"/> 18	239	226	219

(a)

Cor_GB_ID	R	G	B
<input type="checkbox"/> 1	182	137	191
<input type="checkbox"/> 2	211	191	225
<input type="checkbox"/> 3	194	143	209
<input type="checkbox"/> 4	196	158	211
<input type="checkbox"/> 5	229	215	234
<input type="checkbox"/> 6	231	218	232
<input type="checkbox"/> 7	222	197	233
<input type="checkbox"/> 8	218	184	230
<input type="checkbox"/> 9	194	135	210
<input type="checkbox"/> 10	169	89	186
<input type="checkbox"/> 11	222	196	233
<input type="checkbox"/> 12	177	101	191
<input type="checkbox"/> 13	182	107	198
<input type="checkbox"/> 14	228	213	237
<input type="checkbox"/> 15	212	171	228
<input type="checkbox"/> 16	212	176	221
<input type="checkbox"/> 17	222	193	232
<input type="checkbox"/> 18	172	98	188

(b)

Fig. 4.10: Dados das tabelas de cores de Fundo e de Leucócitos e Plaquetas.

#### 4.1.4 Estudo da distância de cores

A seguir, serão mostradas algumas imagens resultantes da marcação de pixels das classes de cores que foram marcados de preto para cada classe de cor. Serão mostrados apenas os resultados das imagens que não foram escolhidas como padrão.

##### Resultado da marcação de pixels das cores da Classe de Fundo

As Figuras de 4.11 a 4.16 foram colocadas como exemplo visual das marcações dos pixels das Classes de cores de Fundo utilizando as distâncias euclidianas escolhidas.

As Figuras 4.11, 4.12 e 4.13 mostram os resultados da marcação dos pixels da Figura 4.4, pelas distancias euclidianas 4, 5 e 10. As Figuras 4.14, 4.15 e 4.16 mostram os resultados da marcação dos pixels da Figura 4.6, pelas distancias euclidianas 4, 5 e 10. Pode-se notar nas Figuras de 4.12, 4.13 e nas Figuras 4.15 e 4.16, que os pixels das hemácias também foram marcados de preto à medida que se aumentava a distância euclidiana, causando a deformação das células de hemácias

nas imagens. No exemplo da Figura 4.17 poderá se ter uma melhor compreensão do que foi descrito anteriormente.

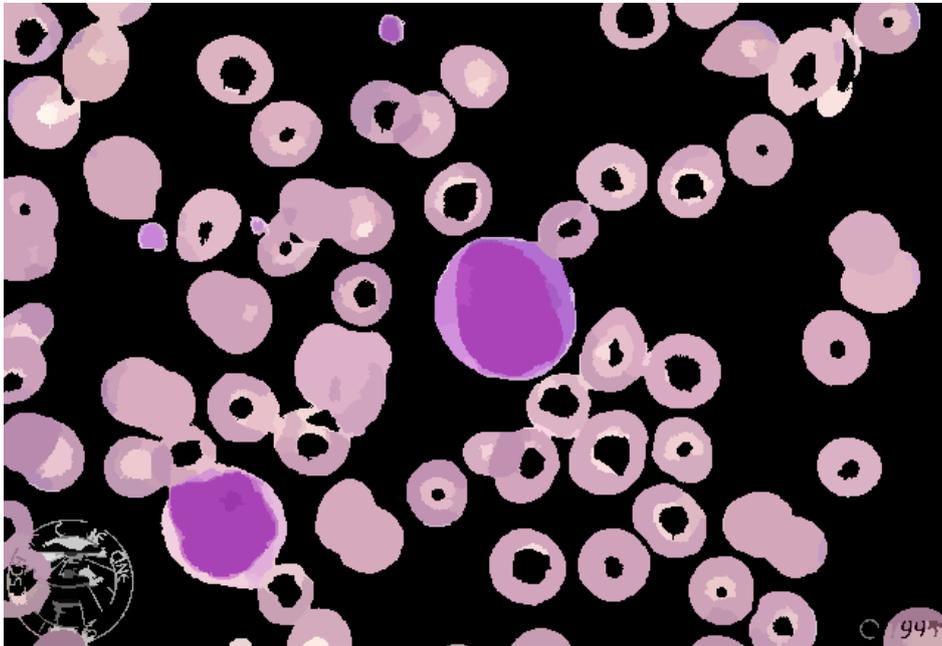


Fig. 4.11: Imagem da Figura 4.4 com os pixels de fundo marcados pela distancia 4.

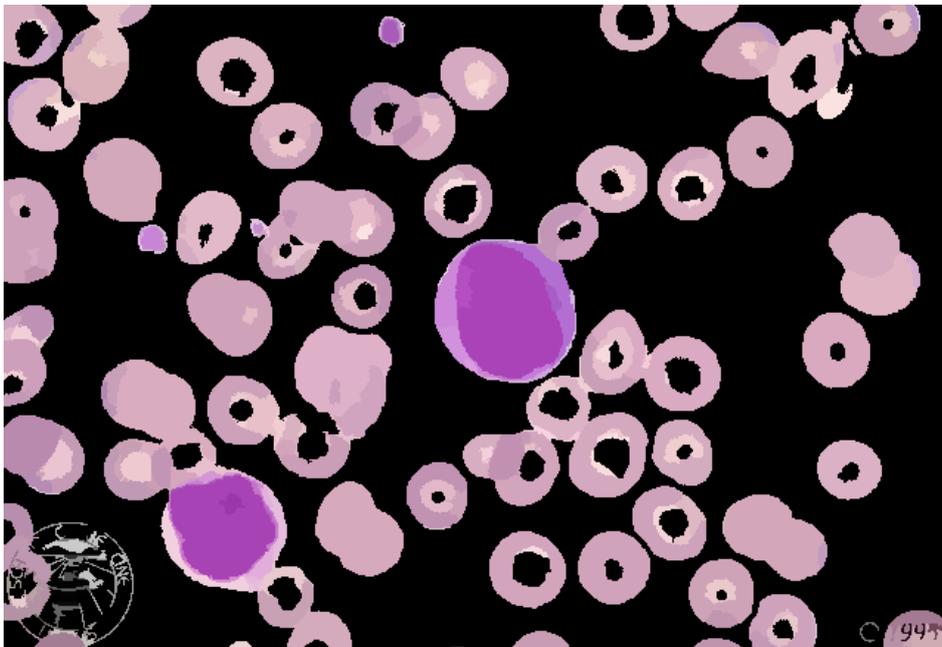


Fig. 4.12: Imagem da Figura 4.4 com os pixels de fundo marcados pela distancia 5.

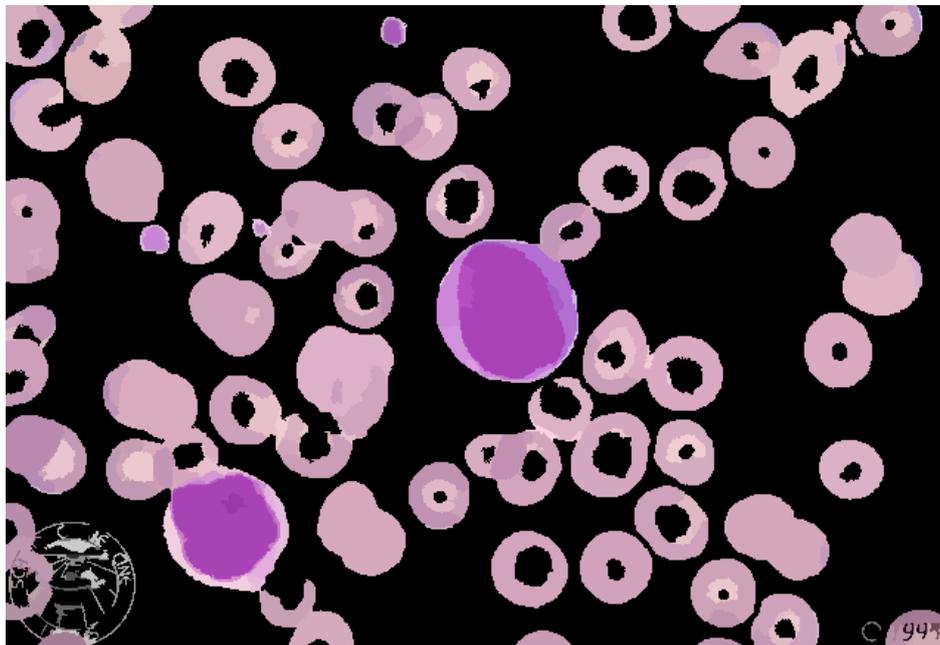


Fig. 4.13: Imagem da Figura 4.4 com os pixels de fundo marcados pela distancia 10.

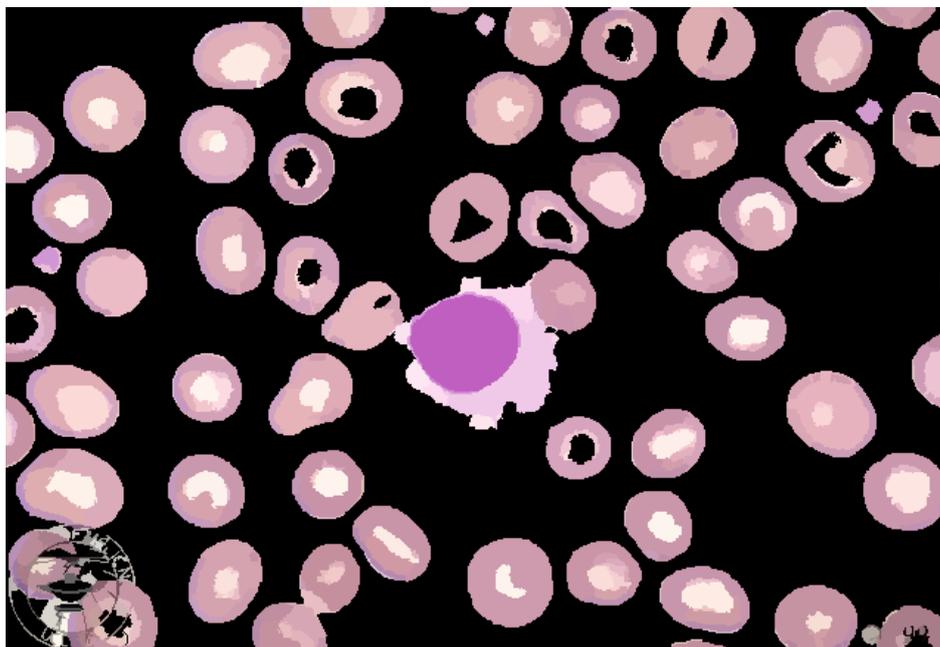


Fig. 4.14: Imagem da Figura 4.6 com os pixels de fundo marcados pela distancia 4.

