



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

RICARDO DA SILVA BARBOZA

Filtragem de Ruídos em Imagens Digitais

Recife, agosto de 2013

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Filtragem de Ruídos em Imagens Digitais

por

RICARDO DA SILVA BARBOZA

Tese submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco como parte dos requisitos para a obtenção do grau de Doutor em Engenharia Elétrica.

Prof. Dr. Rafael Dueire Lins (orientador)

Recife, agosto de 2013

Catálogo na fonte
Bibliotecária Margareth Malta, CRB-4 / 1198

B239f Barboza, Ricardo da Silva.
Filtragem de ruídos em imagens digitais / Ricardo da Silva Barboza. -
Recife: O Autor, 2013.
104 folhas, il., gráfs., tabs.

Orientador: Prof. Dr. Rafael Dueire Lins.
Tese (Doutorado) – Universidade Federal de Pernambuco. CTG.
Programa de Pós-Graduação em Engenharia Elétrica, 2013.
Inclui Referências e Apêndice.

1. Engenharia Elétrica. 2. Marcador. 3. Sublinhado. 4. Envelhecimento.
5. Forense. 6. Rastreamento de Objetos. I. Lins, Rafael Dueire.
(Orientador). II. Título.

UFPE

621.3 CDD (22. ed.)

BCTG/2016-148



UNIVERSIDADE FEDERAL DE PERNAMBUCO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

A comissão examinadora da Defesa da Tese de Doutorado

Filtragem de Ruídos em Imagens Digitais

defendida por

Ricardo da Silva Barboza

Considera o candidato APROVADO

Recife, 30 de agosto de 2013.

Banca Examinadora:

Prof. Dr. Rafael Dueire Lins
(orientador e membro titular interno)

Profa. Dra. Maria Lencastre Pinheiro de Menezes Cruz
(membro titular externo)

Prof. Dr. Valdemar Cardoso da Rocha Júnior
(membro titular interno)

Profa. Dra. Judith Kelner
(membro titular externo)

Prof. Dr. Jean Marc-Ogier
(membro titular externo)

Prof. Dr. Daniel Marques Oliveira
(membro titular externo)

Aos meus pais, João Raimundo de Freitas Barboza (*in memoriam*) e Selma da Silva Barboza.

À minha esposa Andréia Cristina Martins Barboza.

Aos meus filhos Brenda, Rebeca e Rafael.

AGRADECIMENTOS

A Deus.

Ao meu orientador Prof. Rafael Dueire Lins, pela paciência e perseverança.

Aos professores da UFPE e da UEA.

Aos meus colegas do DINTER, em especial Raimundo Oliveira e Jucimar Maia Jr.

Aos meus alunos e orientandos.

A coordenação do DINTER em Manaus, Prof. Francis Wagner e Prof. Antenor.

A todos que me ajudarem de maneira direta ou indireta.

Resumo da Tese apresentada à UFPE como parte dos requisitos necessários
para a obtenção do grau de Doutor em Engenharia Elétrica.

Filtragem de Ruídos em Imagens Digitais

RICARDO DA SILVA BARBOZA

Agosto/2013

RESUMO: Entre as etapas que estão presentes em um sistema de processamento digital de imagens encontram-se a aquisição, a segmentação, a descrição e o reconhecimento e interpretação. Esta tese examina como a remoção de ruídos pode trazer benefícios na etapa de segmentação. A filtragem de ruído com conhecimento do que se deseja retirar e/ou manter na imagem resulta na diminuição dos dados que serão processados nas etapas posteriores diminuindo o tempo necessário para o processamento e melhorando a eficiência destes processos. Durante o curso do trabalho são analisados casos de uso onde se demonstra os benefícios da filtragem de ruído na etapa de segmentação em algoritmos originais. No primeiro caso de uso é demonstrada a identificação de artefatos que se desejam remover da imagem final com os exemplos de remoção de marcadores de texto e de sublinhados feitos à mão. Com o conhecimento do comportamento do ruído de marcadores é apresentado um algoritmo para a sumarização de texto envolto por marcadores. Em seguida é mostrado que o ruído pode ser tratado como fonte útil nos casos de análise forense podendo ser utilizado para a restauração de áreas danificadas de documentos. Por fim apresentamos uma melhoria significativa na identificação de gestos pelo método de Viola-Jones através do encaminhamento de partes segmentadas de imagens contendo somente o que é identificado como sendo movimento e pele, melhorando tanto a taxa de reconhecimento de gestos como a quantidade de quadros processados por segundo.

Palavras-chave: Marcador, Sublinhado, Envelhecimento, Forense, Rastreamento de Objetos.

Abstract of the Thesis presented to the UFPE as part of the necessary requirements
for the title of Doctor in Electrical Engineering.

Noise Filtering in Digital Images

RICARDO DA SILVA BARBOZA

August/2013

ABSTRACT: Among the steps that are present in a digital image processing system one can cite the acquisition, segmentation, description and recognition and interpretation. This thesis examines how noise reduction can be beneficial in the segmentation step. The noise filtering with knowledge of what is desired to remove and/or hold in the image results in a decrease of the data to be processed in later steps decreasing the time required for processing and improving the efficiency of these processes. During the course of this work is demonstrated the benefits of filtering noise in the segmentation step and original algorithms are analyzed. The first case is presented to identify whether to remove artifacts from the final image with examples of removal of highlighters and underscores handmade. With the knowledge of the behavior of noise markers an algorithm for text summarization which is surrounded by highlighter is presented. Then it is shown that noise can be treated as a useful source in cases of forensic analysis can be used to restore damaged areas of documents. Finally, is present a significant improvement in the identification of gestures by Viola-Jones method by forwarding segmented images containing only what is identified as moving and skin, improving both the rate of recognition of gestures as the number of processed frames per second.

Keywords: Highlighting, Underline, Aging, Foreense, Object Tracking.

LISTA DE FIGURAS

Figura 1.1: Primeira imagem digital já produzida [90]	16
Figura 2.1: Texto com marcador da cor laranja. Embora o texto original seja de Winston Churchill, a seqüência de letras destacadas pelo marcador remete a outra frase, desta vez de Mahatma Gandhi.	20
Figura 2.2: (a) Imagem com marcador. (b) Imagem processada com Algoritmo 2.1 (limite = 20). (c) Imagem 2.1b binarizada (limiar = 253) para visualização do efeito de borda.	22
Figura 2.3: (a) Documento com marcadores de várias cores. (b) Marcadores removidos após processamento do Algoritmo 2.1.	22
Figura 2.4: (a) Documento com marcador amarelo. (b) Marcador removido após processamento pelo Algoritmo 2.2. (c) Resultado obtido trocando-se a componente vermelha pela componente verde no Algoritmo 2.2.	23
Figura 2.5: (a) Imagem de papel envelhecido com marcador amarelo. (b) Mesma imagem em escala de cinza. (c) Mesma imagem após processamento pelo Algoritmo 2.1.	23
Figura 2.6: (a) Imagem original. (b) Imagem com marcador Pilot. (c) Imagem com marcador Pointer. (d) Imagem com marcador Zolben.	25
Figura 2.7: Comparação de intensidade entre regiões com e sem marcador Pilot da Figura 2.6b. (b) Componente Vermelha. (c) Componente Verde. (d) Componente Azul.	27
Figura 2.8: Comparação de intensidade entre regiões com e sem marcador Pointer da Figura 2.6c. (b) Componente Vermelha. (c) Componente Verde. (d) Componente Azul.	28
Figura 2.9: Comparação de intensidade entre regiões com e sem marcador Zolben da Figura 2.6d. (b) Componente Vermelha. (c) Componente Verde. (d) Componente Azul.	29
Figura 2.10: Variação da componente azul nos marcadores das Figuras 2.7 e 2.9.	29
Figura 2.11: (a) Pixels de tons mais escuros da Figura 2.5a, os mesmos são fracamente afetados pelo selecionador.	30
Figura 2.12: Histogramas da imagem 2.4a. (a) Área não afetada pelo selecionador. (b) Área afetada pelo selecionador.	30
Figura 2.13: (a) Imagem Original. (b) Máscara encontrada pela aplicação do algoritmo de Otsu na componente azul da Figura 2.13a sem os pixels do texto. (c) Máscara final com ruído de sal e pimenta filtrado.	30
Figura 2.14: (a) Imagem após aplicação do primeiro passo do Algoritmo 2.4 (b) Histograma da área afetada pelo marcador amarelo.	31
Figura 2.15: (a) Imagem após aplicação do segundo passo do Algoritmo 2.4. (b) Histograma da área afetada pelo marcador amarelo.	31

Figura 2.16: (a) Imagem após aplicação do terceiro passo do Algoritmo 2.4. (b) Histograma da área afetada pelo marcador amarelo.	32
Figura 2.17: (a) Imagem com selecionador ciano. (b) Histograma normalizado da componente vermelha da imagem. Linha amarela representa a área da textura, linha vermelha representa área afetada pelo selecionador (sem área do texto), o eixo x representa a intensidade da componente. .	33
Figura 2.18: Histograma normalizado da Componente vermelha da Figura 2.17. A linha amarela representa a área da textura, a linha vermelha representa área afetada pelo selecionador e a linha verde representa a área afetada após a aplicação do coeficiente a (sem área do texto), o eixo x representa a intensidade da componente.	35
Figura 2.19: (a) Imagem com selecionador ciano removido. (b) Histograma normalizado da componente vermelha da Figura 2.17a. A linha amarela representa a área da textura, a linha vermelha representa área afetada pelo selecionador após aplicação do coeficiente a . A linha verde representa área afetada pelo selecionador após aplicação do coeficiente b (sem área do texto), o eixo x representa a intensidade da componente.....	35
Figura 2.20: Várias imagens afetadas por selecionadores, e o resultado do processamento da técnica descrita neste trabalho nas respectivas áreas afetadas das imagens.	36
Figura 2.21: Amostra de texto com sublinhado.	37
Figura 2.22: (a) Exemplo de sublinhado que não intercepta o texto. (b) Exemplo de sublinhado que intercepta o texto.	37
Figura 2.23: Cubo representando o espaço de cores RGB. A linha que liga os pontos (0,0,0) e (1, 1, 1) é composta por tons de cinza [23].....	38
Figura 2.24: Resultado do processamento das imagens da Figura 2.22 com o Algoritmo 2.5.	39
Figura 2.25: Máscaras de segmentação obtidas após aplicação de filtro de remoção de sal e pimenta na Figura 2.24.....	39
Figura 2.26: Imagens resultantes com o sublinhado removido.	39
Figura 2.27: Remoção de sublinhados traçados por diversos tipos de canetas. (a) Imagem original. (b) Resultado após aplicação do método apresentado.....	40
Figura 2.28: Documento envelhecido com sublinhado. (a) imagem original. (b) segmentação do ruído por sublinhado. (c) remoção de ruído sal e pimenta. (d) máscara final obtida pela dilatação de (c).	40
Figura 2.29: Documento envelhecido da Figura 2.25a com sublinhado removido.....	41
Figura 2.30: Imagem binarizada com sublinhados tocando e interceptando o texto.....	42
Figura 2.31: Identificação dos componentes do texto da Figura 2.30 pelo algoritmo <i>flood-fill</i> , os retângulos são somente para ilustrar os objetos encontrados.	43
Figura 2.32: Tempo médio de processamento dos algoritmos de afinamento de imagem de uma folha de A4 digitalizada em 300 dpi.....	43

Figura 2.33: (a) Objeto segmentado da imagem original da Figura 2.27. (b) esqueletização do mesmo. (c) remoção da linha. (d) dilatação de (c) com o operador AND e a imagem (a).....	44
Figura 2.34: Imagem com ruídos de linhas e sublinhados removidos.	44
Figura 2.35: Fluxograma do algoritmo proposto nesta seção para retirada de sublinhados em imagens binarizadas.....	45
Figura 3.1: (a) Imagem Original. (b) Resultado do processamento da imagem (a) pelo Algoritmo 3.1. Parâmetros: <i>limiar</i> = 170; <i>distância</i> = 35.....	48
Figura 3.2: Influência do parâmetro <i>limiar</i> na imagem resumo. (a) <i>limiar</i> = 70. (b) <i>limiar</i> = 170. (c) <i>limiar</i> = 210.....	49
Figura 3.3: Gráfico da taxa de acerto de reconhecimento de caracteres com a variação do parâmetro <i>limiar</i>	50
Figura 3.4: (a) Zoom em uma área afetada por marcador amarelo de forma não uniforme. (b) Figura 3.4a processada através do Algoritmo 3.1.....	50
Figura 3.5: Imagem resumo de texto afetado por marcador aplicado de maneira não uniforme e processado pelo Algoritmo 3.1.....	51
Figura 3.6: Figura 3.4a após pré-processamento e aplicação do Algoritmo 3.1.	52
Figura 3.7: Imagem resumo com aplicação do pré-processamento.	52
Figura 3.8: Gráfico da taxa de acerto de reconhecimento caracteres com a variação do parâmetro <i>limiar</i> , após a aplicação da etapa de pré-processamento.....	53
Figura 4.1: (a) Exemplo de traço de caneta azul. (b) Valores de médias e desvios padrões de suas componentes RGB versus o limiar de decisão da separação dos pixels do traço e dos pixels do fundo do papel.	59
Figura 4.2: Distribuições da marca Bic obtidos da Tabela 4.1, as curva mais a esquerda são comparações entre traços da mesma caneta, a curva vermelha comparações de distâncias entre canetas Bic e as 3 curvas mais a direita são comparações de textos entre Bic e as Marcas FaberCastell, Pentel e Pilot.	60
Figura 4.3: Notched Box Plot das distâncias entre os objetos de texto da Marca Bic comparados com as marcas em estudo, as colunas correspondem as distribuições da Figura 4.2.	61
Figura 4.4: Micrografia de exemplos de traços feitos com caneta Bic. (a) e (c) são traços recentes enquanto que (b) e (d) possuem um ano. Os aumentos foram de 30x para (a) e (b) e 200x para (c) e (d).....	62
Figura 4.5: Exemplo de documento de interesse forense.....	63
Figura 4.6: Exemplo de uma certidão de nascimento a partir de 1990, impresso em papel com "marca d'água" e selo numerado com holograma emitido pela Casa da Moeda do Brasil.....	64
Figura 4.7: Exemplos de amostras retiradas em áreas próximas ao centro dos documentos analisados. Na imagem de 43 anos os histogramas das componentes verde e vermelhas são impulsos próximos	

a 255, já na imagem de 50 a normal da componente azul se deslocou a esquerda e o impulso da componente verde começa a tomar forma de uma normal.....	67
Figura 4.8: Gráficos de Média (a) e Desvio Padrão (b) das amostras analisadas em relação a suas idades em anos.	68
Figura 4.9: Comparação da média da intensidade da componente Azul entre duas fontes de documentos. Os quadrados representam amostras individuais de uma segunda fonte de documentos.	68
Figura 4.10: Geração de textura tendo como parâmetro de entrada somente a idade do papel. No caso idade = 40 anos. (a) Amostra real, (b) imagem sintetizada.	69
Figura 5.1: Diagrama resumido da metodologia adotada para o desenvolvimento dos algoritmos deste capítulo.....	73
Figura 5.2: (a) Aumenta volume, (b) Diminui volume, (c) troca canal, (d) Movimenta cursor e (e) seleciona opções na IDTV.....	74
Figura 5.3: Detecção da mão direita aberta e da mão esquerda em punho após as etapas de <i>skin detection</i> e de <i>motion detection</i>	75
Figura 5.4 - <i>Wavelet</i> de Haar.....	77
Figura 5.5 - <i>Haar-Like Features</i> utilizadas neste trabalho.....	77
Figura 5.6: (a) Imagem Original. (b) Adição de linha e coluna para cálculo da imagem integral. ..	80
Figura 5.7: Soma dos pixels utilizando os valores diretamente na imagem de interesse.....	80
Figura 5.8: Definição das regiões em uma imagem integral.....	81
Figura 5.9: Processo de detecção da tonalidade de pele associado ao <i>motion detection</i>	83
Figura 5.10: Barra de navegação do iGesture com a opção Jogo ativada.	85
Figura 5.11: Diagrama do processo de reconhecimento de gestos utilizando motion, skin, Haar cascade e CamShift.	87
Figura 5.12: Diagrama comparativo das performances dos métodos adotados.	88
Figura 5.13: Diagrama comparativo das etapas utilizadas para o reconhecimento de gestos.	89

LISTA DE TABELAS

Tabela 2.1: Alteração de Componentes RGB por marcadores.....	20
Tabela 4.1: Resultados das Médias e Desvio-Padrão das Distâncias entre 4 marcas de canetas a partir de documentos digitalizados em 300 dpi.	60
Tabela 5.1: Comparativo de resultados entre o método deste trabalho e o skin detector padrão.....	86

LISTA DE ALGORITMOS

Algoritmo 2.1: Pseudocódigo para o algoritmo de remoção de marcadores em documentos monocromáticos.	21
Algoritmo 2.2: Pseudocódigo para filtragem de marcador amarelo em documentos monocromáticos.	23
Algoritmo 2.3: Programa em Java do pseudocódigo do Algoritmo 2.1 pronto para utilização com a ferramenta ImageJ.	25
Algoritmo 2.4: Pseudocódigo para filtragem de marcador amarelo em documentos coloridos.	31
Algoritmo 2.5: Pseudocódigo para segmentação inicial da imagem.	38
Algoritmo 3.1: Pseudocódigo para o algoritmo de criação de imagem resumo de partes afetadas por marcadores de um conjunto de imagens.	47
Algoritmo 4.1: Programa em R Script para o cálculo da distância de mahalanobis conforme fórmula 4.1.	58

SUMÁRIO

1. INTRODUÇÃO	16
1.1. OBJETIVO.....	18
1.2. ESTRUTURA DESTA TESE.....	18
2. REMOÇÃO DE RUÍDOS.....	19
2.1. REMOÇÃO DE MARCADORES DE TEXTO.....	21
2.1.1. <i>Remoção de marcador de texto em imagens com fundo monocromático.....</i>	<i>21</i>
2.1.2. <i>Remoção de marcador de texto amarelo em imagens com fundo colorido.....</i>	<i>23</i>
2.1.3. <i>Remoção de marcador de texto em imagens com fundo colorido sem iterações</i>	<i>32</i>
2.2. REMOÇÃO DE SUBLINHADOS	36
2.2.1. <i>Remoção de sublinhados em documentos coloridos</i>	<i>38</i>
2.2.2. <i>Remoção de sublinhados em documentos envelhecidos.....</i>	<i>40</i>
2.2.3. <i>Remoção de sublinhados em documentos binarizados.....</i>	<i>41</i>
2.3. CONCLUSÃO	45
3. GERAÇÃO DE RESUMOS APARTIR DE MARCADORES	46
3.1. GERANDO RESUMO COM TEXTOS REALÇADOS	47
3.2. A RELAÇÃO ENTRE O PARÂMETRO LIMIAR E SUA INFLUÊNCIA NO DESEMPENHO NO PÓS- PROCESSAMENTO POR OCRS	49
3.3. PRÉ-PROCESSAMENTO PARA AJUSTE DE MARCADORES APLICADOS IRREGULARMENTE	50
3.4. CONCLUSÃO	53
4. ANÁLISE FORENSE.....	54
4.1. IDENTIFICAÇÃO DE TRAÇOS DE CANETAS ESFEROGRÁFICAS	55
4.1.1. <i>Distância de Mahalanobis</i>	<i>56</i>
4.1.2. <i>Limiar de decisão dos traços de canela esferográfica</i>	<i>58</i>
4.1.3. <i>Distribuição das distâncias entre os traços de caneta</i>	<i>59</i>
4.1.4. <i>Considerações sobre o envelhecimento da tinta</i>	<i>61</i>
4.2. ESTUDO SOBRE O EFEITO DO ENVELHECIMENTO NA COR DE PAPEL BRANCO.....	62
4.2.1. <i>Características dos documentos analisados</i>	<i>65</i>
4.2.2. <i>Resultados</i>	<i>65</i>
4.2.3. <i>Sintetização de Texturas de Documentos Envelhecidos.....</i>	<i>69</i>
4.3. CONCLUSÃO	69
5. IDENTIFICAÇÃO DE GESTOS MANUAIS PARA MANIPULAÇÃO DE INTERFACES	71
5.1. CENÁRIO DE APLICAÇÃO DO RECONHECIMENTO DE GESTOS	72

5.2.	ENGENHARIA DE USABILIDADE	73
5.3.	TV DIGITAL INTERATIVA	75
5.4.	VISÃO COMPUTACIONAL	76
5.4.1.	<i>Criação dos Classificadores de Gestos</i>	76
5.4.2.	<i>Uso dos Classificadores</i>	81
5.5.	IGESTURE	83
5.6.	OTIMIZAÇÃO DO ALGORITMO DE ACOMPANHAMENTO DE OBJETOS ATRAVÉS DO CAMSHIFT.....	86
5.7.	CONCLUSÃO	89
6.	CONCLUSÕES E TRABALHOS FUTUROS	90
6.1.	TRABALHOS FUTUROS	91
7.	PUBLICAÇÕES	93
8.	REFERÊNCIAS.....	94
APÊNDICE A – CÓDIGO FONTE DO FILTRO DE REMOÇÃO DE MARCADOR EM OUTRAS		
PLATAFORMAS		
	CÓDIGO FONTE DO FILTRO DE REMOVEDOR DE MARCADOR EM OBJECTIVE-C.....	101
	CÓDIGO-FONTE DO FILTRO DO REMOVEDOR DE MARCADOR EM DELPHI2009	103

1. INTRODUÇÃO

"By relieving the brain of all unnecessary work, a good notation sets it free to concentrate on more advanced problems, and in effect increases the mental power of the race."

Alfred North Whitehead,

"An introduction to mathematics".

A primeira imagem digital surgiu em 1957, quando Russell Kirsch digitalizou uma foto de seu filho de três meses com resolução de 176×176 pixels. A baixa resolução dessa foto, mostrada na Figura 1.1 deveu-se ao fato do computador de então ser incapaz de realizar armazenamento em alta densidade [90]. Hoje mesmo na manipulação de vídeo, encontram-se padrões que digitalizam e armazenam imagens com resoluções de até 4096×2304 pixels (conhecidos como 4K).



Figura 1.1: Primeira imagem digital já produzida [90]

O processamento digital de imagens é uma área de pesquisa em constante expansão. A cada dia surgem novos dispositivos, com maior capacidade de processamento e memória, conseqüentemente de armazenamento de imagens, possibilitando novas aplicações e trazendo dificuldades e problemas relevantes e originais a serem resolvidos. As imagens digitais hoje estão onipresentes em diversas áreas da atividade humana, desde o laser onde as fotos convencionais foram, há mais de uma década, substituídas por digitais até na medicina onde os seculares Raios-X (que eram impressos em “chapas” de acetato) foram definitivamente trocadas por imagens geradas digitalmente visualizadas por médicos na tela dos seus computadores e integrando os prontuários digitais dos pacientes.

Tal revolução também atingiu as bibliotecas. O papel, um meio milenar de transmissão de informação em livros, revistas, jornais, sofre desgaste natural, pode ser danificado pelo manuseio incorreto, exige grande espaço de armazenamento e está limitado em acessibilidade. A digitalização desse legado apresenta inúmeras vantagens. Ela possibilita, por exemplo, a reprodução de documentos de maneira fácil. O documento no formato digital, aliado a Internet, possibilita uma maior divulgação a custo e tempo menores, comparado ao transporte físico.

Caso o formato digital possua texto é possível a pesquisa por palavras-chave do mesmo. Vale ressaltar que essa diminuição de custo e facilidade na divulgação, torna o acesso à informação mais democrática. Porém, a manipulação de documentos em formato digital traz desafios a serem vencidos. Por exemplo, documentos danificados por idade, fungos, manchas e anotações de usuários podem ser “restaurados”, inclusive “completando” partes de informação perdidas. A referência [2] propõe uma forma ampla de classificação dos ruídos que atingem documentos físicos e digitais, bem como o relacionamento entre eles, no mesmo trabalho são citados métodos para evitar e remover tais ruídos.

Apesar de extensa pesquisa realizada em [2] e pelo autor desta tese, não foi encontrada técnica para remoção de ruídos como os criados por marcadores de texto e sublinhados que interceptavam o texto de forma destrutiva.

Landon [94] relata a existência de projetos de digitalização de livros em larga escala que necessitam de técnicas de remoção de ruído específicas para os problemas encontrados nos documentos. Visto que existem entre 74 e 175 milhões de títulos únicos de livros no mundo [82], os esforços para a diminuição do ruído presente nos originais é de grande valia para os esforços de armazenagem, indexação e recuperação da informação destes grandes projetos.

A pesquisa conduzida nesta tese iniciou com vistas a encontrar técnicas de remoção dos ruídos resultados de marcadores de texto e de sublinhados feitos à mão e prosseguiu buscando novos algoritmos de processamento digital de imagens, visão computacional e análise forense. Em todos levando em consideração a etapa de remoção de ruídos.

Na área de análise forense [38] é apresentada uma modelagem do envelhecimento do papel branco durante meio século. Nela se observa a lenta alteração da cor do papel mapeada em uma curva, fruto

da pesquisa. Foi possível também formular um método para a comparação de escrita realizada por canetas. Os métodos propostos necessitam somente de um scanner de mesa para a entrada das imagens que serão analisadas, não sendo necessários equipamentos caros ou destruição da fonte a ser analisada.

Em visão computacional, um algoritmo rápido para a identificação de gestos com vista à manipulação de interface gráfica foi formulado, em conjunto, com sua implementação e teste em um ambiente de interação com TV digital. O resultado proposto não necessita de equipamento específico como outras propostas [89] e pode ser embarcado em *setup boxes*.

1.1. Objetivo

O objetivo deste trabalho é o de apresentar a importância da etapa de remoção de ruídos em imagens digitais através de casos de uso que mostram a resolução dos problemas de supressão de marcadores de texto e de sublinhados manuais e da filtragem de objetos para posterior processamento por visão computacional e análise forense.

1.2. Estrutura desta tese

Esta tese é composta, além desta introdução, por mais cinco outros capítulos. O capítulo 2 apresenta algoritmos para remoção de ruídos em imagens de documentos digitalizados, especificamente ruídos relacionados a marcadores de texto e sublinhados. O Capítulo 3 utiliza idéias do capítulo 2 para a geração de imagens resumos das áreas afetadas por marcadores. O Capítulo 4 descreve a utilização de imagens digitalizadas para análise forense de canetas esferográficas e de papel envelhecido. O Capítulo 5 apresenta dois algoritmos para a identificação de gestos manuais para a manipulação de interfaces. O Capítulo 6 conclui este trabalho com as considerações finais e propostas de trabalhos futuros.

2. REMOÇÃO DE RUÍDOS

Informação é tudo aquilo que muda o meu comportamento futuro. O resto é ruído.

Stephen Kanitz

Frequentemente leitores fazem anotações em documentos, motivados por razões diferentes. À exceção de poucos casos, como no exemplo em que Fermat fez anotações no livro Aritmética de Diofanto de Alexandria [91], resultando em pesquisas durante mais de 300 anos, essas anotações adicionam pouco ou nada para a informação do documento por si só.

O sublinhado é uma espécie de indicação no qual o leitor, de alguma forma, procura enfatizar partes de um texto para referência futura.

Em 1962, a caneta com ponta de feltro foi inventada no Japão por Yukio Horie [1]. Este tipo de caneta criou a possibilidade de leitores destacarem os seus textos com essas canetas marcadoras, não se limitando ao sublinhado convencional.

Uma caneta selecionadora, caneta marcadora de feltro, ou simplesmente um marcador, é uma caneta que tem a sua própria fonte de tinta e geralmente uma ponta feita de um material poroso, tal como o feltro ou nylon. Marcadores, como o usado nesta frase, são marcadores permanentes preenchidos com tinta fluorescente transparente, usados para cobrir textos, enfatizando tal conteúdo. Marcadores são frequentemente utilizados para fazer anotações em livros didáticos sendo populares hoje em dia. A Figura 2.1 mostra um texto com partes destacadas por um marcador laranja.

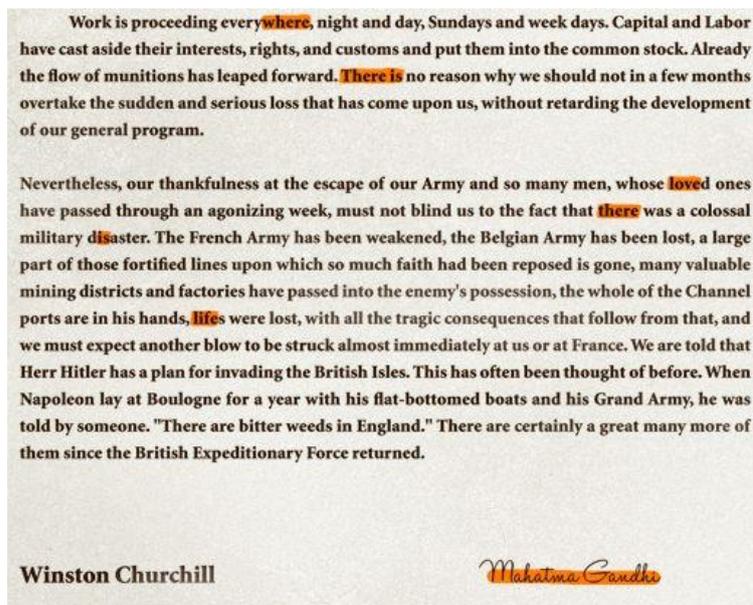


Figura 2.1: Texto com marcador da cor laranja. Embora o texto original seja de Winston Churchill, a sequência de letras destacadas pelo marcador remete a outra frase, desta vez de Mahatma Gandhi¹.

Muitos marcadores vêm em cores brilhantes, muitas vezes fluorescentes, que brilham sob a luz de lâmpadas ultravioleta que emitem comprimentos de onda próximos à luz visível entre 380 e 420 nm, comumente conhecidos como luz negra. A cor mais comum para marcadores é a amarela, mas eles também são encontrados em azul, verde, laranja e magenta. A Tabela 2.1 apresenta as cores mais comuns de marcadores. Para cada linha da mesma tabela é apresentado qual(is) componente(s) RGB o marcador reduz a intensidade de luminosidade.

Tabela 2.1: Alteração de Componentes RGB por marcadores

Marcador	Cor	Componente Afetada
	Amarela	Azul
	Azul	Vermelho/Verde
	Verde	Vermelho/Azul
	Laranja	Verde/Azul
	Magenta	Vermelho/Verde/Azul

O realce do texto por selecionadores pode ser visto como anotação pessoal, que fisicamente pode danificar o documento original. Uma taxonomia recente realizada por Lins [2] sobre ruídos em documentos incluiu a marcação de texto como "ruído físico". Ainda segundo Lins [2], a literatura

¹ Retirado de <http://cargocollective.com/diegorosendo/Sharpie-Marcador-de-Texto>

técnica não oferece nenhuma solução para a remoção de realce em imagens digitalizadas de documentos.

2.1. Remoção de Marcadores de Texto

A remoção de marcadores de texto é mais complexa do que se pode imaginar à primeira vista. Isto se deve ao processo de envelhecimento da tinta e pelo fato de que há casos em que a mesma não interage uniformemente com o papel.

Para a remoção de selecionadores há duas patentes que tratam do assunto [9] [10], mas o texto delas não mostra resultados reais nem valores práticos para a sua execução, citando somente um caminho vago para seu desenvolvimento. Assim é difícil estabelecer uma forma de comparação entre elas e o método aqui proposto.

2.1.1. Remoção de marcador de texto em imagens com fundo monocromático

Como se pode observar na Tabela 2.1, a maioria dos marcadores de texto disponível comercialmente afeta um ou mais componentes de cores do documento original. Análises realizadas demonstraram que o marcador diminui o valor da intensidade do componente de cor original em relação às áreas não marcadas. Esta observação foi o ponto de partida para o desenvolvimento de um algoritmo para a remoção de realce em documentos monocromáticos mostrado no Algoritmo 2.1.

```
Para todos os pixels  $P(i) = (Red_i, Green_i, Blue_i)$  na imagem faça  
   $rg(i) = |Red_i - Green_i|$ ;  
   $rb(i) = |Red_i - Blue_i|$ ;  
   $gb(i) = |Green_i - Blue_i|$ ;  
  se  $((rg(i) > limite) \text{ or } (rb(i) > limite) \text{ or } (gb(i) > limite))$  then  
    se  $(Red_i \geq Blue_i)$  e  $(Red_i \geq Green_i)$  então  
       $P(i) \leftarrow (Red_i, Red_i, Red_i)$ ;  
    se  $(Green_i \geq Red_i)$  e  $(Green_i \geq Blue_i)$  então  
       $P(i) \leftarrow (Green_i, Green_i, Green_i)$ ;  
    se  $(Blue_i \geq Red_i)$  e  $(Blue_i \geq Green_i)$  então  
       $P(i) \leftarrow (Blue_i, Blue_i, Blue_i)$ ;
```

Algoritmo 2.1: Pseudocódigo para o algoritmo de remoção de marcadores em documentos monocromáticos.