Na Figura 4.17(a) pode ser observado que as hemácias não foram tão deformadas quanto nas figuras 4.17(b) e 4.17(c). A Figura 4.17(a) foi gerada marcando-se os pixels que tinham distância euclidiana menor ou igual 4. A Figura 4.17(b) apresenta

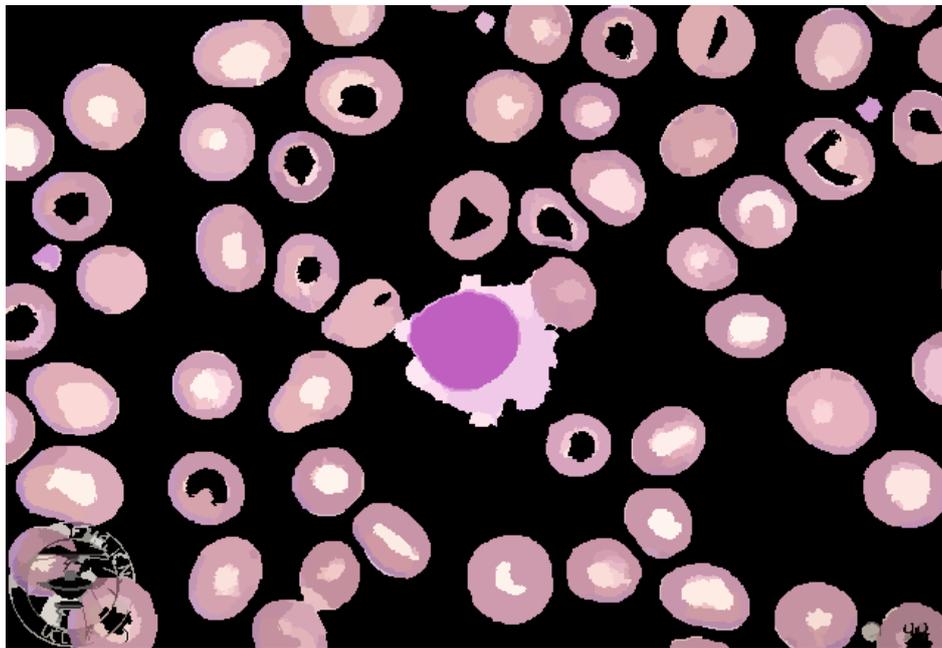


Fig. 4.15: Imagem da Figura 4.6 com os pixels de fundo marcados pela distancia 5.

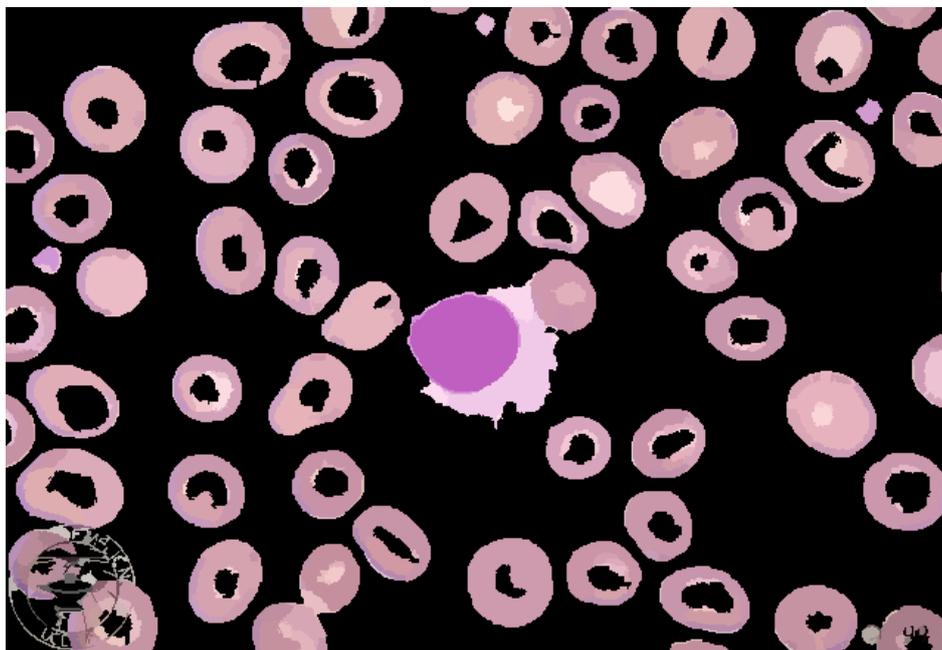


Fig. 4.16: Imagem da Figura 4.6 com os pixels de fundo marcados pela distancia 10.

uma deformação da hemácia (parte marcada). Esta figura foi gerada marcando-se os pixels que tinham distância euclidiana menor ou igual a 5, enquanto que a Figura 4.17(c) apresenta as deformações das hemácias que estão nas partes marcadas. Esta

Figura foi gerada marcando-se os pixels que tinham distância euclidiana menor ou igual a 10.

Nas Letras da Figura 4.17, pode se ver claramente que a distância euclidiana 4 foi a que apresentou melhores resultados.

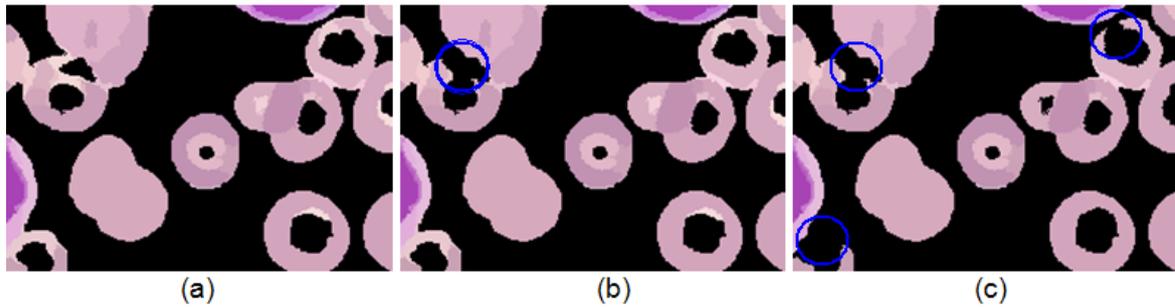


Fig. 4.17: Exemplos de imagens resultantes da marcação utilizando as distâncias selecionadas 4 (a), 5(b) e 10(c).

### Resultado da marcação de pixels das cores da Classe de Leucócitos e Plaquetas

A seguir, serão mostradas algumas imagens resultantes da marcação de pixels das classes de cores de Leucócitos e Plaquetas que foram marcados de preto, sendo que só serão mostrados os resultados das imagens que não foram escolhidas como padrão.

Os Leucócitos e Plaquetas da Figura 4.18 foram marcados pela distância euclidiana 5, sendo que é possível notar que quanto maior a distância euclidiana, maior é a quantidade de pixels de hemácias que são marcados como pixels de Leucócitos e Plaquetas, isto pode ser visto nas Figuras 4.19 e 4.20. Os pixels das hemácias que são marcados de preto são indicados com um círculo azul na imagem.

A distância euclidiana 5 foi escolhida como a distância padrão para as cores de Leucócitos e Plaquetas por marcar o maior número de pixels de Leucócitos e Plaquetas bem como por marcar a menor quantidade de pixels de hemácias como sendo de Leucócitos e Plaquetas das imagens.

O resultado da marcação dos pixels de Leucócitos e Plaquetas, utilizando a distância euclidiana 5, não mostrou bons resultados para a imagem da Figura 4.6, por não marcar todos os pixels dos Leucócitos e Plaquetas. Para resolver os casos em que isto ocorre, é preciso recolher as cores das células de Leucócitos e Plaquetas destas imagens como descrito na Subseção 1.1 do Capítulo 1 de Fundamentações.

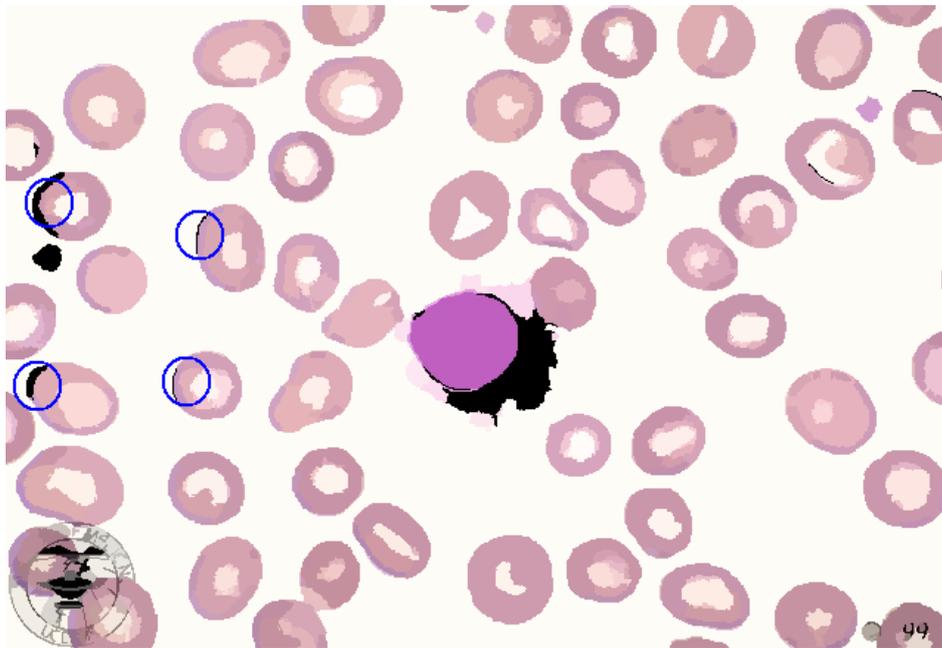


Fig. 4.18: Imagem da Figura 4.6 com os pixels de Leucócitos e Plaquetas marcados pela distância euclidiana 4.