No Algoritmo 2.1 para cada pixel da imagem é verificado se algum dos componentes RGB excede um limite pré-definido. Se assim for, todos os outros componentes são trocados para o valor do componente com o maior valor. A Figura 2.2 apresenta os resultados de uma execução do Algoritmo 2.1.

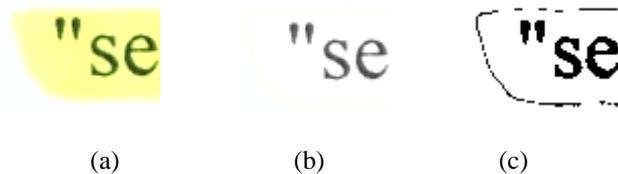


Figura 2.2: (a) Imagem com marcador. (b) Imagem processada com Algoritmo 2.1 (limite = 20). (c) Imagem 2.1b binarizada (limiar = 253) para visualização do efeito de borda.

O parâmetro limite influencia o resultado do Algoritmo 2.1 à medida que valores elevados tendem a afetar menos pixels na imagem produzindo um efeito “borda” na fronteira das áreas de destaque (Figura 2.2 b e c). Através da análise de imagens que foram submetidas ao Algoritmo 2.1 e alterado o parâmetro limite foi encontrado que ajustando o seu valor para 4 o efeito borda não era mais visível.

A Figura 2.3a mostra partes de texto destacado com marcadores de várias cores. Após aplicação do Algoritmo 2.1 (limite = 4), para filtrar a marcação, o resultado pode ser encontrado na Figura 2.3b. Em todos os casos, houve remoção do marcador. O marcador de cor magenta deixou alguns vestígios, no entanto.

01	Aprovação em disciplina de graduação extracurricular	01	Aprovação em disciplina de graduação extracurricular
02	Curso de Inglês	02	Curso de Inglês
03	Estágio extracurricular	03	Estágio extracurricular
04	Iniciação Científica	04	Iniciação Científica
05	Monitoria	05	Monitoria
06	Palestrante ou Expositor em cursos e mini-cursos	06	Palestrante ou Expositor em cursos e mini-cursos
07	Palestras, seminários, congressos, etc (ouvinte)	07	Palestras, seminários, congressos, etc (ouvinte)
08	Participação como Estagiário em Projeto de Pesquisa	08	Participação como Estagiário em Projeto de Pesquisa
09	Participação como Expositor em oficinas	09	Participação como Expositor em oficinas
10	Participação em Cursos de Extensão ou Treinamento	10	Participação em Cursos de Extensão ou Treinamento
11	Participação em Empresa Júnior	11	Participação em Empresa Júnior
12	Participação em Organização de eventos	12	Participação em Organização de eventos
13	Participação em projetos de extensão	13	Participação em projetos de extensão
14	Publicação de Artigo Completo em meio eletrônico	14	Publicação de Artigo Completo em meio eletrônico
15	Publicação de Artigo Completo impresso	15	Publicação de Artigo Completo impresso
16	Publicação de Artigo Resumido impresso	16	Publicação de Artigo Resumido impresso
17	Publicação Resumo com Pôster em meio eletrônico	17	Publicação Resumo com Pôster em meio eletrônico
18	Publicação Resumo com Pôster impresso	18	Publicação Resumo com Pôster impresso
19	Voluntariado	19	Voluntariado

Figura 2.3: (a) Documento com marcadores de várias cores. (b) Marcadores removidos após processamento do Algoritmo 2.1.

O Algoritmo 2.1 pode ser otimizado se a cor do marcador for conhecida *a priori*. Por exemplo, se o marcador utilizado for amarelo, cor amplamente utilizada para a seleção de texto, sabe-se que apenas a componente azul é alterada em documentos monocromáticos (como mostrado na tabela 2.1). Assim, a comparação realizada no Algoritmo 2.1 necessita testar somente a distância da intensidade da componente azul para a componente vermelha ou verde. O pseudocódigo otimizado para este caso está no Algoritmo 2.2. Apesar de se ter utilizado a componente vermelha, trocando-a pela componente verde irá trazer os mesmos resultados no presente caso (Figura 2.4c).

Para todos os pixels $P(i) = (Red_i, Green_i, Blue_i)$ na imagem faça if ($ Red_i - Blue_i > limite$) then $P(i) \leftarrow (Red_i, Red_i, Red_i)$
Algoritmo 2.2: Pseudocódigo para filtragem de marcador amarelo em documentos monocromáticos.

A aplicação do Algoritmo 2.2 na imagem da Figura 2.4a (limite = 4) resulta na imagem da Figura 2.4b, onde se pode observar que o marcador amarelo foi adequadamente removido.

```
"L I G A / D E S L I G A
"-----
"sequencia deve ser feita sem pausa no modo
```

(a)

```
"L I G A / D E S L I G A
"-----
"sequencia deve ser feita sem pausa no modo
```

(b)

```
"L I G A / D E S L I G A
"-----
"sequencia deve ser feita sem pausa no modo
```

(c)

Figura 2.4: (a) Documento com marcador amarelo. (b) Marcador removido após processamento pelo Algoritmo 2.2. (c) Resultado obtido trocando-se a componente vermelha pela componente verde no Algoritmo 2.2.

Ambos, os Algoritmos 2.1 e 2.2, são de simples tradução para ferramentas de processamento digital de imagens. Seguindo as orientações apresentadas em [3] [4] o programa foi traduzido para a linguagem Java (Algoritmo 2.3) utilizada na ferramenta ImageJ [5]. Tal ferramenta é escrita em Java, possui código aberto e uma ampla comunidade de apoio aos que desejam testar novas técnicas de processamento digital de imagens através de plug-ins.

2.1.2. Remoção de marcador de texto amarelo em imagens com fundo colorido

Os algoritmos apresentados até o momento funcionam para imagens monocromáticas digitalizadas em cores. Esta seção tem por objetivo apresentar a proposta de remoção de selecionador amarelo em imagens com fundo colorido, como a mostrada na Figura 2.5a.

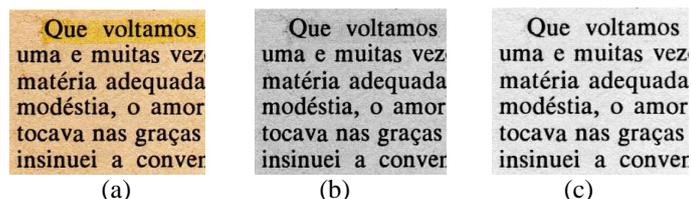


Figura 2.5: (a) Imagem de papel envelhecido com marcador amarelo. (b) Mesma imagem em escala de cinza. (c) Mesma imagem após processamento pelo Algoritmo 2.1.

A aplicação direta do Algoritmo 2.1 em uma imagem colorida resulta em uma imagem em tons de cinza com o marcador removido. A Figura 2.5 mostra um exemplo de um documento com a cor de fundo convertido em escala de cinza padrão e processada pelo Algoritmo 2.1. Pode-se observar que a imagem obtida a partir do Algoritmo 2.1 removeu a sombra do marcador que aparece na Figura 2.5b.

Como se pode observar na Figura 2.5, o Algoritmo 2.1 não é adequado para imagens com fundo colorido. Para o desenvolvimento de uma nova solução, inicialmente foi analisado o efeito do marcador amarelo em imagens com degradê em cinza.

```
import ij.*;
import ij.process.*;
import ij.gui.*;
import java.lang.*;
import java.awt.*;
import ij.plugin.filter.*;

public class removedorMarcacao implements PlugInFilter {
    ImagePlus imp;
    static final int R = 0, G = 1, B = 2;

    public int setup(String arg, ImagePlus imp) {
        this.imp = imp;
        return DOES_RGB;
    }

    public void run(ImageProcessor ip) {
        ColorProcessor cp = (ColorProcessor) ip;
        int[] RGB = new int[3];
        int rg, rb, gb, limite = 4;

        for (int v = 0; v < cp.getHeight(); v++){
            for (int u = 0; u < cp.getWidth(); u++){
                cp.getPixel(u,v,RGB);
                rg = Math.abs(RGB[R] - RGB[G]);
                rb = Math.abs(RGB[R] - RGB[B]);
                gb = Math.abs(RGB[G] - RGB[B]);
                if ((rg > limite) ||
                    (rb > limite) ||
                    (gb > limite)) {
                    if ((RGB[R] >= RGB[B]) && (RGB[R] >= RGB[G])) {
                        RGB[B] = RGB[R];
                        RGB[G] = RGB[R];
                    }
                    if ((RGB[B] >= RGB[R]) && (RGB[B] >= RGB[G])) {
                        RGB[R] = RGB[B];
                        RGB[G] = RGB[B];
                    }
                    if ((RGB[G] >= RGB[R]) && (RGB[G] >= RGB[B])) {
                        RGB[R] = RGB[G];
                        RGB[B] = RGB[G];
                    }
                }
                cp.putPixel(u,v, RGB);
            }
        }

        ip.invert();
    }
}
```

```

        imp.updateAndDraw();
        IJ.wait(500);
        ip.invert();
        imp.updateAndDraw();
    }
}

```

Algoritmo 2.3: Programa em Java do pseudocódigo do Algoritmo 2.1 pronto para utilização com a ferramenta ImageJ.

A Figura 2.6 apresenta uma tarja com 32 tons de cinza indo do preto ao branco. Após a impressão da mesma, uma linha foi feita com marcadores de três fabricantes diferentes.

A análise foi realizada com amostras de diferentes áreas com e sem marcadores da tarja. O resultado foi traçado nos histogramas mostrados nas Figuras 2.7, 2.8 e 2.9, onde se nota que a componente vermelha não sofreu alteração pelo marcador amarelo. A componente verde sofreu uma atenuação suave, e a componente azul sofreu uma atenuação forte nos tons mais claros. Este comportamento é ligado ao uso dos marcadores, no qual tons mais claros, geralmente de fundo de papel sofrem maiores alterações, enquanto os tons mais escuros, ligados ao texto permanecem inalterados.

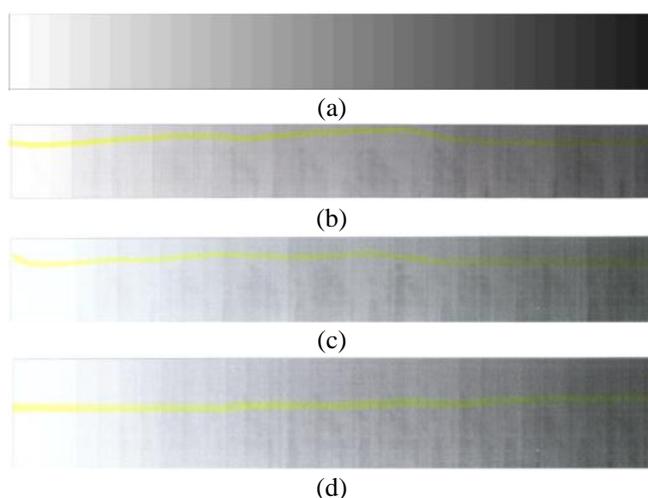


Figura 2.6: (a) Imagem original. (b) Imagem com marcador Pilot. (c) Imagem com marcador Pointer. (d) Imagem com marcador Zolben.

Reunindo a componente azul das Figuras 2.7c e 2.9c na Figura 2.10 pode-se notar que a mesma é afetada de forma similar. Desta forma pode-se mapear a atenuação causada pelo marcador na componente azul como:

$$g(x) = \frac{b-x}{b-a} f(a) + \frac{x-a}{b-a} f(b) \quad (2.1)$$

onde:

$$b = 255 \text{ (branco).}$$

$a = 0$ (preto).

$f(a) = 0$; (valor da intensidade do pixel preto não alterado após o marcador).

$f(b) = 165$; (a média do valor da componente azul no fundo branco após ter sido afetada pelo marcador amarelo).

x = Intensidade Azul inicial

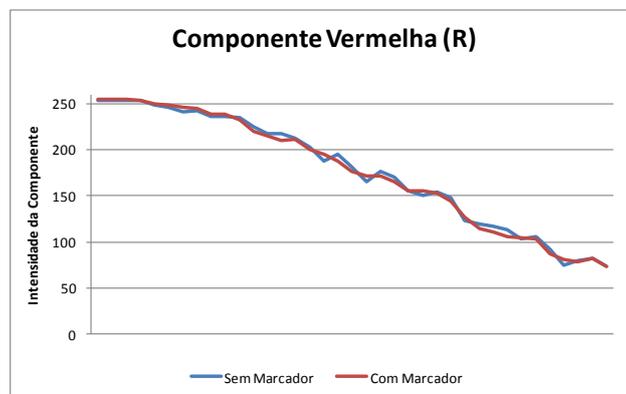
$g(x)$ = Intensidade Azul final

Assim, a fórmula para restabelecer a intensidade original da componente azul da área afetada pelo marcador é:

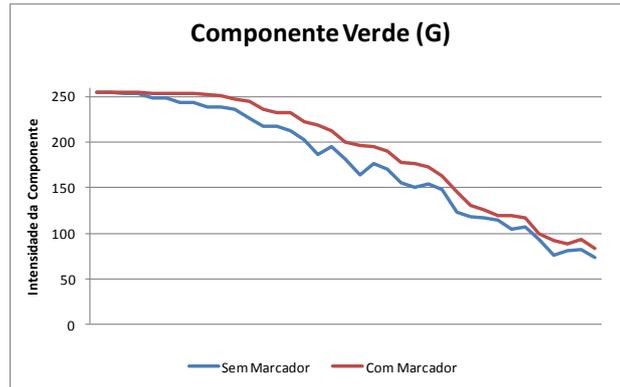
$$g(x) = (x / 165) * 255 \quad (2.2)$$

No próximo passo é necessário identificar a área destacada de amarelo para a qual a componente azul será corrigida. No caso da imagem mostrada na Figura 2.5a, é necessário a divisão em três áreas: o texto, a textura sem marcador, e a área realçada de amarelo.

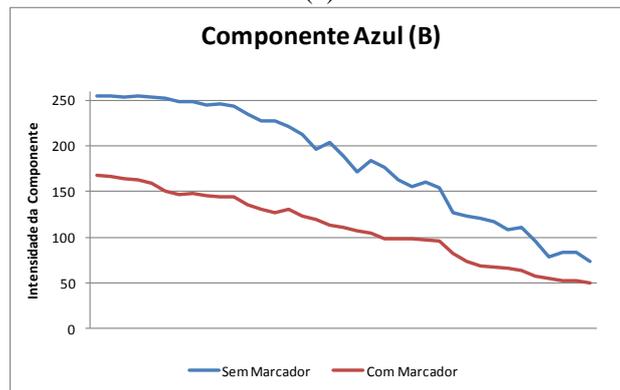
Em geral, a separação do texto e fundo em documentos degradados é uma tarefa complexa [2].



(a)

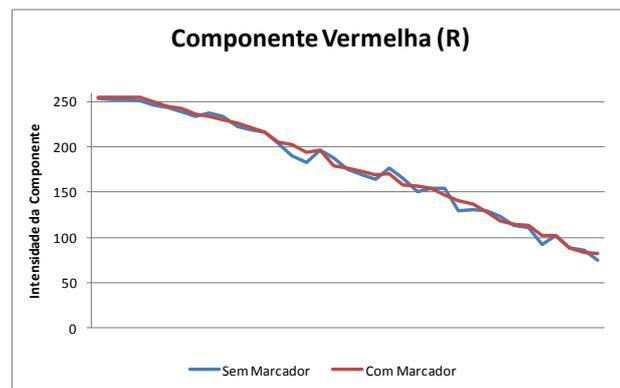


(b)

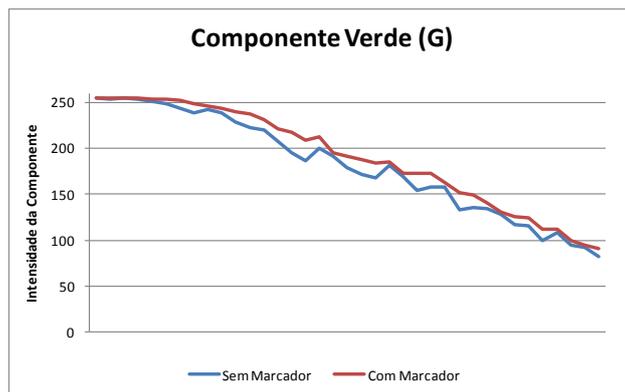


(c)

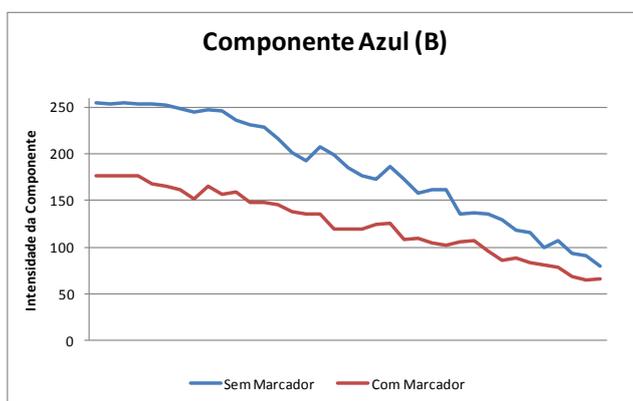
Figura 2.7: Comparação de intensidade entre regiões com e sem marcador Pilot da Figura 2.6b. (b) Componente Vermelha. (c) Componente Verde. (d) Componente Azul.



(a)

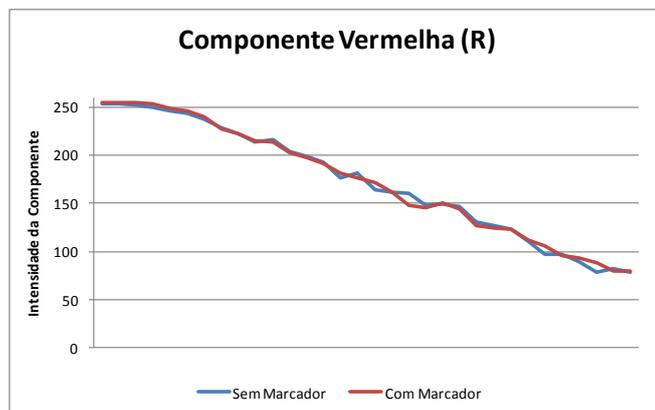


(b)

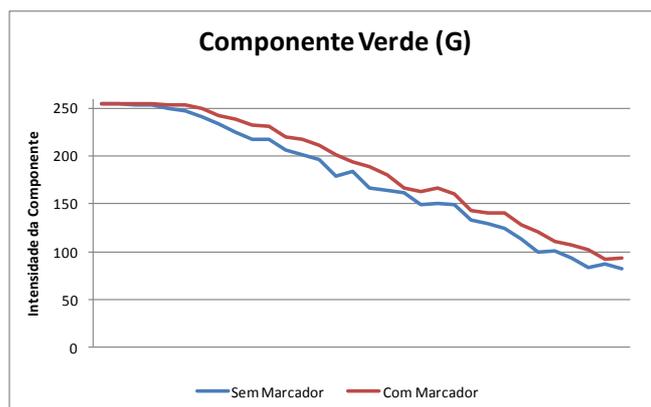


(c)

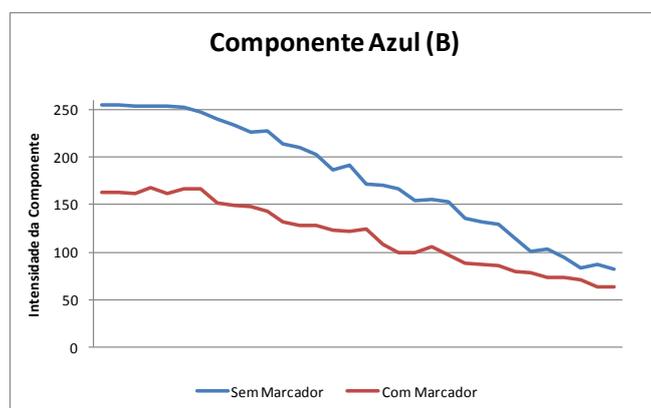
Figura 2.8: Comparação de intensidade entre regiões com e sem marcador Pointer da Figura 2.6c. (b) Componente Vermelha. (c) Componente Verde. (d) Componente Azul.



(a)



(b)



(c)

Figura 2.9: Comparação de intensidade entre regiões com e sem marcador Zolben da Figura 2.6d. (b) Componente Vermelha. (c) Componente Verde. (d) Componente Azul.

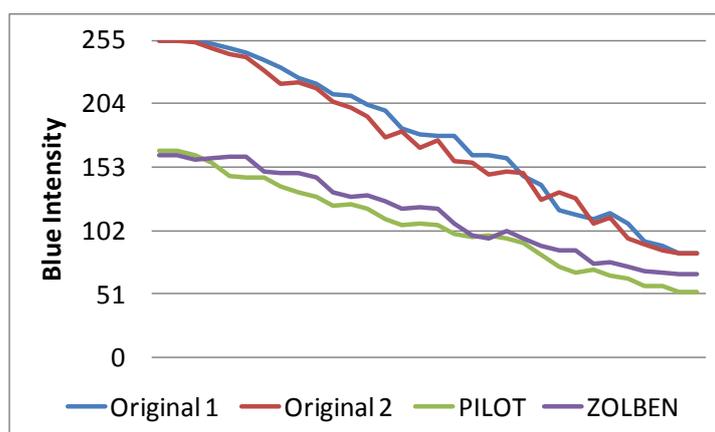


Figura 2.10: Variação da componente azul nos marcadores das Figuras 2.7 e 2.9.

A área de texto é formada por pixels com tons escuros com uma intensidade média de menos do que 128, como mostrado na Figura 2.11. Uma forma automática para encontrar o limiar de binarização da imagem utiliza o algoritmo de Otsu [7]. O mesmo pressupõe que a imagem possui

duas classes e pixels que correspondem a um histograma bimodal. Em imagem com alto grau de degradação uma opção é a utilização de algoritmos como o proposto em [8], que possuem bons resultados com documentos com ruídos advindos do verso da imagem.

Que voltamos
 uma e muitas vez
 matéria adequada
 modestia, o amor
 tocava nas graças
 insinuei a conver

Figura 2.11: (a) Pixels de tons mais escuros da Figura 2.5a, os mesmos são fracamente afetados pelo selecionador.

Para encontrar a segunda área da imagem, a que foi afetada pelo marcador, foi verificado que imagens como a que está sendo analisada na Figura 2.5 (documentos envelhecidos), possuem uma concentração de pixels verdes e vermelhos nos tons mais altos, como mostrado na Figura 2.12, nas áreas não afetadas pelo selecionador. Nas áreas afetadas pelo selecionador há uma mudança na concentração de pixels azuis, os mesmos são deslocados para a esquerda, fazendo com que haja o aparecimento de um histograma bimodal na componente azul, aplicando o algoritmo de Otsu somente nesta componente, e ignorando o texto já encontrado anteriormente, resulta na Figura 2.13b, que após a aplicação de um filtro de remoção de ruído sal e pimenta [14], encontra-se a área afetada pelo selecionador na Figura 2.13c.

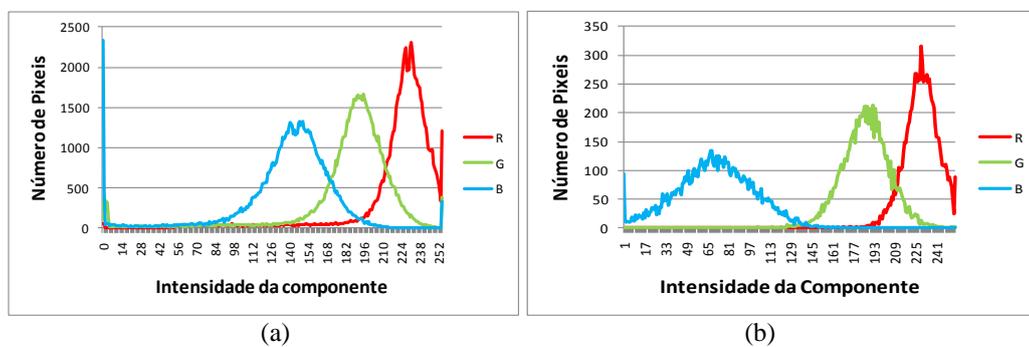


Figura 2.12: Histogramas da imagem 2.4a. (a) Área não afetada pelo selecionador. (b) Área afetada pelo selecionador.

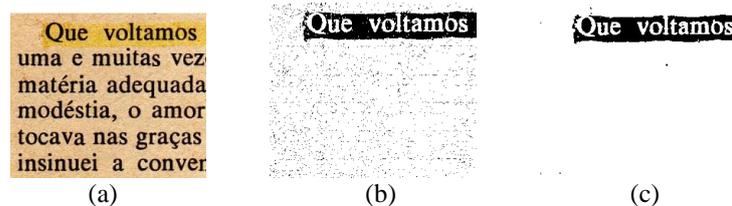


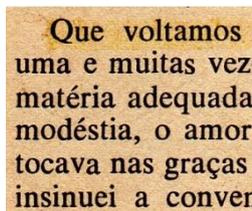
Figura 2.13: (a) Imagem Original. (b) Máscara encontrada pela aplicação do algoritmo de Otsu na componente azul da Figura 2.13a sem os pixels do texto. (c) Máscara final com ruído de sal e pimenta filtrado.

O algoritmo para remoção do marcador age na área da Figura 2.5a relativa à máscara encontrada na Figura 2.13c. O mesmo altera a componente azul pela aplicação da fórmula (2.2).

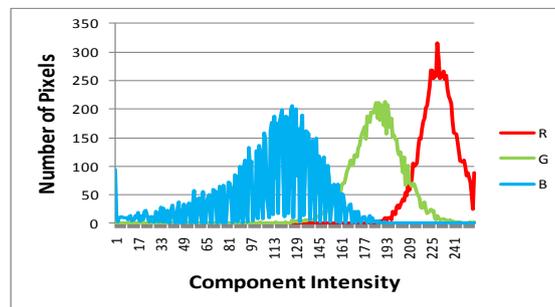
Para todos os pixels $P(i) = (Red_i, Green_i, Blue_i)$ na área afetada pelo selecionador amarelo faça $P(i) \leftarrow (Red_i, Green_i, (Blue_i/165) * 255)$

Algoritmo 2.4: Pseudocódigo para filtragem de marcador amarelo em documentos coloridos.

Aplicando o Algoritmo 2.4 na área afetada pelo selecionador da Figura 2.5a, foi obtida a Figura 2.14a, nela o feito do selecionador, apesar de enfraquecido, ainda está visível, mas o histograma da componente azul se deslocou para próximo a área não afetada pelo selecionador. Desta forma o Algoritmo 2.4 foi aplicado sucessivamente gerando as imagens e histogramas das Figuras 2.15 e 2.16.

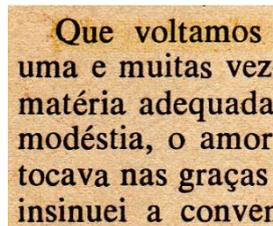


(a)

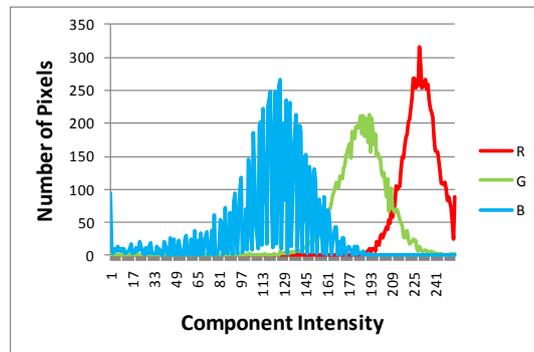


(b)

Figura 2.14: (a) Imagem após aplicação do primeiro passo do Algoritmo 2.4 (b) Histograma da área afetada pelo marcador amarelo.

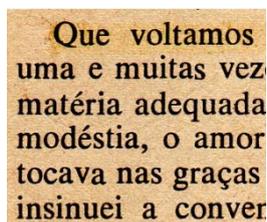


(a)

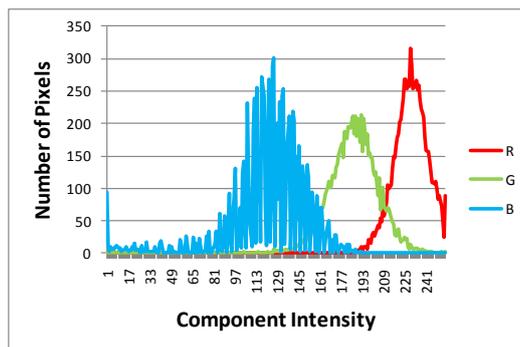


(b)

Figura 2.15: (a) Imagem após aplicação do segundo passo do Algoritmo 2.4. (b) Histograma da área afetada pelo marcador amarelo.



(a)



(b)

Figura 2.16: (a) Imagem após aplicação do terceiro passo do Algoritmo 2.4. (b) Histograma da área afetada pelo marcador amarelo.

Pode-se observar que após a terceira aplicação do Algoritmo 2.4, resultando na Figura 2.16a, a área com o selecionador está mais fraca.

O método proposto nesta seção e publicado em [11] funciona através de iterações. Na próxima seção será apresentado um método mais geral que trabalha com a análise dos histogramas das imagens não sendo necessárias iterações para o resultado final.

2.1.3. Remoção de marcador de texto em imagens com fundo colorido sem iterações

A remoção de marcadores é uma tarefa longe de ser simples, complicando-se à medida que a tinta interage com o substrato em que foi depositada e é influenciada pelo tempo, condições de armazenamento e componentes químicos utilizados em sua fabricação.

Ambas as sessões anteriores apresentaram formas para a eliminação de selecionadores, e foram as primeiras a serem publicadas no meio científico para esse fim [11].

Esta seção generaliza o processo de remoção de marcadores em documentos coloridos para outras cores de marcadores além do amarelo, e retira a necessidade do processo ser incremental.

Os histogramas da componente azul nas Figuras 2.14b, 2.15b e 2.16b possuem um comportamento no qual a envoltória possui descontinuidades, tal comportamento também é encontrado em histogramas de imagens que sofreram alteração por realce de contraste. Esta similaridade foi o ponto de partida para o processo de pesquisa apresentado nesta seção.

O realce de contraste é uma técnica amplamente utilizada em processamento digital de imagens desde áreas médicas [12] à de satélites [13]. Tal técnica mapeia os tons de uma imagem de um intervalo a outro, alterado os limites dos tons da imagem original, através de funções lineares, quadráticas, logarítmicas, entre outras [14].

A escolha da função de transformação que reverterá o ruído será realizada pela análise do histograma das imagens.

Para fins de comparação dos histogramas, geraremos os histogramas normalizados para visualizar as diferenças entre as áreas com marcadores e sem marcadores, visto que estas áreas são geralmente formadas por uma quantidade de pixels diferentes.

O histograma normalizado pode ser visto como uma função densidade de probabilidade. Para uma dada imagem X , a função densidade de probabilidade $p(X_k)$ é definida como:

$$p(X_k) = n^k / n \quad (2.3)$$

Para $k = 0, 1, \dots, L - 1$, onde n^k representa o número de vezes que o nível X_k apareceu na imagem X e n é o número total de pixels na área em estudo [15].

O trabalho desenvolvido nas sessões anteriores permitiu a observação de que existe uma relação linear entre as áreas afetadas por marcadores e as áreas não afetadas por marcadores. Desta forma, a escolha de uma função linear adequada $y = ax + b$, é a chave para cancelar o efeito do selecionador.

Outro exemplo, típico das imagens em estudo, pode ser visualizado na Figura 2.17a, onde uma área foi afetada com selecionador ciano. O histograma normalizado é apresentado pela linha vermelha (Figura 2.17b) e o amarelo representa a área do papel não afetada pelo selecionador. Ambos excluem áreas textuais do documento, devido ao fato que marcadores não alteram significativamente a sua aparência [16].