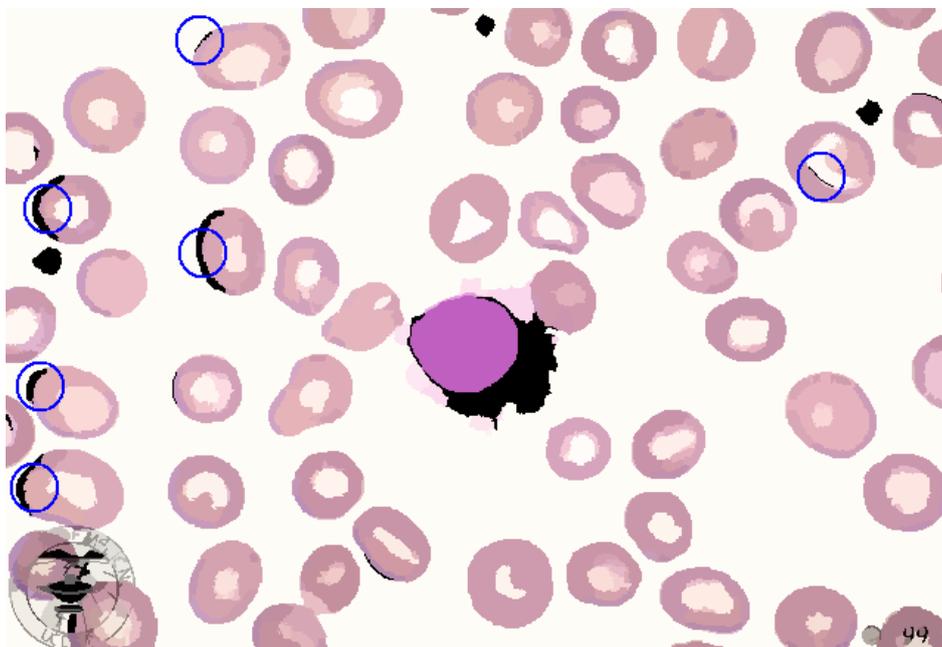


Fig. 4.19: Imagem da Figura 4.6 com os pixels de Leucócitos e Plaquetas marcados pela distância euclidiana 5.

#### 4.1.5 Mapa de Bits das regiões de Classes de cores

Neste tópico, serão mostrados os resultados da seleção das regiões de cores da imagem, que possibilitou a criação de duas imagens a partir da imagem do sangue,

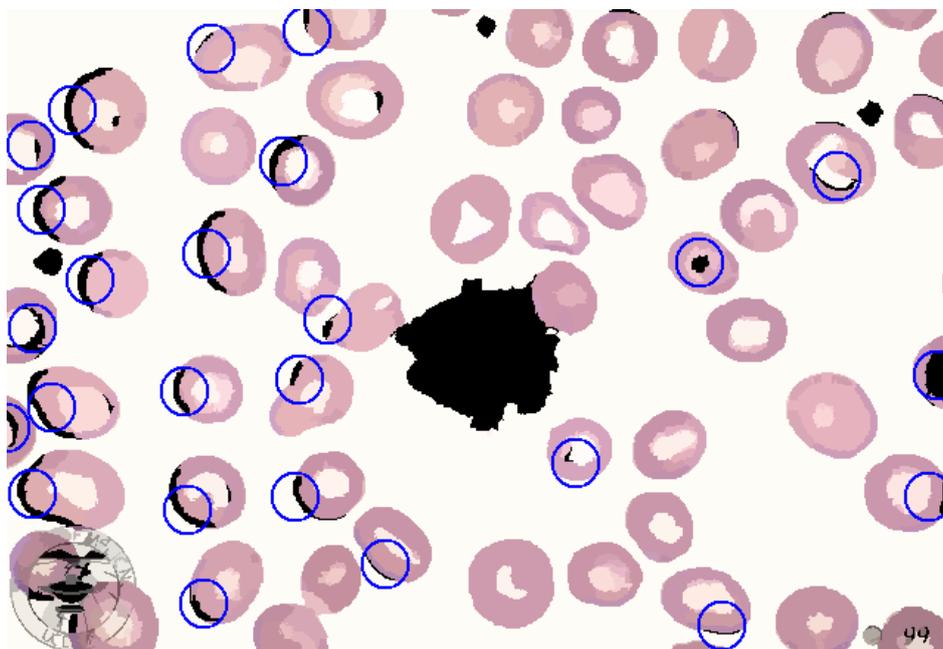


Fig. 4.20: Imagem da Figura 4.6 com os pixels de Leucócitos e Plaquetas marcados pela distância euclidiana 10.

sendo que em uma, estão às células das hemácias (Figuras 4.22, 4.24 e 4.26) e na outra as células de Leucócitos e Plaquetas (Figuras 4.21, 4.23 e 4.25). O que possibilitou a criação destes dois tipos de imagens foram os passos descritos na Subseção 3.2.2 da Seção 1 do Capítulo 3 aplicado no mapa de bits. As figuras que apresentam os Leucócitos e Plaquetas foram criadas a partir da utilização das regiões do mapa de bits das classes de cores destas células para filtrar os pixels de Fundo e de Hemácias da imagem de sangue. As figuras que apresentam as Hemácias foram criadas da mesma maneira.



Fig. 4.21: Imagens de Leucócitos e Plaquetas da Figura 4.1

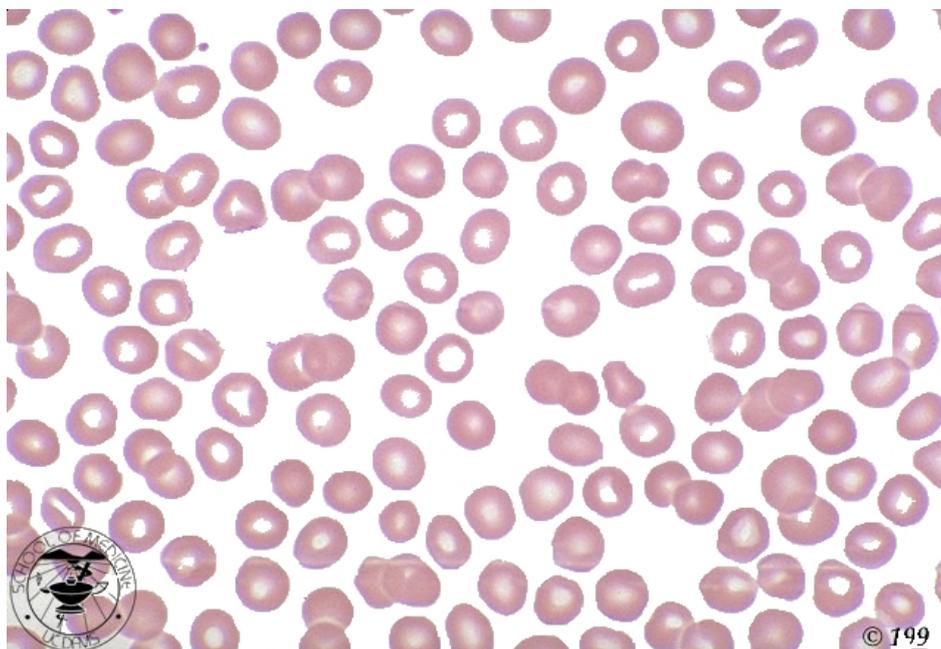


Fig. 4.22: Imagem com as hemácias da Figura 4.1

#### 4.1.6 Mapa de bits das Bordas das células

Serão mostradas a seguir as imagens com as bordas das células que foram geradas pela verificação da vizinhança de 4 dos pixels das regiões de cores, utilizando o





Fig. 4.25: Imagem com os Leucócitos e Plaquetas da Figura 4.5

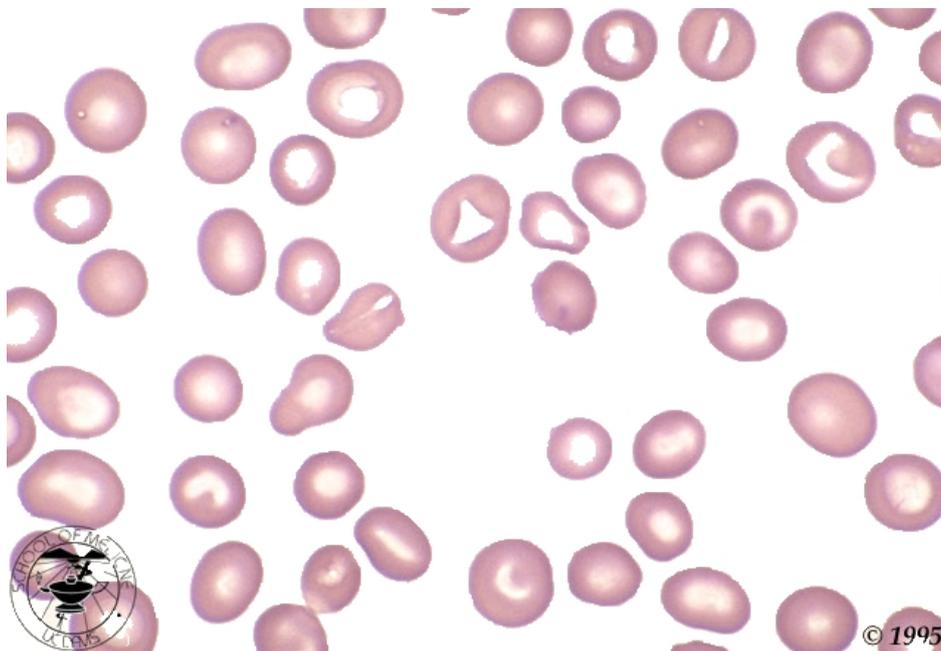


Fig. 4.26: Imagem com as hemácias da Figura 4.5

## 4.2 Extração

A seguir, serão mostrados os resultados da extração das células a partir das bordas das imagens. Antes da extração, foi removido das imagens, o símbolo da