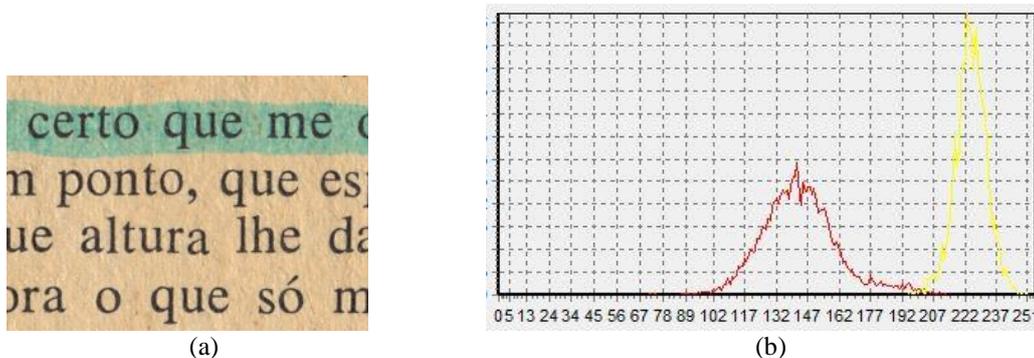


Figura 2.17: (a) Imagem com selecionador ciano. (b) Histograma normalizado da componente vermelha da imagem. Linha amarela representa a área da textura, linha vermelha representa área afetada pelo selecionador (sem área do texto), o eixo x representa a intensidade da componente.

Como observado nas distribuições da Figura 2.17b, o marcador ampliou a distribuição da componente vermelha e alterou sua média para tons mais escuros, ou seja, ambos média e desvio padrão da amostra com o marcador foram alterados em relação à área sem marcador.

Um comportamento semelhante ocorreu com as componentes verde e azul entre as áreas afetadas pelo marcador e não afetadas pelo marcador.

Também se pode observar que as distribuições possuem um formato de distribuição normal, devido ao envelhecimento natural que ocorreu com o papel, encontramos este comportamento em outras amostras de papel envelhecido também.

Como ambas as linhas amarela e vermelha da Figura 2.17b apresentam um formato de distribuição normal, o marcador como já mostrado na seção anterior realizou uma transformação linear que também pode ser visualizado nos histogramas.

Como o objetivo é encontrar a função inversa para desfazer o efeito do marcador, e cada componente é afetada de forma diferente, não só pela cor do marcador mas também por outros fatores já citados (tipo de papel, sua gramatura e tempo que marcador foi utilizado no papel), uma equação única, com valores fixos, se torna uma tarefa complexa. Ao invés disto serão utilizados parâmetros das próprias regiões afetadas e não afetadas pelo marcador para reverter seus efeitos.

Tendo como ponto de partida os histogramas normalizados como os da Figura 2.17 (calculados através da fórmula 2.3) e utilizando as seguintes propriedades do valor esperado $E(X)$ e do desvio padrão $\sigma(X)$ [17]:

$$\sigma(aX + b) = a\sigma(X) \quad (2.4)$$

$$E(aX + b) = aE(X) + b \quad (2.5)$$

Sendo $\sigma_1(X)$ o desvio padrão original de alguma componente RGB da textura original, o selecionador afetará a mesma de forma que o desvio padrão da área afetada será $\sigma_2(aX)$, veja que b que é um deslocamento da intensidade da componente não afetará o valor do desvio-padrão.

Sendo assim pode-se utilizar a equação 2.4 para encontrar um valor aproximado a através de da equação 2.6 que será parte da equação linear $y = ax + b$ e retornar ao valor do desvio padrão original multiplicando a pela intensidade da componente em questão dos pixels da área afetada.

$$a = \sigma_1(X) / \sigma_2(X) \quad (2.6)$$

Aplicando o coeficiente a ao histograma normalizado, representado pela linha vermelha da Figura 2.17b, obtêm-se o histograma representado pela linha verde da Figura 2.18.

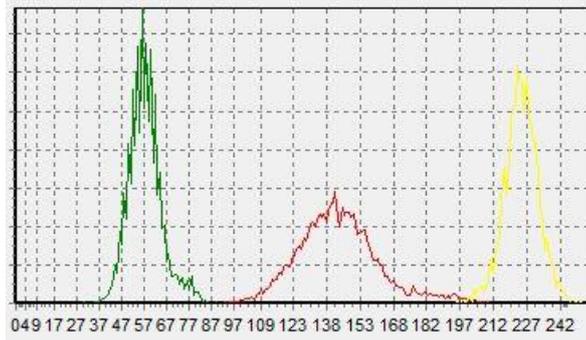


Figura 2.18: Histograma normalizado da Componente vermelha da Figura 2.17. A linha amarela representa a área da textura, a linha vermelha representa área afetada pelo selecionador e a linha verde representa a área afetada após a aplicação do coeficiente a (sem área do texto), o eixo x representa a intensidade da componente.

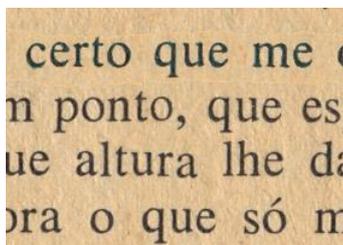
Pode-se notar que além do desvio-padrão o valor esperado também foi alterado, como esperado pela equação 2.5. Resta deslocar o histograma pelo coeficiente b , que pode ser encontrado pela diferença entre o valor esperado dos histogramas representados pela linha amarela e verde da Figura 2.18.

De posse da equação 2.5, e desejando que os valores esperados sejam semelhantes, deve-se deslocar a intensidade dos pixels de b unidades por:

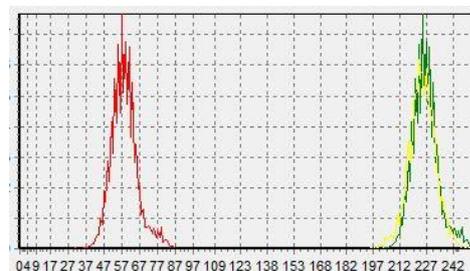
$$b = E_1(X) - E_2(X) \quad (2.7)$$

Sendo $E_1(X)$ o valor esperado da área da textura original da intensidade da componente em questão e $E_2(X)$ o valor esperado do histograma intermediário, conseguido após a aplicação do coeficiente a anteriormente encontrado.

A Figura 2.19 mostra o resultado final do histograma da componente vermelha da Figura 17a (Figura 2.19b na linha verde) e o aspecto final (Figura 2.19a) após a aplicação da mesma técnica nas componentes azul e verde, evidenciando a remoção do marcador em questão.



(a)



(b)

Figura 2.19: (a) Imagem com selecionador ciano removido. (b) Histograma normalizado da componente vermelha da Figura 2.17a. A linha amarela representa a área da textura, a linha vermelha representa área afetada pelo selecionador após aplicação do coeficiente a . A linha verde representa área afetada pelo selecionador após aplicação do coeficiente b (sem área do texto), o eixo x representa a intensidade da componente.

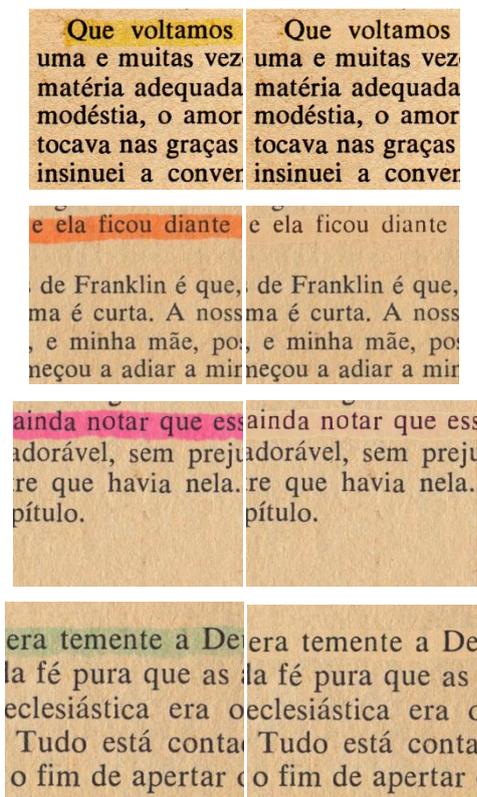


Figura 2.20: Várias imagens afetadas por selecionadores, e o resultado do processamento da técnica descrita neste trabalho nas respectivas áreas afetadas das imagens.

Pode-se utilizar a mesma técnica com marcadores de outras cores. A Figura 2.20 mostra imagens com selecionadores de diversas cores e o resultado após o processamento das imagens com a técnica discutida nesta seção.

Verifica-se que esta técnica é superior à apresentada na seção 2.1.2, eliminado o efeito do marcador e sem a necessidade de iterativamente analisar o resultado do algoritmo.

2.2. Remoção de Sublinhados

O ato de sublinhar textos também é uma atitude frequente de leitores. Pode-se encontrar vários livros digitalizados na Internet que possuem o sublinhado como forma de destaque de passagens que são importantes aos leitores deles. A Figura 2.21 mostra um exemplo de texto destacado por sublinhados. Nesta seção serão apresentados métodos para a remoção destes ruídos.

O mesmo leitor pode estar interessado em diferentes aspectos de um documento cada vez que ele o lê. O que é importante para uma pessoa pode ser considerado irrelevante para outra. Desta forma, em geral, o ato de sublinhar pode ser percebido como um ruído físico que danifica o documento [2].

In other contexts of signal and system analysis, rather than analyzing existing systems, our interest may be focused on the problem of designing systems to process signals in particular ways. Economic forecasting represents one very common example of such a situation. We may, for example, have the history of an economic time series, such as a set of stock market averages, and it would be clearly advantageous to be able to predict the future behavior based on the past history of the signal. Many systems, typically in the form of computer programs, have been developed and refined to carry out detailed analysis of stock market averages and to carry out other kinds of economic forecasting. Although most such signals are not totally predictable, it is an interesting and important fact that from the past history of many of these signals, their future behavior is somewhat predictable; in other words, they can at least be approximately extrapolated.

Figura 2.21: Amostra de texto com sublinhado.

A remoção de linhas retas que percorrem o texto, como as que se encontram em formulários e cadernos pautados foram abordados por vários autores na literatura técnica [18] [19] [20] [21]. O sublinhado traçado pelos leitores é mais complexo. Raramente usa-se régua para sublinhar textos e muitas vezes o segmento de linha toca ou cruza as áreas textuais. A Figura 2.22 apresenta outros exemplos de sublinhado feito à mão. Sublinhados como os da Figura 2.22b, que tocam ou interceptam o texto, também prejudicam processos de reconhecimento ótico de caracteres (*Optical Character Recognition* – OCR).

Para a remoção de sublinhado feito à mão Pinto [22] propõe integrar agrupamento por lógica difusa nas propriedades das cores das imagens originais e morfologia matemática. Segundo os autores, essa técnica foi aplicada com sucesso em livros do século XIX. Esta técnica possui um ponto a ser melhorado que são os casos em que a linha cruza a palavra como no exemplo da Figura 2.22b.

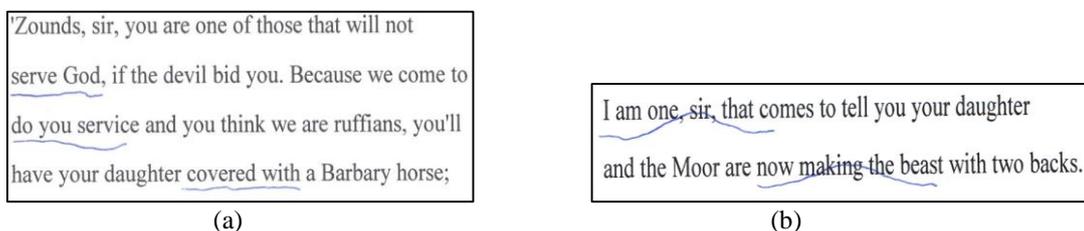


Figura 2.22: (a) Exemplo de sublinhado que não intercepta o texto.
(b) Exemplo de sublinhado que intercepta o texto.

Serão apresentadas duas técnicas para a remoção de sublinhados. Uma com vistas a imagens como a da Figura 2.22, digitalizadas em cores, e uma segunda apropriada para documentos binarizados como apresentado na Figura 2.21. Ambas tratarão a problemática da interceptação do texto pelo sublinhado.

2.2.1. Remoção de sublinhados em documentos coloridos

Para a remoção de sublinhados em documentos que foram digitalizados em cores será utilizada a informação presente na cor da caneta utilizada para o sublinhado.

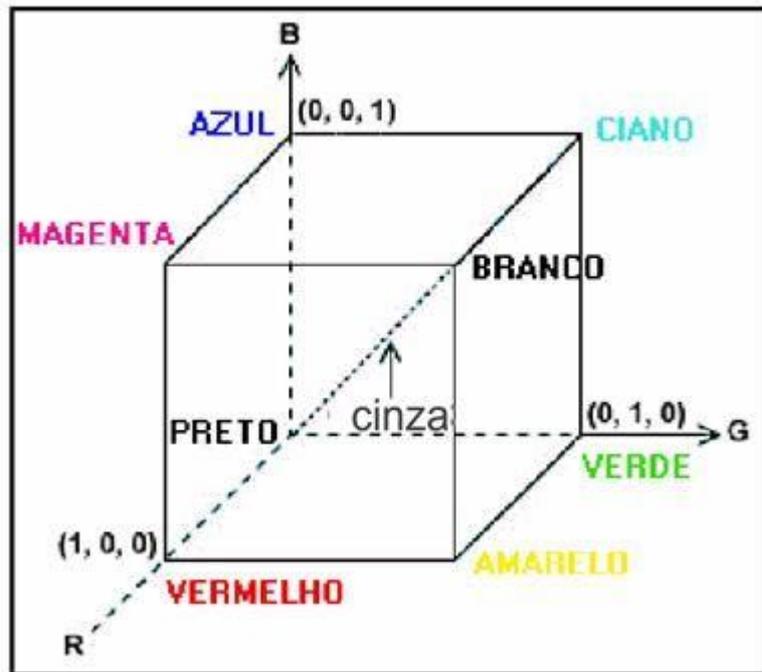


Figura 2.23: Cubo representando o espaço de cores RGB.
A linha que liga os pontos (0,0,0) e (1, 1, 1) é composta por tons de cinza [23].

Será aproveitado aqui o método de segmentação de documentos apresentado para a remoção de marcadores na seção 2.1.2. Esse método envolve uma verificação do limite da distância entre um pixel colorido e a linha imaginária entre os pontos RGB (0, 0, 0) preto e (1, 1, 1) branco. Tal linha forma os tons de cinza (Figura 2.23). Como no caso dos marcadores o objetivo é identificar os pixels que integram o conjunto de ruído que se deseja eliminar. Partindo do Algoritmo 2.1 foi gerado o Algoritmo 2.5 que identifica os pixels formados por canetas coloridas.

```
Para todos os pixels  $P(i) = (Red_i, Green_i, Blue_i)$  na imagem faça  
se  $((Blue_i - limite) > ((Red_i + Green_i) / 2))$  ou  
 $((Red_i - limite) > ((Green_i + Blue_i) / 2))$  ou  
 $((Green_i - limite) > ((Red_i + Blue_i) / 2))$  então  
Marque o pixel como ruído;
```

Algoritmo 2.5: Pseudocódigo para segmentação inicial da imagem.

Segmentando as imagens da Figura 2.22 com o Algoritmo 2.5 resultou nas imagens da Figura 2.24. Alguns pixels que não eram do conjunto formado pelo sublinhado também foram marcados.

No caso em questão, a causa foi o ruído introduzido pelo scanner nas bordas das letras devido à descontinuidade, este tipo de ruído é citado em [2]. Após a binarização e aplicação de um filtro de remoção de sal e pimenta o resultado é apresentado na Figura 2.25.

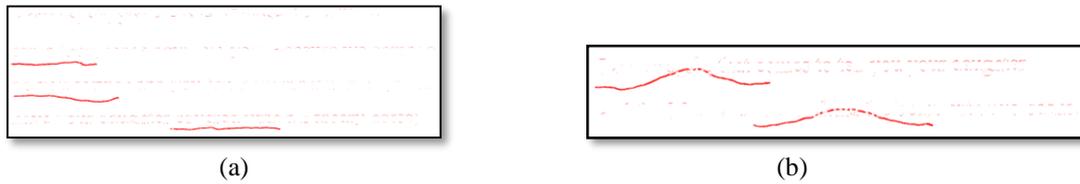


Figura 2.24: Resultado do processamento das imagens da Figura 2.22 com o Algoritmo 2.5.