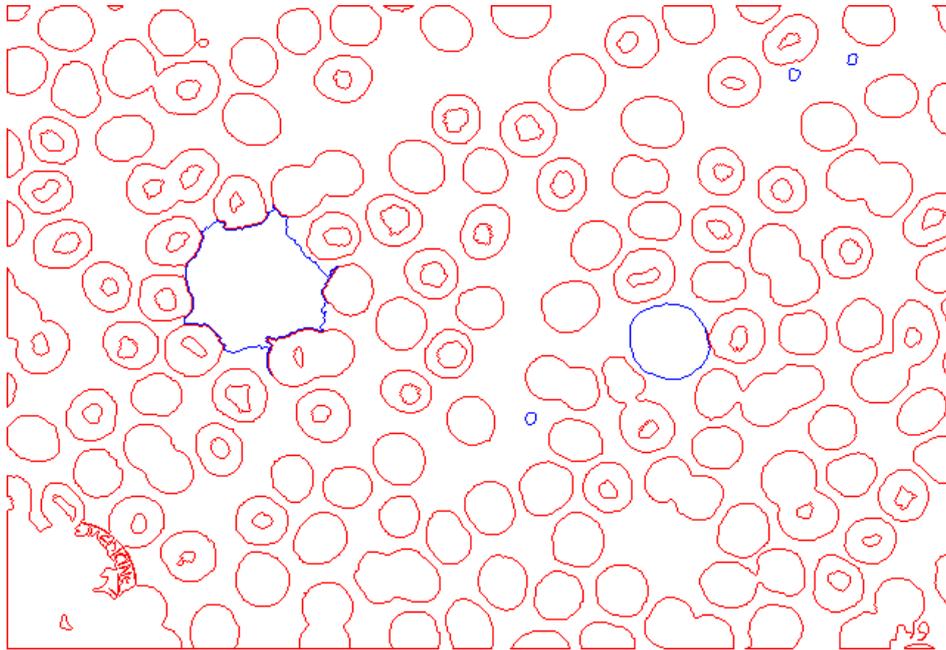


Fig. 4.27: Imagem com as bordas das células da Figura 4.1

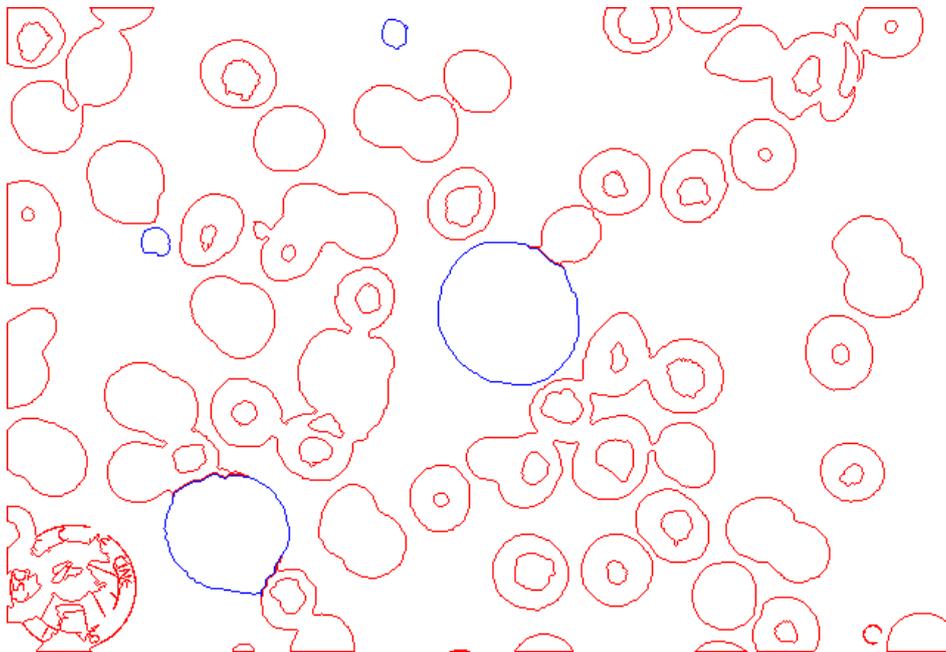


Fig. 4.28: Imagem com as bordas das células da figura 4.3

Universidade Dans da Califórnia por não fazerem parte do contexto das imagens de sangue. Todas as mostradas imagens neste tópico, foram geradas pelo desenvolvimento da técnica proposta na Seção 2 do Capítulo 3. As Figuras 4.30, 4.32, 4.34

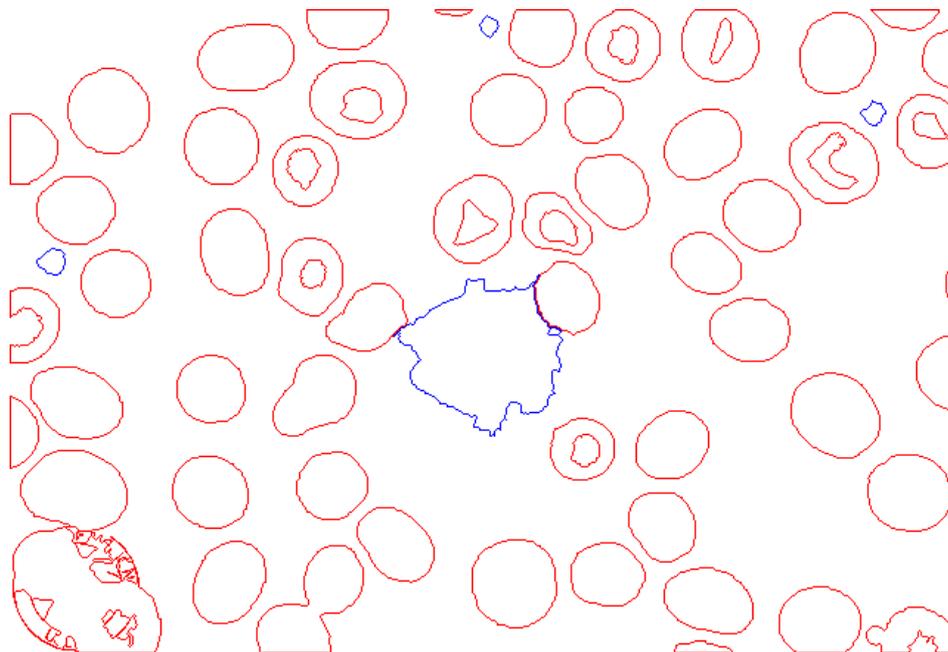


Fig. 4.29: Imagem com as bordas das células da figura 4.6

apresentas as células das Hemácias que foram extraídas das Figuras 4.1, 4.3 e 4.5 respectivamente. As Figuras 4.31, 4.33, 4.35 apresentam as células de Leucócitos e Plaquetas que foram extraídas das Figuras 4.1, 4.3 e 4.5 respectivamente.

### 4.3 Classificação de Patologias

Serão mostrados os resultados do estudo feito para avaliar qual das técnicas Bhattacharyya, Mahalanobis e Jeffries-Matusita apresentava o maior número de células de uma mesma patologia, nas três primeiras linhas da consulta e a associação feita entre os valores apresentados nas três primeiras linhas de alguns Leucócitos em que foi possível constatar de forma numérica que estes Leucócitos realmente estão em uma mesma classe, sendo que mais a frente serão mostrados os critérios utilizados para selecionar os Leucócitos para a associação.

A Figura 4.37 mostra os resultados de um determinado teste, ordenado pela técnica de Bhattacharyya, onde é possível identificar nas três linhas seguintes a