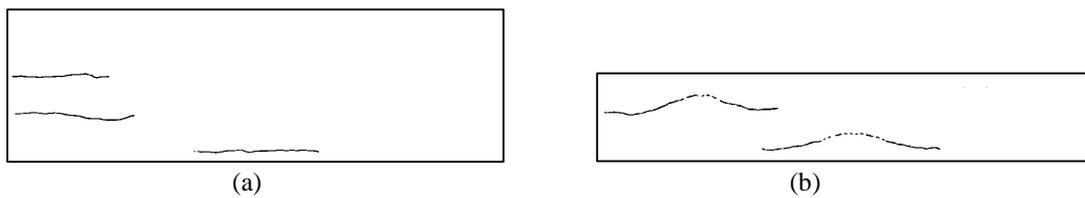


Figura 2.25: Máscaras de segmentação obtidas após aplicação de filtro de remoção de sal e pimenta na Figura 2.24.

As imagens da Figura 2.25 são utilizadas como guia para a substituição dos pixels pelo fundo original da imagem, no caso pela cor branca, na Figura 2.22.

A Figura 2.26 apresenta o resultado final onde o sublinhado foi completamente removido em relação à Figura 2.22.

'Zounds, sir, you are one of those that will not
serve God, if the devil bid you. Because we come to
do you service and you think we are ruffians, you'll
have your daughter covered with a Barbary horse;

(a)

I am one, sir, that comes to tell you your daughter
and the Moor are now making the beast with two backs.

(b)

Figura 2.26: Imagens resultantes com o sublinhado removido.

A Figura 2.27 mostra outros exemplos de remoção de sublinhados com diversos tipos de canetas, utilizando o método desta seção.

RODERIGO

Sir, I will answer any thing. But, I b
If't be your pleasure and most wise c
As partly I find it is, that your fair d
At this odd-even and dull watch o' t
Transported, with no worse nor bett
But with a knave of common hire, a
To the gross clasps of a lascivious M
If this be known to you and your all
We then have done you bold and sa
But if you know not this, my manne
We have your wrong rebuke. Do nc
That, from the sense of all civility,

(a)

RODERIGO

Sir, I will answer any thing. But, I b
If't be your pleasure and most wise c
As partly I find it is, that your fair d
At this odd-even and dull watch o' t
Transported, with no worse nor bett
But with a knave of common hire, a
To the gross clasps of a lascivious M
If this be known to you and your all
We then have done you bold and sa
But if you know not this, my manne
We have your wrong rebuke. Do no
That, from the sense of all civility,

(b)

Figura 2.27: Remoção de sublinhados traçados por diversos tipos de canetas. (a) Imagem original. (b) Resultado após aplicação do método apresentado.

2.2.2. Remoção de sublinhados em documentos envelhecidos

Esta seção aborda o problema da remoção de sublinhado em documentos com fundo envelhecido, tal como o mostrado na Figura 2.28a.

Documentos envelhecidos tendem a ter fundo não branco e apresentam uma textura cujo matiz depende de uma série de fatores, desde a qualidade inicial do papel à sua porosidade, à umidade e temperatura que o documento foi mantido, manuseamento, entre outros fatores.

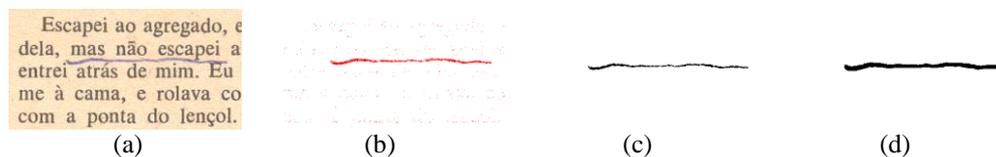


Figura 2.28: Documento envelhecido com sublinhado. (a) imagem original. (b) segmentação do ruído por sublinhado. (c) remoção de ruído sal e pimenta. (d) máscara final obtida pela dilatação de (c).

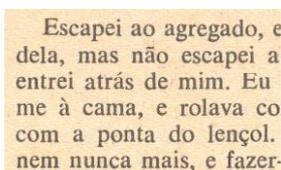
Após a análise de mais de 1.500 documentos entre 1 e 52 anos provindos de bibliotecas e arquivos da universidade foi encontrado uma tendência no decaimento da cor branca presente no documento original [24]. A componente azul tende a esvanecer-se em primeiro lugar, seguida da componente verde, e da componente vermelha. Ruídos como sublinhando feitos por canetas alteram este cenário. No caso dos pixels formados pelo sublinhado na imagem mostrada na Figura 2.28a, a componente

azul possui tons mais intensos do que a componente vermelha e a componente vermelha tons com valores maiores que a verde.

Identificando os pixels que apresentam a lógica (azul > vermelho e vermelho > verde), temos a Figura 2.28b. Após binarização e filtragem sal e pimenta obtém-se a máscara binária mostrada na Figura 2.28c. Para melhorar a qualidade da filtragem de ruído no texto sublinhado, em documentos envelhecido a máscara obtida deve ser dilatada, originando a máscara apresentada na Figura 2.28d.

Papel envelhecido não possui uma cor uniforme. O envelhecimento cria uma textura sobre o fundo do papel. Em [25] a análise de documentos com um século de idade mostrou que a textura em papel segue uma distribuição Rayleigh. Em documentos com meio século de idade a textura de fundo de papel é melhor representada por uma distribuição normal [24].

Assim, uma opção para recompor a imagem de fundo, mantendo a sua textura original, seria a substituição dos pixels na imagem original marcados como afetados pelo sublinhado por pixels escolhidos aleatoriamente do fundo não afetado pelo sublinhado, desta forma a distribuição final da área afetada pelo sublinhado manteria a distribuição da textura original. O resultado da remoção do sublinhado da Figura 2.28a com este processo é mostrado na Figura 2.29.



Escapei ao agregado, e
dela, mas não escapei a
entrei atrás de mim. Eu
me à cama, e rolava co
com a ponta do lençol.
nem nunca mais, e fazer.

Figura 2.29: Documento envelhecido da Figura 2.25a com sublinhado removido.

2.2.3. Remoção de sublinhados em documentos binarizados

Os métodos das seções 2.2.2 e 2.2.1 possuem bons resultados quando utilizados em documentos digitalizados em cores, mas não são adequados para documentos binarizados. Nesta seção será apresentado um algoritmo para tais documentos, como o da Figura 2.30.

Na etapa de segmentação, para cada pixel da imagem identificado como sendo do texto aplicamos um algoritmo *flood fill* [4], levando em consideração pixels conexos com vizinhança-8 [51]. O resultado desta etapa na imagem 2.30 pode ser visualizado na Figura 2.31.

Separando os objetos que possuem uma altura menor que 15% em relação à sua base e que ocupam 20% da área do retângulo imaginário de seu limite, identificamos o texto afetado pelas linhas.

Na próxima etapa utilizamos o algoritmo de esqueletização de Zhang-Suen [52] nos objetos obtidos na etapa anterior. A esqueletização é utilizada aqui como forma de simplificação do problema de retirada do sublinhado.

Existem alguns algoritmos de esqueletização na literatura como os algoritmos de Zhang-Suen, Hilditch e Nagendraprasad-Wang-Gupta ou NWG. Estes algoritmos foram estudados por Widiarti em [53] e o que apresentou os melhores resultados foi o Zhang-Suen, com o melhor tempo de execução entre os algoritmos analisados (Figura 2.32). O mesmo é o que foi utilizado neste trabalho.

O procedimento do algoritmo Hilditch é fazer várias iterações de erosão em um objeto, onde a cada iteração os pixels que satisfazem os seguintes testes são eliminados:

1. O número de pixels entre os pixels vizinhos é de 2 (dois) a seis (6) e há apenas um padrão de mudança de fundo a objeto;
2. Há um pixel de fundo no pixel vizinho da borda superior ou no lado esquerdo ou do lado direito e há um pixel de fundo para o vizinho de cima, ou do lado esquerdo ou abaixo dele.
3. O Algoritmo Zhang-Suen, criado por Zhang e Suen [52], é um algoritmo paralelo de afinamento, em que um novo valor para um dado pixel na iteração corrente depende do valor gerado na iteração anterior, ele utiliza uma janela de 3x3 pixels e cada pixel é conectado quando possui algum vizinho em uma vizinhança-8.

O Algoritmo NWG é um algoritmo de afinamento que funciona removendo sucessivamente um subconjunto da fronteira dos objetos.

IAGO
Farewell; ~~for I must~~ leave you:
It seems not meet, nor wholesome to my place,
To be produced--as, if I stay, I shall--
Against the Moor: for, ~~I do know~~, the state,
However this may gall him with some cheque,
~~Cannot with safety~~ cast him, for he's embark'd
With such loud reason to the Cyprus wars,
Which even now stand in act, that, for their souls,
Another of his fathom they have none,
To lead ~~their business~~: in which regard,
Though I do hate ~~him as~~ I do hell-pains.
Yet, for necessity of present life,

Figura 2.30: Imagem binarizada com sublinhados tocando e interceptando o texto.

IAGO
 Farewell, for I must leave you:
 It seems not meet, nor wholesome to my place,
 To be produced--as, if I stay, I shall--
 Against the Moor: for, I do know, the state,
 However this may gall him with some cheque,
Cannot with safety cast him, for he's embark'd
 With such loud reason to the Cyprus wars,
 Which even now stand in act, that, for their souls,
 Another of his fathom they have none,
 To lead their business: in which regard,
 Though I do hate him as I do hell-pains.
 Yet, for necessity of present life,

Figura 2.31: Identificação dos componentes do texto da Figura 2.30 pelo algoritmo *flood-fill*, os retângulos são somente para ilustrar os objetos encontrados.

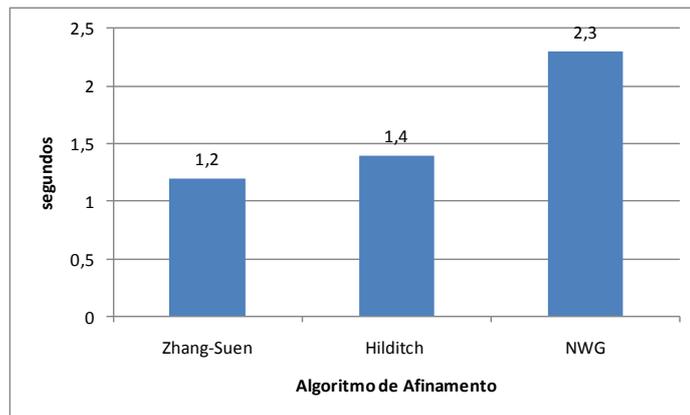


Figura 2.32: Tempo médio de processamento dos algoritmos de afinamento de imagem de uma folha de A4 digitalizada em 300 dpi.

Como nos objetos segmentados com os sublinhados os pontos extremos são coincidentemente pontos extremos da linha ao qual desejamos remover, encontrando o menor caminho entre estes dois pontos e removendo os pixels que fazem parte do mesmo obtemos a imagem da Figura 2.33c. Neste ponto do algoritmo pode-se verificar a simplificação realizada pela esqueletização, a remoção é realizada sobre uma linha de um pixel de largura.

O passo seguinte é o de dilatação em conjunto com uma operação AND com a imagem original. Desta forma mantemos os limites do texto original e eliminamos praticamente toda a linha traçada à mão.

Substituindo os objetos na imagem original completa-se o processo de eliminação do ruído (Figura 2.34).

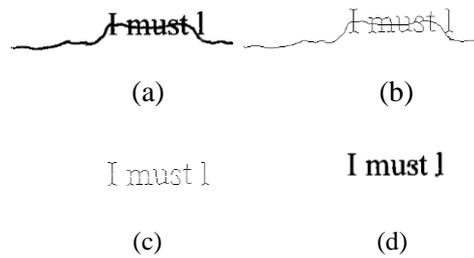


Figura 2.33: (a) Objeto segmentado da imagem original da Figura 2.27. (b) esqueletização do mesmo. (c) remoção da linha. (d) dilatação de (c) com o operador AND e a imagem (a).

IAGO
 Farewell; for I must leave you:
 It seems not meet, nor wholesome to my place,
 To be produced--as, if I stay, I shall--
 Against the Moor: for, I do know, the state,
 However this may gall him with some cheque,
 Cannot with safety cast him, for he's embark'd
 With such loud reason to the Cyprus wars,
 Which even now stand in act, that, for their souls,
 Another of his fathom they have none,
 To lead their business: in which regard,
 Though I do hate him as I do hell-pains.
 Yet, for necessity of present life,
 I must show out a flag and sign of love,

Figura 2.34: Imagem com ruídos de linhas e sublinhados removidos.

Os métodos de dilatação e de operação AND em imagens binárias foram retirados de [54], que apresenta um bom texto sobre morfologia matemática.

O algoritmo completo pode ser visualizado no fluxograma da Figura 2.35.

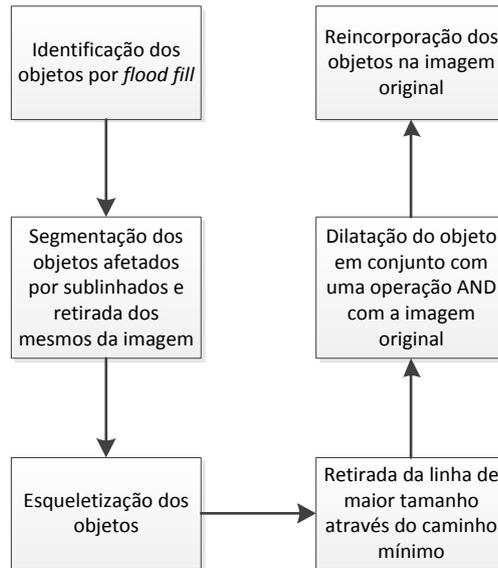


Figura 2.35: Fluxograma do algoritmo proposto nesta seção para retirada de sublinhados em imagens binarizadas.

2.3. Conclusão

Neste capítulo foram apresentados algoritmos para a remoção de marcadores de texto e sublinhados para situações em aberto na literatura.

Em ambos os casos houve análise da remoção destes ruídos em documentos envelhecidos. Esta análise foi útil para o capítulo de análise forense onde as observações no papel envelhecido foram descritas em detalhe.

3. GERAÇÃO DE RESUMOS APARTIR DE MARCADORES

*A arte é um resumo da natureza feito
pela imaginação.*

Eça de Queiroz

No capítulo anterior a tinta deixada por marcadores foi tratada como ruído e foram propostos algoritmos para sua remoção. No entanto, se por um lado a seleção de texto possa ser trada como ruído, por outro a mesma oferece condições de identificação dos trechos importantes do texto podendo servir como guia para a geração de resumos.

Este capítulo complementa o capítulo anterior expandindo a manipulação de imagens de textos afetados por marcadores possibilitando a geração de resumos das partes realçadas. O novo algoritmo foi especificado e testado com as diferentes cores de marcadores disponíveis no mercado (amarelo, azul, verde, laranja e ciano). Além disso, fez-se um estudo sobre a otimização dos parâmetros do algoritmo para permitir a maximização da taxa de transcrição correta por reconhecimento ótico de caracteres (*Optical Character Recognition – OCR*).

3.1. Gerando resumo com textos realçados

O realce de texto por marcadores afeta pelo menos uma componente RGB diminuindo a sua intensidade. O Algoritmo 2.1 testa se as componentes de um documento originalmente monocromático, afetado por marcadores e digitalizado como um documento em cores são mais afastadas umas das outras do que um valor controlado, que é chamado de *distância*.

Aos pixels cujo valor das componentes RGB ultrapassa o limiar da *distância* são atribuídos o valor do componente de maior intensidade.

Para a geração dos resumos de áreas com marcadores, o Algoritmo 2.1 deve ser modificado, o resultado está no Algoritmo 3.1.

```
aux <- 1;
flag <- FALSO;
Para todas as imagens a serem processadas faça
  Para j de 0 até a altura da imagem atual - 1 faça
    Para i de 0 até a largura da imagem_atual - 1 faça
      R <- imagem_atual.pixel[i,j].vermelho;
      G <- imagem_atual.pixel[i,j].verde;
      B <- imagem_atual.pixel[i,j].azul;
      Se (R + G + B) div 3 > limiar então
        RG <- módulo(R - G);
        RB <- módulo(R - B);
        GB <- módulo(G - B);
        Se ((RG > distância) ou (RB > distância) ou (GB > distância)) então
          Se aux <> j então
            aux <- j
            y <- y + 1
            crie mais uma linha na imagem_resumo;
          Senão
            Flag <- VERDADEIRO;
          FimSe;
        Senão
          Se Flag = VERDADEIRO então
            Flag = FALSO;
          FimSe;
        Fimse;
      Senão
        Se Flag = VERDADEIRO então
          Imagem_resumo.pixel[i,y] <- preto;
        FimSe;
      FimSe;
    FimPara;
  FimPara;
FimPara;
```

Algoritmo 3.1: Pseudocódigo para o algoritmo de criação de imagem resumo de partes afetadas por marcadores de um conjunto de imagens.