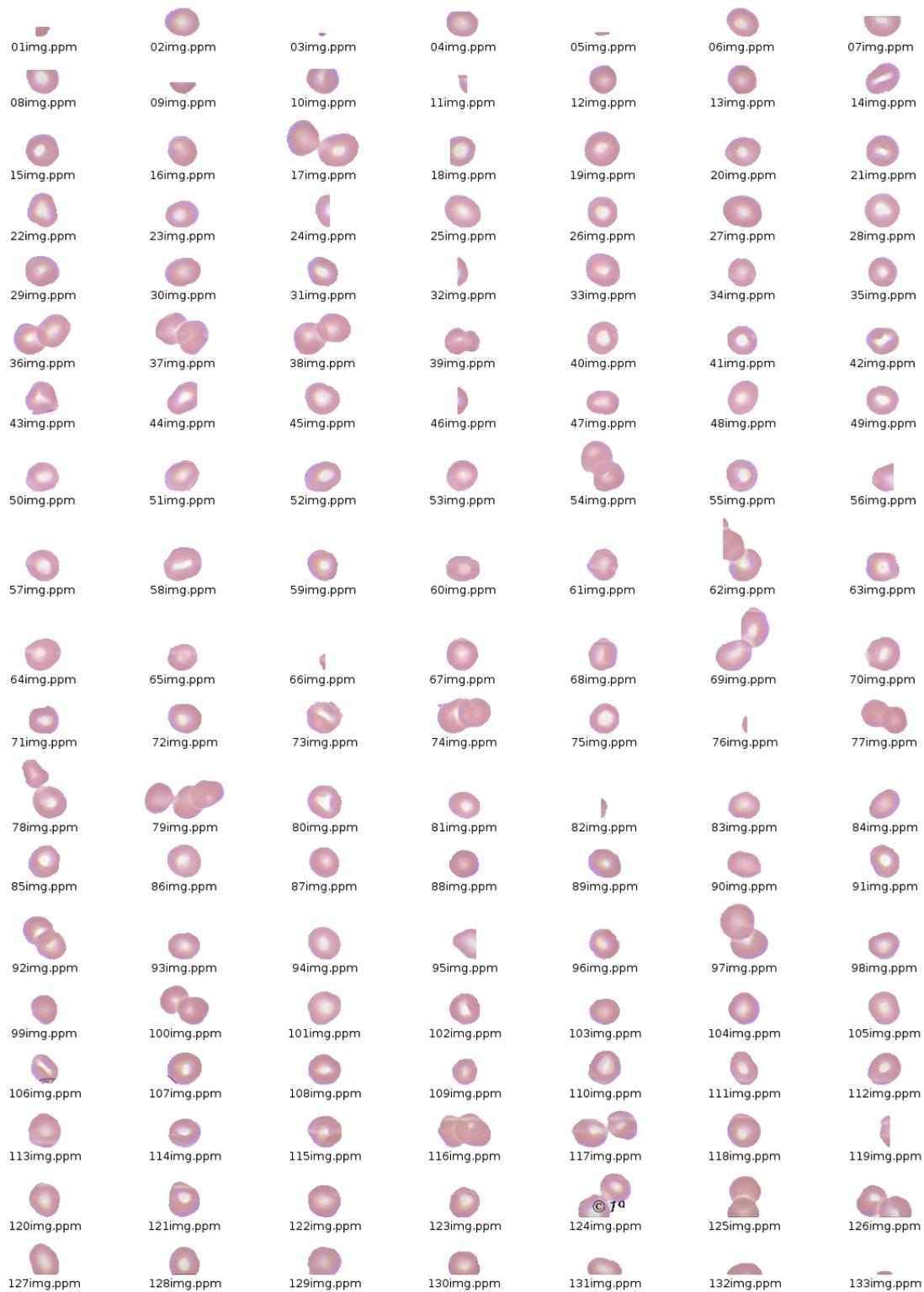


Fig. 4.30: Imagem com as hemácias das células da Figura 4.1

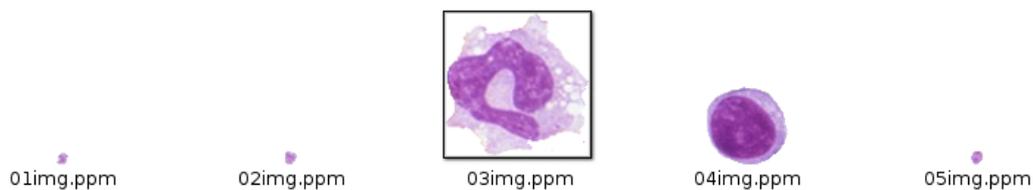


Fig. 4.31: Imagem com os Leucócitos e Plaquetas da Figura 4.1

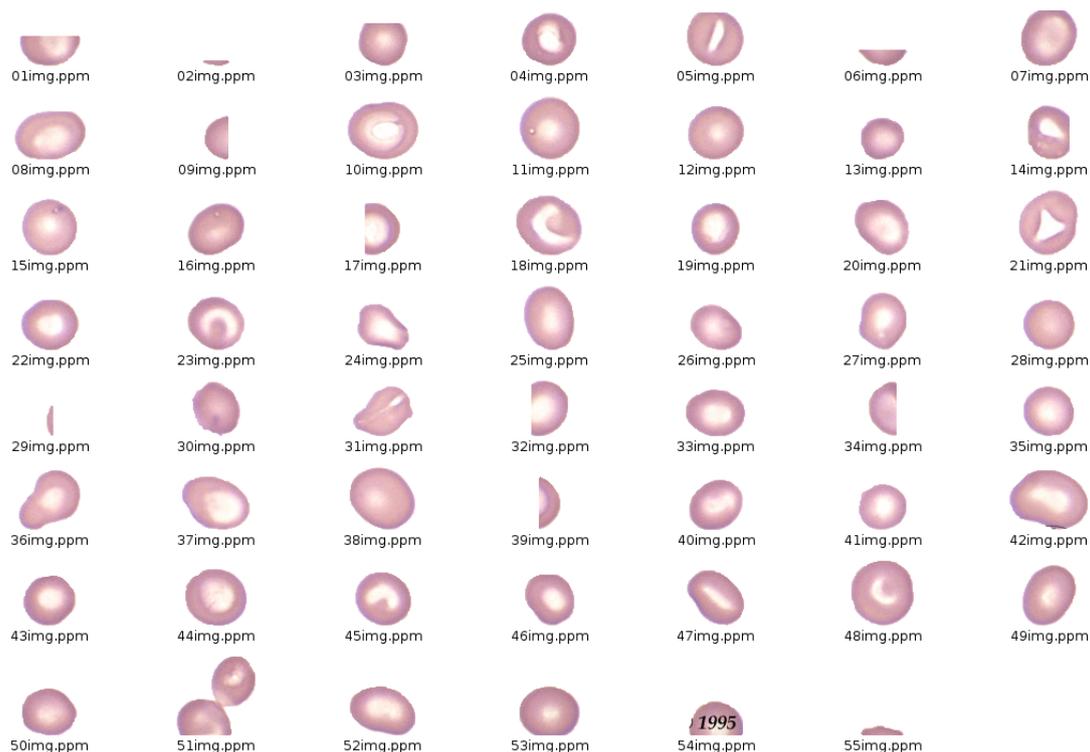


Fig. 4.32: Imagem que apresenta as hemácias da Figura 4.3

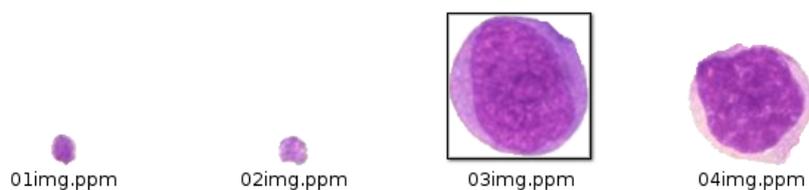


Fig. 4.33: Imagem que apresenta os Leucócitos e Plaquetas da Figura 4.3

primeira linha, ao menos um Leucócito que seja da mesma patologia do Leucócito da Primeira linha, pois o Leucócito selecionado na figura, tem código 199 na coluna 'Celula\_ID', sendo que este é o Leucócito 'Célula Teste 6', da Figura 3.53 do Capítulo 3, verificando-se a Tabela 3.3 do mesmo capítulo. Pode-se constatar que o Leucócito

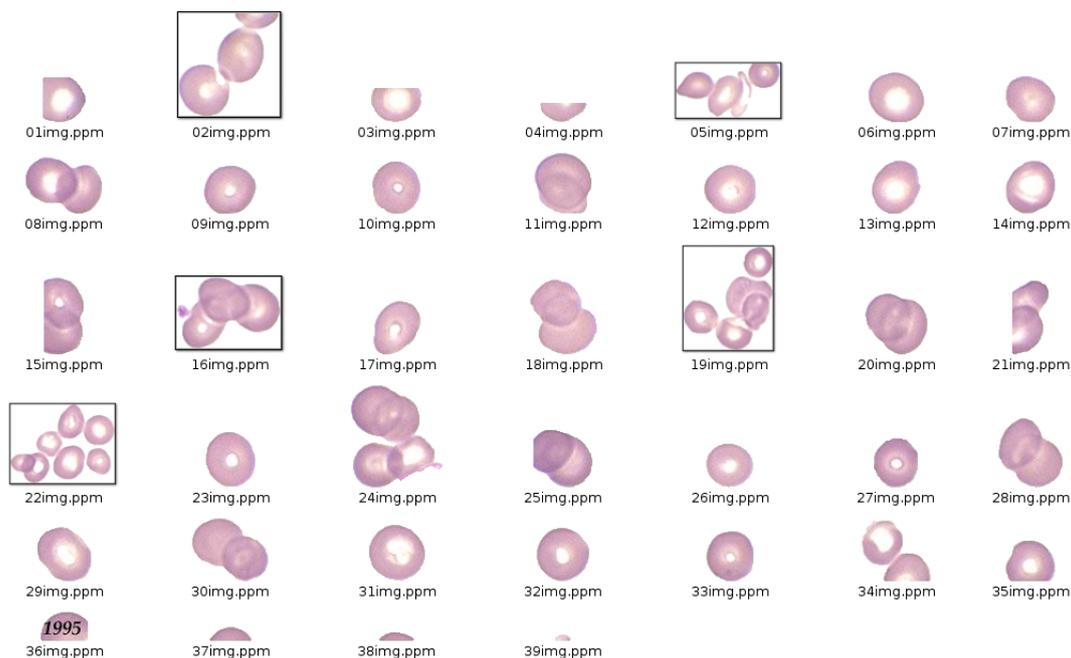


Fig. 4.34: Imagem que apresenta as hemácias da Figura 4.5



Fig. 4.35: Imagem que apresenta os Leucócitos e Plaquetas da Figura 4.5

230 e 199 pertencem a patologia de Desordem MyeloProliferativo. A partir deste exemplo é que foram feitas as contagens que estão na tabela da Figura 4.39 deste capítulo.

A Figura 4.38 mostra os resultados que são apresentados na Figura 3.54 do Capítulo 3, ordenado pelos resultados da técnica de Jeffries-Matusita, em que se pode ver que também foi possível encontrar um Leucócito da mesma patologia nas três primeiras linhas da consulta.

A tabela da Figura 4.39 mostra a contagem feita para cada componente RGB, para as quantidades de Leucócitos da mesma patologia, que apareciam nas três primeiras linhas da consulta ordenada pelas técnicas de Bhattacharyya, Mahalanobis

ResBM_ID	TesteBM_ID	Celula_ID	Banda	B	M	JM	
<input type="checkbox"/>	307	7	230	R	0	0	0
<input type="checkbox"/>	325	7	312	R	67522.445	2095.708	103.176
<input type="checkbox"/>	322	7	279	R	77932.023	3051.168	88.136
<input checked="" type="checkbox"/>	304	7	199	R	89322.477	2100.562	79.48
<input type="checkbox"/>	301	7	192	R	110454.797	2243.121	79.799
<input type="checkbox"/>	298	7	115	R	132168.531	2828.279	84.93
<input type="checkbox"/>	316	7	270	R	132505.062	4327.686	89.818
<input type="checkbox"/>	292	7	31	R	137543.141	3126.408	113.114
<input type="checkbox"/>	328	7	373	R	148840.578	2659.51	100.46
<input type="checkbox"/>	319	7	277	R	157304.109	3875.992	80.056
<input type="checkbox"/>	310	7	250	R	159997.844	4096.659	77.859
<input type="checkbox"/>	331	7	382	R	274635.938	3844.979	94.771
<input type="checkbox"/>	295	7	93	R	292172.25	4160.249	128.963
<input type="checkbox"/>	313	7	262	R	328505.031	5781.507	89.994
<input type="checkbox"/>	334	7	447	R	1237461	6901.032	115.523
<input type="checkbox"/>	289	7	21	R	2318966	11956.889	142.115

Fig. 4.36: Consulta dos Resultados dos cálculos de similaridade feitos para o Leucócito 230, na coluna 'Celula\_ID' da primeira linha.

ResBM_ID	TesteBM_ID	Celula_ID	Banda	B	M	JM	
<input type="checkbox"/>	1	1	21	R	0	0	0
<input type="checkbox"/>	7	1	93	R	1585370.25	7617.044	96.345
<input checked="" type="checkbox"/>	4	1	31	R	488931.625	4881.236	105.909
<input type="checkbox"/>	43	1	382	R	460067.625	4993.766	109.361
<input type="checkbox"/>	46	1	447	R	689960.938	4030.127	116.674
<input type="checkbox"/>	31	1	277	R	1155749.625	9353.474	117.429
<input type="checkbox"/>	37	1	312	R	308768.062	3872.294	120.153
<input type="checkbox"/>	25	1	262	R	772100.188	8163.068	124.54
<input type="checkbox"/>	19	1	230	R	1680685	9626.42	127.435
<input type="checkbox"/>	22	1	250	R	443898.906	5834.24	129.152
<input type="checkbox"/>	40	1	373	R	396078.031	4328.676	130.398
<input type="checkbox"/>	16	1	199	R	691038.562	5093.059	134.131
<input type="checkbox"/>	34	1	279	R	990726.062	8027.383	136.864
<input type="checkbox"/>	28	1	270	R	911816.938	9007.464	138.164
<input type="checkbox"/>	10	1	115	R	1390545.5	6656.327	141.537
<input type="checkbox"/>	13	1	192	R	619879.062	6221.566	144.801

Fig. 4.37: Consulta dos Resultados dos cálculos de similaridade feitos para o Leucócito 21, na coluna 'Celula\_ID' da primeira linha.

e Jeffries-Matusita, para as patologias que tivessem no mínimo 3 imagens exemplo, sendo que, as patologias que atendem a este requisito são as patologias de número: 3, 7 e 8, da Tabela 3.3 do Capítulo de 3.

Obs.: Um asterisco em vermelho na tabela da Figura 4.39, significa que foram encontrados 4 Leucócitos da mesma patologia logo depois da primeira linha da con-

Cód. das Células Teste	Técnica de Bhattacharyya			Técnica de Mahalanobis			Técnica de Jeffries-Matusita		
	R	G	B	R	G	B	R	G	B
5	0	0	0	1	0	0	1	0	0
6	0	0	1	0	0	0	2	1	1
7	0	0	1	0	0	1	0	1	0
8	1	1	0	2	1	0	3	2	3
9	2	1	1	2	1	2	3*	2	3*
10	2	1	1	2	1	1	3	3	3
11	2	1	3	2	1	3	3	2	3
12	1	2	3*	3	1	3*	2	2	2
13	0	1	3	1	0	1	1	3	0
14	1	1	0	1	1	0	1	1	1
15	1	1	1	1	1	1	1	1	1
16	0	0	0	0	0	0	1	1	0
Total	10	9	14	15	7	12	21	19	17

Fig. 4.38: Tabela com o resultado da contagem.

sulta.

A Figura 4.40, mostra um exemplo de identificação dos quatro Leucócitos pertencentes a mesma patologia, onde a primeira linha é o resultado do cálculo de similaridade feito para o Leucócito de número 9 da Tabela da Figura 4.39. As linhas destes Leucócitos aparecem marcados em azul.

ResBM_ID	TesteBM_ID	Celula_ID	Banda	B	M	JM
409	9	262	R	0	0	0
415	9	277	R	73322.609	1865.816	56.702
406	9	250	R	174828.312	3752.681	74.326
412	9	270	R	56277.781	1531.282	75.386
418	9	279	R	225844.953	4558.686	84.621
403	9	230	R	328505.031	5781.507	89.994
427	9	382	R	130418.156	4432.914	100.496
400	9	199	R	187377.672	4666.829	102.241
421	9	312	R	75461.078	2891.121	103.618
394	9	115	R	394908.656	6693.592	106.123
424	9	373	R	106366.484	3726.739	106.921
397	9	192	R	191465.844	4798.948	109.316
388	9	31	R	218926.922	4470.809	110.159
391	9	93	R	239327.641	4247.937	118.916
430	9	447	R	309320.438	4533.688	121.395
385	9	21	R	772259.875	8239.037	136.689

Fig. 4.39: Exemplo em se encontra os 4 Leucócitos da mesma patologia nas 4 linhas seguintes da primeira linha.

Na tabela da Figura 4.39, a parte que aparece marcada em cinza é justamente a parte que foi utilizada para fazer a associação dos resultados das três primeiras

linhas de todas as consultas para os Leucócitos de código de 8 a 12.

Foi possível constatar que os Leucócitos de Código 8 a 12 da coluna de ‘Código das Células Teste’ são de uma mesma classe. Estes dados foram escolhidos porque, das três técnicas utilizadas, a de Jeffries-Matusita foi a que apresentou os valores das similaridades dos Leucócitos da mesma patologia logo nas linhas seguintes da primeira linha da consulta. Por exemplo, a Figura 3.55 do Capítulo 3, mostra que foram contados os valores de três Leucócitos encontrados nas três primeiras linhas da consulta ordenada pelos resultados das técnicas de Jeffries-Matusita. Mas, ordenando a mesma consulta pelos resultados das técnicas de Bhattacharyya, como se pode ver na Figura 4.41, aparece somente o resultado de um Leucócito que pertence a mesma patologia do Leucócito da primeira linha. Este resultado aparece somente na terceira linha. A técnica de Jeffries-Matusita, foi a que apresentou maior similaridade dos Leucócitos pertencentes a uma mesma classe ou patologia logo nas primeiras linhas da consulta ordenada nos componentes R e B do formato RGB, como pode ser visto também na Figura 4.40.

ResBM_ID	TesteBM_ID	Celula_ID	Banda	B	M	JM
358	8	250	R	0	0	0
373	8	312	R	35556.434	1862.902	97.528
376	8	373	R	47377.59	1979.718	104.429
367	8	277	R	51283.559	1459.221	61.097
379	8	382	R	67230.352	2533.447	98.815
370	8	279	R	69224.977	1777.751	71.047
340	8	31	R	98789.656	3072.848	112.608
352	8	199	R	104490.586	3532.332	88.013
364	8	270	R	120675.953	3155.195	86.192
349	8	192	R	152234.203	3926.622	99.676
355	8	230	R	159997.844	4096.659	77.859
343	8	93	R	161867.469	3186.611	121.977
346	8	115	R	164005.125	3941.961	97.871
361	8	262	R	174828.312	3752.681	74.326
382	8	447	R	730463.688	4598.892	130.882
337	8	21	R	1082176.75	8161.985	143.666

Fig. 4.40: Exemplo da Consulta ordenada pelos resultados de Bhattacharyya.

O número da ‘Células Teste’ das respectivas patologias aparecem listados na tabela da Figura 4.42, estes números podem ser consultados na Tabela 2 do capítulo

de Metodologias.

A tabela da Figura 4.42, mostra na coluna 1, os números de referência das linhas. Na coluna 2, os códigos das células teste dados aos Leucócitos da Figura 3.53 do Capítulo de Metodologias. Na coluna 3, os códigos dos Leucócitos no Banco de Dados e na última coluna os códigos dos Leucócitos mais similares aos Leucócitos da terceira coluna.

Os códigos que aparecem em cada linha da última coluna são referentes as três primeiras linhas da consulta ordenada pelos resultados da técnica de Jeffries-Matusita, para os Leucócitos de 8 a 12. Por exemplo, a penúltima coluna, linha 2 da Figura 43, apresenta o número 262 como sendo o código da ‘Célula Teste 9’ da Figura 3.53 do Capítulo de Metodologias, a última coluna, linha 2 da Figura 4.42, tem-se os códigos: 277, 250 e 270, que são os três primeiros códigos que aparecem depois da primeira linha, ‘Celula\_ID’ da Figura 4.40 ao se realizar uma consulta ordenada pelos resultados da técnica de Jeffries-Matusita para o Leucócito de número 262.

IT	Cód. das Células Teste	Cód. dos Leuc. no B.D.	Cód. dos Leuc. das três primeiras linhas
1	8	250	277, 279, 262
2	9	262	277, 250, 270
3	10	270	262, 277, 250
4	11	277	262, 250, 279
5	12	279	277, 250, 192

Fig. 4.41: Tabela com códigos dos Leucócitos que apareciam três linhas seguintes após a primeira linha da consulta.

Fazendo-se um estudo mais apurado dos dados apresentados na Figura 42, foi possível constatar que todos os Leucócitos que estão listados através dos seus códigos na terceira coluna, pertencem a uma mesma classe ou Patologia, devido ao fato de que os códigos dos Leucócitos que aparecem na terceira coluna, estão repetidos nos valores da quarta coluna. Por exemplo, o código 250 aparece listado somente uma vez na terceira coluna, mas este código aparece mais de uma vez na quarta coluna

e isto ocorre em todos os valores listados na terceira coluna, indicando que estes Leucócitos apesar de serem similares, pertencem a uma mesma patologia.

## 4.4 Comentários

Neste capítulo foram mostrados os resultados das técnicas propostas e desenvolvidas no Capítulo de Metodologias. As conclusões serão mostradas no próximo capítulo.

# Capítulo 5

## Conclusões e Trabalhos Futuros

Neste capítulo, serão mostradas as conclusões de cada passo realizado no Capítulo de Fundamentações, sendo que ao final será mostrada uma conclusão geral e em seguida os trabalhos futuros.

### 5.1 Conclusão da Segmentação

A Figura 5.1 seguinte, é a Figura original que serviu de base para os testes de distância de cores bem como para a definição das cores para as classes de cores. As Figuras 5.2, 5.3 e 5.4 são imagens resultantes da segmentação da Figura 5.1. A Figura 5.2 foi gerada pela técnica de Canny [7], a Figura 5.3 foi gerada pela técnica de Segmentação de Imagens Coloridas proposta por Comaniciu [2] e a Figura 5.4 é a imagem com as bordas da Figura 5.1, gerada com a técnica proposta e desenvolvida neste trabalho.

Das técnicas utilizadas como referencias neste trabalho, a técnica de Canny [7] apresentou melhores resultados que a técnica de Comaniciu, por apresentar bordas de células mais limpas, este resultado pode ser observado na Figura 5.3, que apresenta células em que as bordas possuem linhas que dividem as mesmas, gerando

dúvidas no momento da extração das células da imagem.



Fig. 5.1: Imagem padrão utilizada neste trabalho.