Leitores geralmente lêem várias páginas de texto, portanto o algoritmo deve estar preparado para processar várias páginas em sequência. Para tanto, a linha atual da imagem de saída será manipulada por uma variável inteira chamada “aux”.

A decisão se um pixel foi afetado por um marcador segue a mesma lógica do Algoritmo 2.1. Se o valor da diferença entre as componentes RGB exceder o limiar *distância*, o pixel é considerado como afetado pelo marcador. Neste caso no Algoritmo 3.1 a variável lógica “*flag*” é ajustada para o valor VERDADEIRO. A variável *flag* mantém a informação se o pixel sendo processado ocorreu após um pixel afetado por selecionador na mesma linha da imagem original.

Pixels com valores abaixo da variável “*limiar*” são considerados como sendo integrantes do texto, enquanto valores acima são considerados como fundo do papel.

Se um pixel do texto vier após um pixel afetado por um marcador então o pixel na mesma coluna e linha (controlado pela variável “*y*”) é ajustado para preto na imagem resumo.

A Figura 3.1a apresenta uma imagem com texto afetado por marcadores de diversas cores, e na Figura 3.1b o resultado do processamento da Figura 3.1a pelo Algoritmo 3.1.

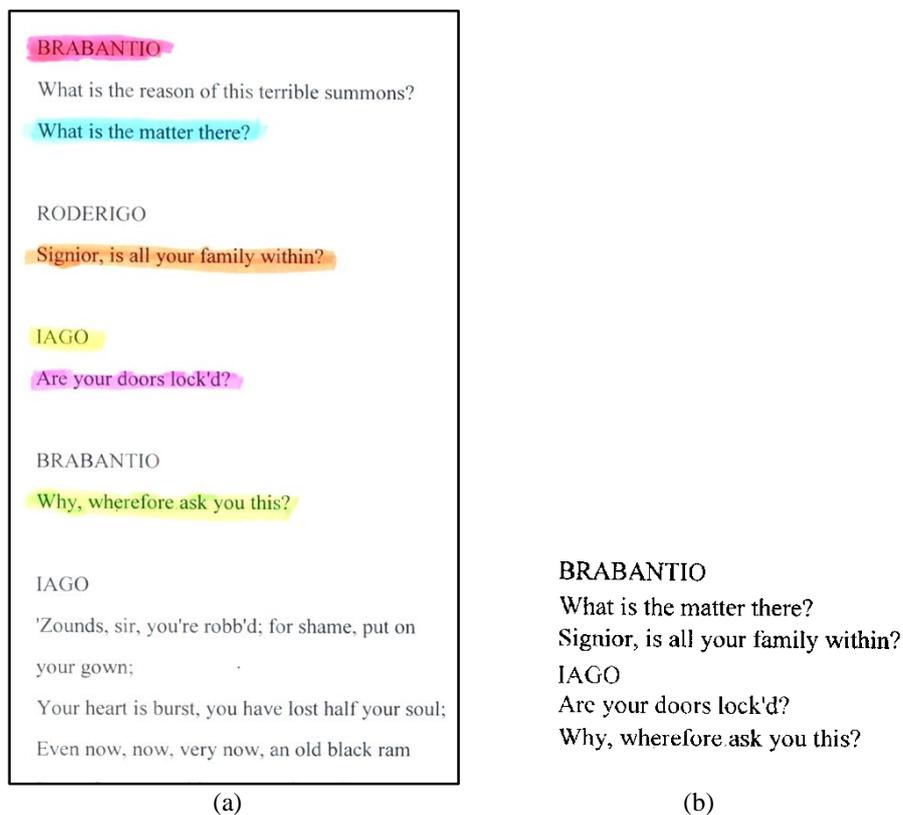


Figura 3.1: (a) Imagem Original. (b) Resultado do processamento da imagem (a) pelo Algoritmo 3.1. Parâmetros: *limiar* = 170; *distância* = 35.

Analisando os resultados do Algoritmo 3.1 foi verificado que valores do parâmetro *distância* próximos a 35 retornavam os melhores resultados da imagem resumo.

Se a cor do marcador for conhecida *a priori*, também pode-se otimizar o código do Algoritmo 3.1 conforme demonstrado no Algoritmo 2.2.

3.2. A relação entre o parâmetro limiar e sua influência no desempenho no pós-processamento por OCRs

Se a imagem resumo for processada posteriormente por um OCR alguns itens devem ser analisados para se poder alcançar uma boa taxa de acertos de caracteres [86] [87], entre estes itens estão: qualidade do documento original, inclinação do texto e qualidade da imagem binarizada. Como o limiar de binarização é controlado pela variável limiar, é analisado seu efeito na qualidade da imagem resumo. A Figura mostra três resultados obtidos pela variação deste parâmetro.

That thou, Iago, who hast had my purse
If ever I did dream of such a matter, Abhor me.
In personal suit to make me his lieutenant.

(a)

That thou, Iago, who hast had my purse
If ever I did dream of such a matter, Abhor me.
In personal suit to make me his lieutenant,

(b)

That thou, Iago, who hast had my purse
If ever I did dream of such a matter, Abhor me.
In personal suit to make me his lieutenant.

(c)

Figura 3.2: Influência do parâmetro *limiar* na imagem resumo.
(a) *limiar* = 70. (b) *limiar* = 170. (c) *limiar* = 210.

Pode-se observar que valores baixos para o parâmetro *limiar* resultam em texto que desvanece na imagem resumo. Valores altos para o mesmo parâmetro resulta na transcrição de ruído da área afetada pelo marcador para a imagem resumo, degradando em muito o desempenho do OCR.

Para uma melhor análise do valor ótimo para o parâmetro *limiar* processamos 10 páginas de textos contendo 1.250 caracteres afetados por marcadores e variamos este parâmetro de 0 a 255. A Figura 3.3 apresenta a taxa de acerto do OCR na imagem resumo durante a variação do parâmetro *limiar*. O OCR utilizado para este teste foi o Cuneiform PRO OCR 6.0 [88].

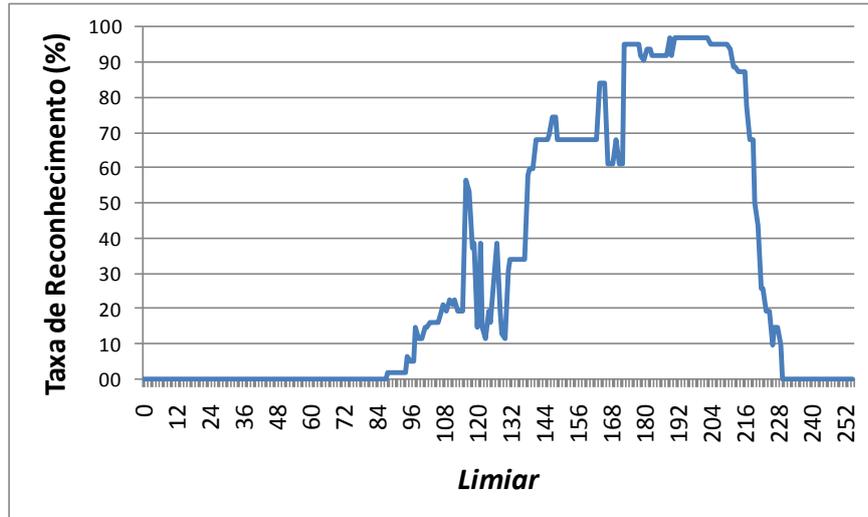


Figura 3.3: Gráfico da taxa de acerto de reconhecimento de caracteres com a variação do parâmetro *limiar*.

As melhores taxas de acerto de reconhecimento foram obtidas ajustando o parâmetro *limiar* entre 190 e 200.

3.3. Pré-processamento para ajuste de marcadores aplicados irregularmente

Leitores de textos escritos com alfabeto latino tendem a realizar a marcação de texto da mesma forma que escrevem, isto é, horizontalmente, da esquerda para a direita.

Nem sempre os leitores são cuidadosos o suficiente para marcar uma área de forma uniforme. Podem-se encontrar casos em que a tinta do marcador está no fim ou o leitor não segurou a caneta marcadora em completo contato com a superfície do papel.

Estes fatores podem resultar em uma área não uniforme afetada por marcador, que não pode ser perceptível a olho nu. A Figura 3.4a mostra um zoom em uma área afetada por marcador de forma não uniforme, nela uma área adjacente à letra “S” não está pintada de amarelo.



Figura 3.4: (a) Zoom em uma área afetada por marcador amarelo de forma não uniforme. (b) Figura 3.4a processada através do Algoritmo 3.1.

Imagens com áreas com marcadores aplicados de forma não uniforme apresentam problemas como o mostrado na Figura 3.4 quando processados pelo Algoritmo 3.1. Nela a superfície irregular se tornou facilmente perceptível na imagem resumo, onde o “S” foi dividido em dois objetos desconexos.

A Figura 3.5 mostra um exemplo do resultado de marcador aplicado de forma não uniforme em uma linha de texto da imagem resumo, na qual se pode observar que o ruído se propagou por toda a linha Figura 3.5b.

IAGO

Call up her father,

Rouse him: make after him, poison his delight,

Proclaim him in the streets; incense her kinsmen,

And, though he in a fertile climate dwell,

Plague him with flies: though that his joy be joy,

Yet throw such changes of vexation on't,

As it may lose some colour.

RODERIGO

Here is her father's house; I'll call aloud.

IAGO

Do, with like timorous accent and dire yell

As when, by night and negligence, the fire

Is spied in populous cities.

RODERIGO

What, ho, Brabantio! Signior Brabantio, ho!

(a)

Call up her father,

As it may lose some colour.

Is spied in populous cities.

(b)

Figura 3.5: Imagem resumo de texto afetado por marcador aplicado de maneira não uniforme e processado pelo Algoritmo 3.1.

Analisando a dinâmica do Algoritmo 3.1 se pode verificar que tais ruídos aparecem na imagem resumo quando existem lacunas nas áreas afetadas pelos marcadores. Quando estas lacunas ocorrem

logo antes de texto o que vem a seguir não é copiado para a imagem resumo. Uma solução é corrigir estas lacunas de marcadores no documento original, se ele ainda estiver disponível. Outra solução seria um pré-processamento da imagem onde estas lacunas seriam preenchidas antes da aplicação do Algoritmo 3.1. Para este pré-processamento todos os pixels conexos com vizinhança-8 aos pixels afetados por marcadores são também identificados como sendo afetados pelo marcador. Esta identificação pode ser feita pelo deslocamento do valor de uma das componentes RGB para valores mais baixos, desta forma há a expansão das áreas afetadas por selecionadores e preenchimento das lacunas.

Pré-processando a imagem 3.4a e em seguida aplicando o Algoritmo 3.1 se obtém a imagem mostrada na Figura 3.6, na qual a letra “S” foi corretamente copiada para a imagem resumo. Nota-se que o texto resultante possui um efeito de negrito em relação à imagem original, o mesmo foi devido à forma de identificação das áreas afetadas por selecionador e marcadas para expansão através da alteração da distância entre os componentes RGB.

A Figura 3.5 também apresenta o mesmo efeito de negrito com a aplicação da lógica de pré-processamento, como se pode observar na Figura 3.7.

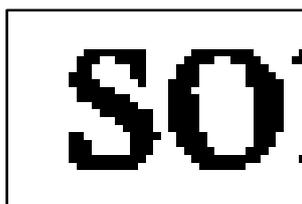


Figura 3.6: Figura 3.4a após pré-processamento e aplicação do Algoritmo 3.1.

Call up her father,
As it may lose some colour.
Is spied in populous cities.

Figura 3.7: Imagem resumo com aplicação do pré-processamento.

A etapa de pré-processamento também afeta a taxa de acerto de reconhecimento de caracteres por OCR, como mostrado na Figura 3.8. As melhores taxas de reconhecimento foram encontradas nos valores de *limiar* entre 127 e 156, para o mesmo conjunto de imagens do gráfico da Figura 3.3. Foi observado que a taxa de reconhecimento chegou à 100% com o pré-processamento e este conjunto de dados.

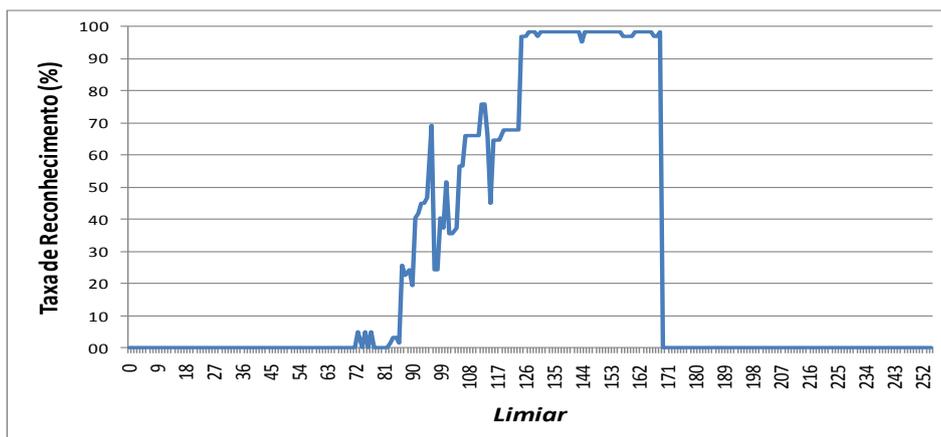


Figura 3.8: Gráfico da taxa de acerto de reconhecimento caracteres com a variação do parâmetro *limiar*, após a aplicação da etapa de pré-processamento.

3.4. Conclusão

Este capítulo apresentou um algoritmo para a geração de imagens resumos de texto envolto por marcadores de texto. As imagens de entrada do algoritmo devem ser originalmente monocromáticas e marcadas por canetas marcadoras de texto coloridas. Utilizamos para o teste imagens digitalizadas à 300 dpi em cores.

A imagem resumo pode ser posteriormente processada por um OCR com bons resultados, como visto nas Figuras 3.7 e 3.8. Se o marcador não for aplicado no documento de forma uniforme, um pré-processamento pode se tornar necessário para o aumento da taxa de acerto de caracteres.

4. ANÁLISE FORENSE

*“Quando você elimina o impossível,
o que restar, não importa o quão
improvável, deve ser a verdade.”*

Sherlock Holmes

A ciência forense é a aplicação de várias técnicas para investigar situações após um fato, e estabelecer o que ocorreu baseado em evidências coletadas. Ela é importante em questões judiciais cíveis e criminais [92].

A análise científica de documentos tem sido uma parte integrante da ciência forense por quase uma centena de anos, e muitas das técnicas desenvolvidas ao longo desse período ainda estão em uso hoje [38]. No entanto, a tecnologia utilizada para a produção de documentos continua a evoluir, os métodos utilizados para produzir falsificações estão cada vez mais sofisticados, as exigências de advogados e tribunais são ainda mais rigorosas.

De acordo com Audrey Giles [38], o exame forense de documentos abrange três áreas principais:

4. Identificar indivíduos através de sua escrita.
5. Determinar se as assinaturas são verdadeiras ou se são simulações.
6. Determinar a origem e a história de documentos.

Uma pesquisa realizada por Srihari e Leedham [39] aborda os dois primeiros itens acima e afirma que: "o exame forense de documentos está em uma encruzilhada, devido aos desafios colocados à sua base científica, bem como devido à disponibilidade de métodos computacionais revolucionários".

Neste capítulo serão abordadas técnicas de análise forense em documentos digitalizados em baixa resolução (300 dpi). Esta escolha se deve a esta resolução ser atingida pelos scanners atuais e por se pretender que não sejam exigidos equipamentos caros para as soluções aqui propostas.

4.1. Identificação de traços de canetas esferográficas

A análise forense de textos escritos com canetas esferográficas é uma importante tarefa. Os profissionais da área geralmente utilizam métodos ópticos e cromatografia [26] para a verificação de documentos. Os métodos conhecidos apresentam a necessidade de equipamentos como microscópios e substâncias químicas, sendo que alguns deles resultam na destruição de parte do material em teste. O interesse da maioria das análises é determinar se dois pedaços de textos foram originados da mesma tinta, desta forma, a comparação de diferentes textos em um documento é o principal objetivo da maioria das investigações [27].