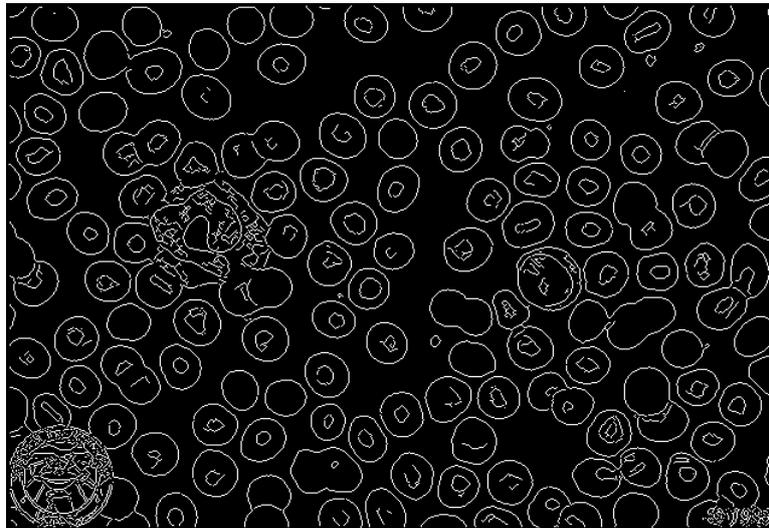


Fig. 5.2: Bordas resultantes da técnica Canny a partir da Figura 1.

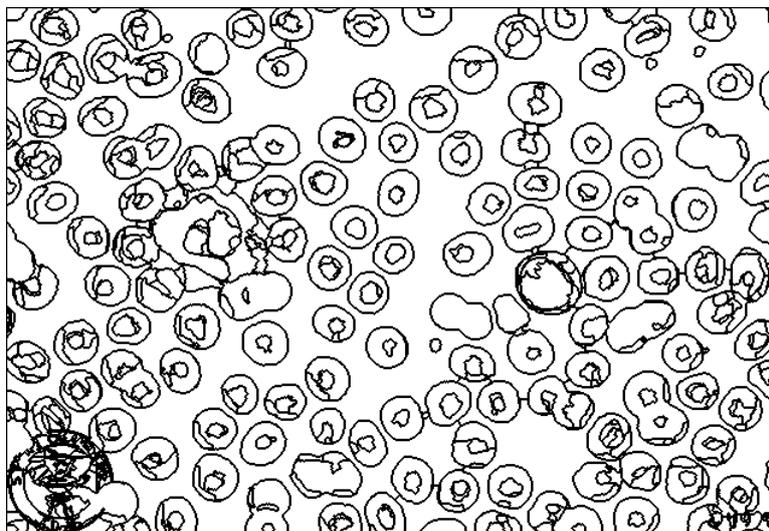


Fig. 5.3: Bordas resultantes da segmentação de imagens coloridas geradas a partir da Figura 1.

As técnicas Canny [7] e de Commaniciu [2] são rápidas computacionalmente se comparadas com outras técnicas (por exemplo em redes neurais), pois levaram menos de 10 segundo para serem executadas na máquina utilizada, o que é vantajoso computacionalmente por levar um tempo muito pequeno para executar a segmentação. A técnica proposta e desenvolvida neste trabalho levou em média 30 segundos o que não foi tão eficiente computacionalmente quanto às técnicas anteriores. Porém, a técnica proposta apresentou maior eficiência na identificação dos segmentos das células de hemácias e de Leucócitos e Plaquetas bem como bordas mais limpas, quando comparada com as técnicas anteriores. Portanto, o tempo gasto pela técnica proposta é compensado pelos melhores resultados obtidos quanto à identificação dos segmentos das células de hemácias e de Leucócitos e Plaquetas, o que foi um dos objetivos buscados neste trabalho, a partir do momento que se optou por criar as classes de cores de Fundo e de Leucócito e Plaquetas. As cores das hemácias foram identificadas por exclusão.

Para o caso das imagens em que os pixels das classes de Fundo, de Leucócitos e Plaquetas não foram marcados quando foi feito o estudo das distâncias, foi preciso

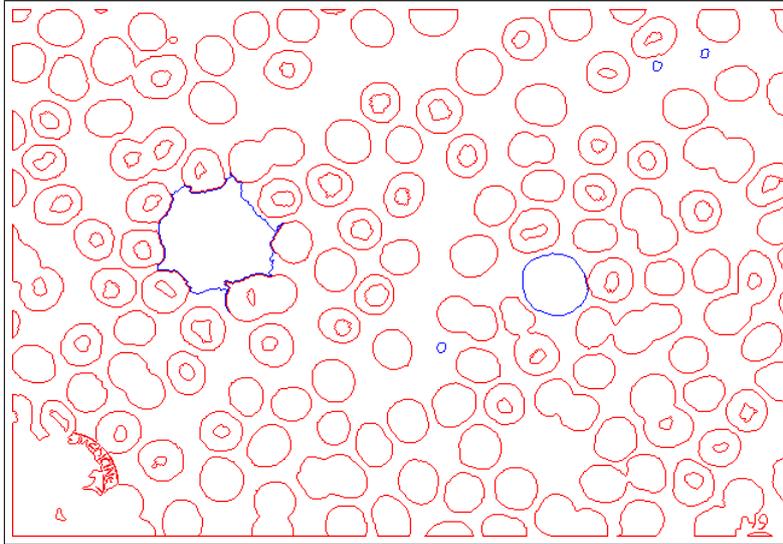


Fig. 5.4: Bordas geradas neste trabalho a partir da Figura 1.

adicionar as cores destas imagens no banco de dados. A adição de cores no banco não afeta a maneira como é realizada a segmentação, apenas melhora o resultado da segmentação nas imagens em que os pixels não foram marcados. Esta adição de cores ocorre apenas devido a diferença de resolução apresentada pelas imagens.

Os problemas encontrados na técnica proposta e desenvolvida é devido ao fato da necessidade de trabalhar com imagens que tenham resoluções parecidas, para evitar que o número de cores no banco de dados cresça de uma forma muito rápida, o que pode levar ao aumento do tempo de segmentação.

Neste trabalho, foram consideradas como imagens com resoluções parecidas, as imagens que apresentavam níveis de intensidades de brilho próximas e que a cor de fundo das imagens também fossem próximas.

Nesta dissertação, não foi realizado um estudo para a verificação do comportamento do crescimento de cores armazenadas no banco de dados. Ficando o mesmo, como proposta de trabalhos futuros.

## 5.2 Conclusão da Extração

A extração de células das imagens de sangue realizadas nesta dissertação foi a que apresentou menores problema quanto a implementação e resultados quando comparada com a segmentação e classificação. A velocidade de extração variou com a quantidade de células apresentadas em cada imagem do sangue, ficando como proposta de trabalho futuro o desenvolvimento de uma técnica que seja ainda mais rápida e mais simples que a desenvolvida e apresentada neste trabalho.

## 5.3 Conclusão da Classificação das Patologias

Os resultados da classificação das patologias utilizando as métricas descritas na parte de fundamentações foram prejudicados pela baixa qualidade das imagens obtidas na internet bem como pelo número reduzido de imagens obtidas para cada patologia.

O que se conclui dos testes realizados, é que quanto maior for o número de exemplos de Leucócitos de uma determinada patologia, melhores serão os resultados da similaridade das imagens obtidas. O teste para verificar a similaridade entre os Leucócitos selecionados mostrou que a técnica de Jeffries-Matusita [6], tem melhores resultados e que esta técnica não só mostrou o grau de similaridade de uma imagem para outra, como também mostrou que foi possível separar os Leucócitos em classes, pela associação dos graus dos Leucócitos mais similares. Apesar disto, não foi possível tirar uma conclusão mais precisa a respeito das técnicas descritas neste trabalho devido a falta de imagens com boa resolução e em grande quantidade.

Com os resultados obtidos, foi possível estabelecer duas hipóteses que poderiam acontecer se o trabalho fosse realizado com imagens com boa qualidade e uma maior quantidade para cada patologia através das métricas descritas neste trabalho:

A primeira, é que pela alta quantidade de exemplos de imagens para cada patologia, a cada classificação realizada, a técnica de Jeffries-Matusita, retornaria os valores das similaridades mais próximas da imagem de teste de uma única patologia.

A segunda, é que haveriam falsos positivos nos resultados, indicando a necessidade da pesquisa de outras técnicas de classificação para melhorar os resultados.

## 5.4 Conclusões Finais

As técnicas utilizadas, propostas e desenvolvidas neste trabalho realizam os passos de segmentação, extração e classificação.

A segmentação utilizando a melhoria da Técnica de Comaniciu [2] obteve os mesmos resultados da técnica de Canny [7], porém, foi possível separar já na segmentação, as bordas das células de Hemácias das células de Leucócitos e Plaquetas. Uma das limitações da técnica de segmentação proposta é a necessidade de trabalhar com imagens que possuam a mesma resolução. Outra limitação, é que se forem utilizadas imagens com resoluções diferentes, pode gerar um crescimento exponencial das cores armazenadas no banco de Dados, o que poderia levar ao aumento do tempo de execução da segmentação. O que não seria bom computacionalmente.

Quando há um crescimento do o número de células dentro da imagem, a técnica de extração apresentou um aumento no tempo de execução, ficando a melhoria do algoritmo (uma técnica mais rápida e mais simples do que a que foi proposta) como trabalho futuro.

Quanto à técnica de classificação que teve os resultados prejudicados por se trabalhar com imagens da internet, ainda pode apresentar melhores resultados com o uso de imagens melhores e em maior quantidade.

As técnicas de Comaniciu [2] e de Birchfield [3] foram melhoradas para os trabalhos desta dissertação. A Técnica de extração foi proposta e desenvolvida. Porém,

as mesmas ainda podem ser melhoradas através de outros trabalhos tendo este como base, podendo assim, completar o desenvolvimento de um Sistema de Apóio ao Diagnóstico Automático de Hemopatologias.

## 5.5 Trabalhos Futuros

Nesta seção serão apresentadas as sugestões de trabalhos futuros na mesma linha do trabalho que foi feito nesta dissertação.

1. Adaptação das técnicas desenvolvidas nesta dissertação para trabalhar com imagens médicas de alta resolução.

Imagens médicas possuem em geral 12-bits de resolução no mínimo, como as imagens utilizadas neste trabalho são de 8-bits de resolução, é preciso adaptar as técnicas desenvolvidas neste trabalho para realizar os passos de segmentação, extração e classificação em imagens médicas de alta resolução.

2. Aplicação de técnicas de CBIR para realizar a Classificação de imagens.

Neste trabalho foram utilizadas técnicas estatísticas que mediam a distância entre as distribuições de cores através de vetores de média e de matrizes de covariância (spatiogram). Não foram aplicadas nenhuma técnica de CBIR para realizar a classificação das patologias nas imagens de sangue. A aplicação de alguma técnica de CBIR poderia dar alguns resultados melhores do que se obteve neste trabalho.

3. Segmentação de imagens do sangue utilizando o espectro de cores.

Um estudo do espectro de frequência de cores poderia estabelecer limites espectrais de cores das classes de fundo e de Leucócitos e Plaquetas, de maneira que isto diminuiria o número de amostras de cores, de forma a possibilitar a segmentação das células utilizando o espectro de cores.

4. Uma técnica rápida para a realização de extração de células em imagens segmentadas.

Uma técnica que realize de forma mais rápida a extração de células do sangue, a partir de imagens segmentadas, pois a velocidade da técnica desenvolvida neste trabalho diminui à medida que o número de células na imagem cresce.

5. Pesquisa do crescimento de cores no banco de Dados.

Quando se decide trabalhar com imagens de resoluções diferentes, é preciso armazenar mais cores das classes de Fundo, de Leucócitos e de Plaquetas no banco de dados para a realização da segmentação, uma pesquisa para saber qual a forma de crescimento destas cores no banco é essencial para uma melhoria deste trabalho.

6. Desenvolvimento de uma solução para o problema anterior.

Se for comprovado que o crescimento de cores no banco de dados é exponencial, é preciso propor e desenvolver uma solução, para que as técnicas desenvolvidas neste trabalho possam ser utilizadas na prática.

# Referências Bibliográficas

- [1] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1992.
- [2] D. Comaniciu and P. Meer. Robust analysis of feature spaces: color image segmentation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, 00:750, 1997.
- [3] S. T. Birchfield and S. Rangarajan. Spatiograms versus histograms for region-based tracking. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2:1158–1163, 2005.
- [4] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin the Calcutta Mathematics Society*, 35:99–110, 1943.
- [5] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, USA, 1999.
- [6] L.V. Dutra and R. Huber. Feature extraction and selection for ers 1/2 insar classification: *Int. J. Remote Sensing*, 20(5):993–1016, 1999.
- [7] Canny F. J. A computational approach to edge detection. *j-IEEE-PAMI*, 8(6):679–698, 1986.

- 
- [8] POZ A. P. D. DO VALE G. MAIA. Processo de detecção de bordas de canny. *Boletim de Ciências Geodésicas*, 8(2):67–78, 2002.
- [9] Bill Green. Canny edge detection tutorial, 2002. Site de Green sobre Canny.
- [10] Ruye Wang. Canny edge detection, 2004. Site de Ruye Wang sobre Canny.
- [11] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.
- [12] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.
- [13] B. Martinkauppi. Face colour under varying illumination - analysis and applications. Academic dissertation, University Oulu, Finlândia., Agosto 2002.
- [14] Spring. Classificação de imagens, 2000. Site de Reconhecimento de Padrões.
- [15] Joaquim P. Marques de Sá. Reconhecimento de padrões, 2000. Site de Reconhecimento de Padrões.
- [16] S. B. Gelfand, C. S. Ravishankar, and E. J. Delp. An iterative growing and pruning algorithm for classification tree design. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(2):163–174, 1991.
- [17] K. Fukunaga and R. R. Hayes. Estimation of classifier performance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(10):1087–1101, 1989.
- [18] Anil K. Jain, Robert P. W. Duin, and Jianchang Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

- 
- [19] M. Basu, H. Bunke, and A. Del Bimbo. Guest editors' introduction to the special section on syntactic and structural pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1009 – 1012, 2005.
- [20] L.T. Tan, C.P. Lian, and C.J.S. De Silva. Comparison between syntactic pattern recognition and the randomized hough transform. *International Conference on Neural Information Processing*, 3:1009 – 1012, 1999.
- [21] D. Casasent. Accuracy effects in pattern recognition neural nets. *International Joint Conference on Neural Networks*, 3:643 – 647, 1992.
- [22] J.B. Kuo, E.J. Wong, and T.L. Chou. A bicmos image sensor circuit for pattern recognition neural network. *IEEE Transactions on Circuits and Systems*, 38(12):1554 – 1556, 1991.
- [23] J.B. Kuo, T.L. Chou, and E.J. Wong. Bicmos edge detector with correlated-double-sampling readout circuit for pattern recognition neural network. *Electronics Letters*, 27(14):1248 – 1250, 1991.
- [24] A. Baraldi and P. Blonda. A survey of fuzzy clustering algorithms for pattern recognition. ii. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 29:786–801, 1999.
- [25] J. M. Keller and H. Qiu. Fuzzy set methods in pattern recognition. In *Proceedings of the 4th International Conference on Pattern Recognition*, pages 173–182, London, UK, 1988. Springer-Verlag.
- [26] T. Chaira and A. K. Ray. Threshold selection using fuzzy set theory. *Pattern Recogn. Lett.*, 25(8):865–874, 2004.
- [27] M. T. Orchard and C. A. Bouman. Color quantization of images. *IEEE Transactions on Signal Processing*, 39(12):2677–2690, 1991.

- [28] R. O. Stehling, M. A. Nascimento, and A. X. Falcão. In *A compact and efficient image retrieval approach based on border/interior pixel classification*, pages 102–109, New York, NY, USA, 2002. ACM Press.
- [29] Dengsheng Zhang and Guojun Lu. A comparison of shape retrieval using fourier descriptors and short-time fourier descriptors. In *PCM '01: Proceedings of the Second IEEE Pacific Rim Conference on Multimedia*, pages 855–860, London, UK, 2001. Springer-Verlag.
- [30] R. Pirrone and M. La Cascia. Texture classification for content-based image retrieval. *Proceedings on International Conference on Image Analysis and Processing*, pages 398–403, 2001.
- [31] K.C. Ravishankar, B.G. Prasad, S.K., Gupta, and K.K. Biswas. In *Dominant color region based indexing for CBIR*, pages 887 – 892. IEEE, 1999.
- [32] A. Folkers and H. Samet. Content-based image retrieval using fourier descriptors on a logo database. *International Conference on Production Research*, 03:30521, 2002.
- [33] K. Jalaja, C. Bhagvati, B.L. Deekshatulu, A.K., and Pujari. Texture element feature characterizations for cbir. *Proceedings on IEEE International Geoscience and Remote Sensing Symposium*, 2:4 pp., 2005.
- [34] Charles E. Jacobs, Adam Finkelstein, and David H. Salesin. Fast multiresolution image querying. *Computer Graphics*, 29(Annual Conference Series).
- [35] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.
- [36] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object

- tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 25(5):564–575, 2003.
- [37] A. Lazarevic, D. Pokrajac, V. Megalooikonomou, and et al. Distinguishing among 3-d distributions for brain image data classification.
- [38] Tzay Y. Youngs and Fu KS. *Handbook of pattern recognition and image processing.* Academic Press, New York, USA, 1986.
- [39] Copenhagen Image Processing System(Chips). Separability analysis, 2000. Site do Programa Chips for Windows.
- [40] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.
- [41] Cecilia Di Ruberto, Andrew Dempster, Shahid Khan, and Bill Jarra. Segmentation of blood images using morphological operators. *icpr*, 03:3401, 2000.
- [42] A. V. Alvarenga, A. F. C. Infantosi, C. M. Azevedo, and W. C. A. Pereira. Aplicação de operadores morfológicos na segmentação e determinação do contorno de tumores de mama em imagens por ultra-som. *Revista Brasileira de Engenharia Biomédica*, 19(2):91–101, 2003.
- [43] I. Cseke. A fast segmentation scheme for white blood cell images. In *Pattern Recognition, 1992. Vol.III. Conference C: Image, Speech and Signal Analysis*, pages 530–533, The Hague, Netherlands, March 1992. IEEE Computer Society.
- [44] Q. Liao and Y. Deng. An accurate segmentation method for white blood cell images. In *Proceedings IEEE International Symposium on Biomedical Imaging*, pages 245–248, Dept. of Electron. Eng., Tsinghua Univ., Beijing, China;, March 2002. IEEE Computer Society.

- [45] M. Marsella and S. Miranda. Neural techniques for image segmentation. *IEEE International Joint Symposia on Intelligence and Systems*, 00:367, 1998.
- [46] Oe S. Hashida M. Shinohara Y. A segmentation method of textured image by using neural network. *Proceedings of International Joint Conference on Neural Networks*, 1:189 – 192, 1993.
- [47] M.A. Ali, G.C. Karmakar, and L.S. Dooley. Fuzzy image segmentation using shape information. *ICME 2005 IEEE International Conference on Multimedia and Expo*, page 4 pp, 2005.
- [48] M.A. Ali, G.C. Karmakar, and L.S. Dooley. Fuzzy image segmentation of generic shaped clusters. *ICIP 2005 IEEE International Conference on Image Processing*, 2:1202–5, 2005.
- [49] C. Tresp, M. Jagar, M. Moser, J. Hiltner, and M. Fathi. A new method for image segmentation based on fuzzy knowledge. *IEEE International Joint Symposia on Intelligence and Systems*, pages 227 – 233, 1996.
- [50] C. J. Taylor, T. F. Cootes, R. H. Davies, and C. J. Twining. Learning to interpret medical images. *IEEE International Symposium on Biomedical Imaging*, pages 34 – 37, 2002.
- [51] Daniela Mayumi Ushizima, Ana C. Lorena, and Andre C. P. L. F. de Carvalho. Support vector machines applied to white blood cell recognition. *Fifth International Conference on Hybrid Intelligent Systems*, 0:379–384, 2005.
- [52] V. Martin, M. Thonnat, and N. Maillot. A learning approach for adaptive image segmentation. *Fourth IEEE International Conference on Computer Vision Systems*, 0:40, 2006.
- [53] UC Davis School of Medicine. Hemolytic anemias, 2001. Faculty of Medicine.

- [54] James D. Foley, Andries V. Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics - Principles and Practice - Second Edition in C*. Addison-Wesley Publishing Company., USA, 1995.