Dentro do campo da análise forense, busca-se informações que possibilitem obter resultados que auxiliem encontrar, com um certo grau de probabilidade, a identificação das tintas das canetas esferográficas. Segundo Reibland [28], o campo de estudo das canetas esferográficas é muito discutido, e questiona-se o uso de métodos aquosos destrutivos para a identificação. De acordo com Sakayanagi [29], os métodos não destrutivos são úteis somente para responder à questão de saber se dois ou mais documentos podem ter a mesma origem. Quando informação mais detalhada é necessária, alguma forma de exame químico destrutivo precisa ser usada.

Os processos como o de cromatografia utilizam dois líquidos, ou misturas de líquidos: um atuando como fase móvel e outro suportado sobre papel atuando como fase estacionária. Nisso ocorre a retenção de substâncias, devido às diferentes afinidades para com as fases estacionária e móvel, separando as tintas de acordo com as suas características. Estes processos são destrutivos ao substrato de papel, além de serem custosos [29].

As tintas de canetas esferográficas são viscosas e são insolúveis em água. Estas tintas são constituídas por corantes, dissolvidos em um ou vários solventes, e resinas as quais outros componentes podem ser incorporados (como aditivos), a fim de modificar as propriedades das tintas, tais como ajustadores de viscosidade, modificadores de elasticidade, inibidores de corrosão ou de lubrificantes para a esfera da caneta [30].

As tintas mais estudadas, para determinar o momento em que foram colocadas em papel, são as tintas de canetas esferográficas, isto principalmente porque os solventes permanecem na tinta durante

algum tempo [31]. De uma maneira geral e simples, é possível afirmar que as tintas de instrumentos de escrita manuais são compostas por uma mistura de corante, e um transportador ou um veículo com um ou mais solventes que possuem uma mistura que depende do fabricante da caneta [30].

Os corantes nestas tintas são importantes para o cientista forense, porque os corantes permanecem no papel em forma de letras, marcas ou linhas após a evaporação do solvente [29].

A assinatura, produzida com caneta esferográfica, é muitas vezes o único traço manuscrito em um documento que está sendo questionado. Desta forma, ela representa o elemento mais importante para a análise de documentos.

A análise de tinta de caneta esferográfica em documentos é questionada muitas vezes no campo da ciência forense, a fim de identificar o instrumento de escrita utilizado na prática de um crime [29]. Quando um teste escrito suspeito de alteração foi apresentado ao autor deste trabalho, durante a averiguação foi verificado a necessidade de um método de baixo custo e eficiente para a realização de um teste de autenticidade na tinta feita por caneta esferográfica na mesma. Por sorte, havia outros testes escritos da mesma disciplina que foram corrigidas pelo autor com a mesma caneta que ainda estavam em sua posse.

O método proposto neste trabalho necessita somente de um scanner de mesa com resolução a partir de 300 dpi, um equipamento barato para os padrões atuais, e documentos com traços de caneta que se deseja verificar se são da mesma origem.

Os passos para a verificação são:

1. digitalização dos documentos;
2. separação dos objetos no texto;
3. mensuração da distância de mahalanobis entre as populações multivaloradas dos objetos;
4. comparação das distribuições das distâncias entre objetos de texto escrito por caneta esferográfica.

4.1.1. Distância de Mahalanobis

Em 1936 P. C. Mahalanobis definiu uma distância baseada em considerações estatísticas que levam em consideração o desvio padrão das variáveis e a sua correlação. A distância de Mahalanobis corresponde à distância euclidiana quando as variáveis são descorrelacionadas e os desvios padrão são iguais [32].

A escolha da distância de mahalanobis para a mensuração se deve ao fato de que a distância euclidiana tradicional é uma raiz da soma da diferença das médias das componentes RGB e, conseqüentemente, uma pequena perturbação pode resultar em uma grande distância euclidiana, ou seja, a distância euclidiana é muito sensível a mudanças de iluminação [32]. Como esta pesquisa foi

realizada com um scanner comum de mesa, a possibilidade de variação da iluminação interferir com os resultados foi minimizada pela escolha desta medida de distância.

Formalmente, a distância de mahalanobis entre duas populações x e y com médias μ_x e μ_y é definida por [33]:

$$D_{xy} = \sqrt{(\mu_x - \mu_y)^T S^{-1} (\mu_x - \mu_y)} \quad (4.1)$$

Onde S^{-1} é a inversa da matriz de covariância ponderada da amostra.

As populações que foram analisadas são formadas por pixels dos traços feitos por canetas esfereográficas em estudo. Cada pixel é formado por um vetor de três componentes R, G, B, correspondentes às cores primárias Vermelha (R), Verde (G) e Azul (B).

Para o cálculo da distância entre dois elementos de texto, dada a formula (4.1) seguimos os seguintes passos:

- 1 – se calcula a média das componentes R, G, B dos dois conjuntos de pixels.
- 2 – constroem-se duas novas populações formadas pela diferença entre as componentes de suas médias.
- 3 – se calcula a matriz de covariância das duas novas populações.
- 4 – se calcula a matriz de covariância ponderada pela fórmula:

$$S^{-1} = \left(\frac{\text{cov}(x) * (n_x - 1) + \text{cov}(y) * (n_y - 1)}{n_x + n_y - 2} \right)^{-1} \quad (4.2)$$

Onde $\text{cov}(x)$ é matriz de covariância de x e n_x é a quantidade de elementos (pixels) em x .

- 5 – se calcula o vetor da diferença das médias das duas populações
- 6 – por fim por (4.1) retira-se a raiz quadrada da multiplicação do vetor diferença de médias, sua transposta e da matriz de covariância ponderada.

Para o processamento dos arquivos textos contendo as componentes RGB dos traços de caneta, foi utilizada a ferramenta R [84]. Esta ferramenta possui suporte para a construção de gráficos, análise de dados e modelos estatísticos utilizando dados que ocupam a memória principal do computador [85]. A fórmula 4.1 foi traduzida para esta ferramenta e está no Algoritmo 4.1.

```

c1<-read.csv2("C:\\Ricardo\\doutorado\\tese\\canetas\\Bic\\300dpi\\1.txt")
c2<-read.csv2("C:\\Ricardo\\doutorado\\tese\\canetas\\Bic\\300dpi\\2.txt")

c1.R.media <- mean(c1$R)
c1.G.media <- mean(c1$G)
c1.B.media <- mean(c1$B)
c1.centrado <- c1
c1.centrado$R <- c1.centrado$R - c1.R.media
c1.centrado$G <- c1.centrado$G - c1.G.media
c1.centrado$B <- c1.centrado$B - c1.B.media

c2.R.media <- mean(c2$R)
c2.G.media <- mean(c2$G)
c2.B.media <- mean(c2$B)
c2.centrado <- c2
c2.centrado$R <- c2.centrado$R - c2.R.media
c2.centrado$G <- c2.centrado$G - c2.G.media
c2.centrado$B <- c2.centrado$B - c2.B.media

c1.cov<-cov(c1.centrado)

c2.cov<-cov(c2.centrado)

c1.qtd <- length(c1$R)
c2.qtd <- length(c2$R)

c.cov.ponderada <- (c1.cov * (c1.qtd - 1) + c2.cov * (c2.qtd - 1)) / (c1.qtd +
c2.qtd - 2)

c.cov.ponderada.inversa <- solve(c.cov.ponderada)

medias<-matrix(nrow=1, ncol=3)
medias[1,1] <- c1.R.media - c2.R.media
medias[1,2] <- c1.G.media - c2.G.media
medias[1,3] <- c1.B.media - c2.B.media

maha <- sqrt(medias%%c.cov.ponderada.inversa%%t(medias))

```

Algoritmo 4.1: Programa em R Script para o cálculo da distância de mahalanobis conforme fórmula 4.1.

4.1.2. Limiar de decisão dos traços de caneta esferográfica

Para a análise dos traços de canetas é necessário inicialmente a separação dos mesmos do fundo do papel. A escolha do limiar no qual será tomada a decisão (do que é pixel de fundo de papel ou pixel de caneta) influencia na população de pixels a ser levada em consideração e, conseqüentemente, nos valores das distâncias entre as populações. Na Figura 4.1 temos um exemplo de traço de caneta esferográfica e os valores de médias e desvios padrões das componentes RGB, calculados a partir dos pixels que possuíam valores menores (mais escuros) que o limiar no eixo x do mesmo traço.

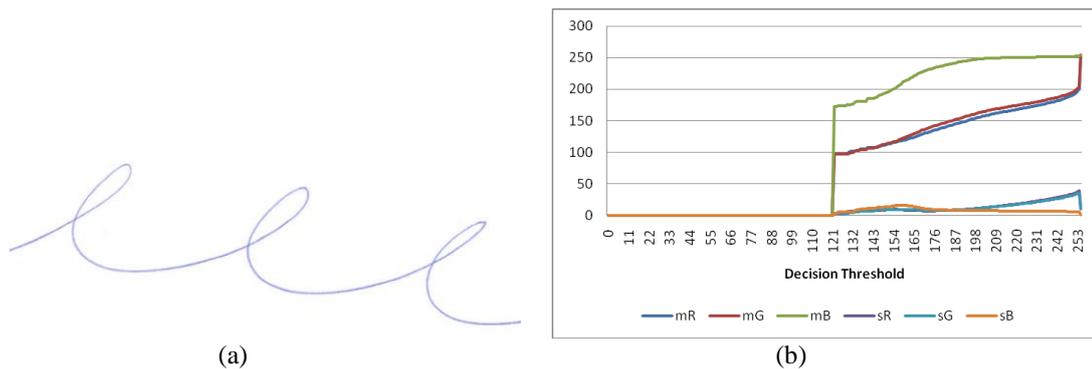


Figura 4.1: (a) Exemplo de traço de caneta azul. (b) Valores de médias e desvios padrões de suas componentes RGB versus o limiar de decisão da separação dos pixels do traço e dos pixels do fundo do papel.

O valor utilizado para o limiar nesta seção foi de 230, o mesmo resultou em uma resposta com menores variações pelo tempo em que o traço foi realizado em documentos de fundo branco com envelhecimento no escuro.

4.1.3. Distribuição das distâncias entre os traços de caneta

O método proposto consiste em dividir as amostras de texto escrito, ou mesmo amostras de traços de canetas, em duas partes. Uma contendo objetos de uma fonte de caneta, e uma segunda parte que deseja-se verificar se os objetos foram escritos com a mesma caneta.

Para cada grupo infere-se as distâncias entre cada par de objeto através da fórmula (4.1), desta forma em um grupo com n objetos encontraremos a combinação de n combinados 2 a 2 ou $C_{n,2}$. Sendo:

$$C_{n,2} = \frac{n!}{2!(n-2)!} \quad (4.3)$$

Realizamos testes com 400 canetas de 4 marcas diferentes encontradas no comércio local e resumimos os resultados na Tabela 4.1.

Tabela 4.1: Resultados das Médias e Desvio-Padrão das Distâncias entre 4 marcas de canetas a partir de documentos digitalizados em 300 dpi.

	Bic	FaberCastell	Pentel	Pilot
	Média; Desvio padrão			
Mesma Caneta	0.338; 0.163	0.382; 0.242	0.267; 0.125	0.370; 0.232
Bic	0.631; 0.348	-	-	-
Faber Castell	1.120; 0.292	0.355; 0.266	-	-
Pentel	1.500; 0.364	1.230; 0.171	0.445; 0.256	-
Pilot	1.370; 0.224	1.230; 0.171	1.560; 0.246	0.549; 0.292

Na linha “Mesma Caneta” da tabela 4.1 estão as médias e desvio padrão quando comparamos texto da mesma caneta das 4 marcas analisadas. Verificamos que mesmo quando são utilizadas outras canetas da mesma marca há um aumento na média encontrada. Esta média aumenta quando se compara textos de outras marcas. Esta observação deve ser levada em consideração na análise dos grupos de textos.

Traçando as curvas da Marca Bic na Figura 4.2 com os dados obtidos da Tabela 4.1, verificamos as distribuições das distâncias.

Uma forma de decisão se os valores correspondem à mesma caneta ou não pode ser analisada a partir do Box Plot da Figura 4.3. Nela podemos verificar que os cones (*notchs*) não se sobrepõem. Segundo [33], se os cones de duas medianas não se sobrepõem, as medianas são significativamente diferentes com 95% de confiança. Comportamento encontrado em todas as comparações 2 a 2 da mesma Figura.

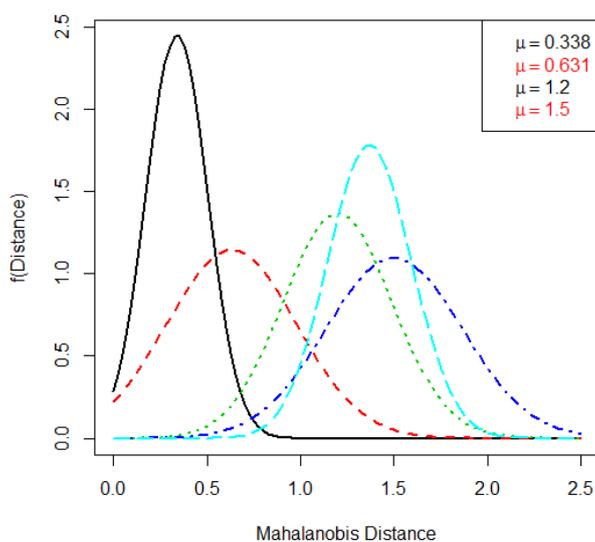


Figura 4.2: Distribuições da marca Bic obtidos da Tabela 4.1, as curva mais a esquerda são comparações entre traços da mesma caneta, a curva vermelha comparações de distâncias entre canetas Bic e as 3 curvas mais a direita são comparações de textos entre Bic e as Marcas FaberCastell, Pentel e Pilot.

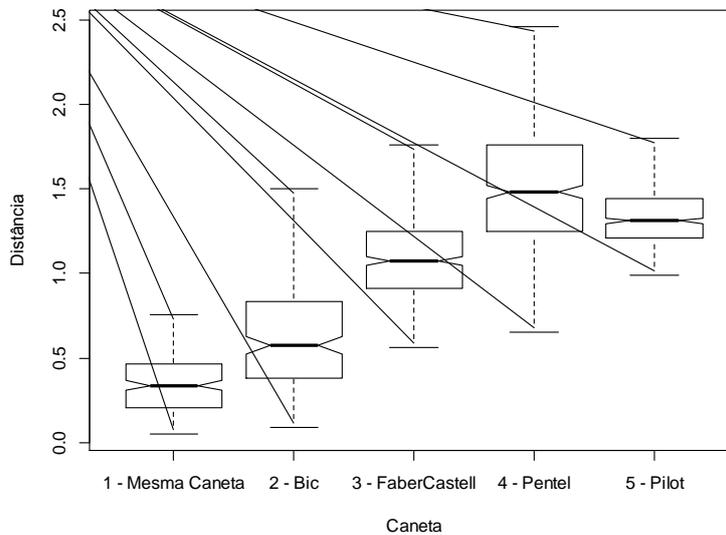


Figura 4.3: Notched Box Plot das distâncias entre os objetos de texto da Marca Bic comparados com as marcas em estudo, as colunas correspondem as distribuições da Figura 4.2.

Os cones do Box plot da Figura 4.3 foram calculados através de [33]:

$$M \pm 1.7(1.25R/1.35\sqrt{N}) \quad (4.4)$$

Onde: M é a mediana, R a amplitude interquartil e N o número de observações.

Imagens digitalizadas em 200 dpi resultaram em valores próximos à da tabela 4.1.

4.1.4. Considerações sobre o envelhecimento da tinta

Verificaram-se mudanças nas medidas quando se levou em conta o fator de envelhecimento do documento armazenado em local escuro, embora sejam necessários alguns meses para o aparecimento de valor perceptível, tal comportamento também foi encontrado por outros processos de verificação ótica como em [34].

O processo dinâmico do envelhecimento de tinta no papel não envolve somente a evaporação dos componentes voláteis, mas possivelmente oxidação, polimerização, interações do papel com a tinta, ou uma combinação destes fatores. Ele monotonicamente desacelera e para após um período [35]. Ele também é influenciado por fatores externos como: luz, umidade, temperatura; ou seja, as condições de armazenagem do documento [36].

Comparando amostras em um microscópio digital (Keyence model VHX-600), foi possível observar o aparecimento de falhas nos traços (Figura 4.4). Mesmo nas imagens digitalizadas em 300

dpi pode-se observar distanciamento dos valores originais. As alterações foram dependentes também das marcas testadas, algumas apresentaram mais alterações que outras.

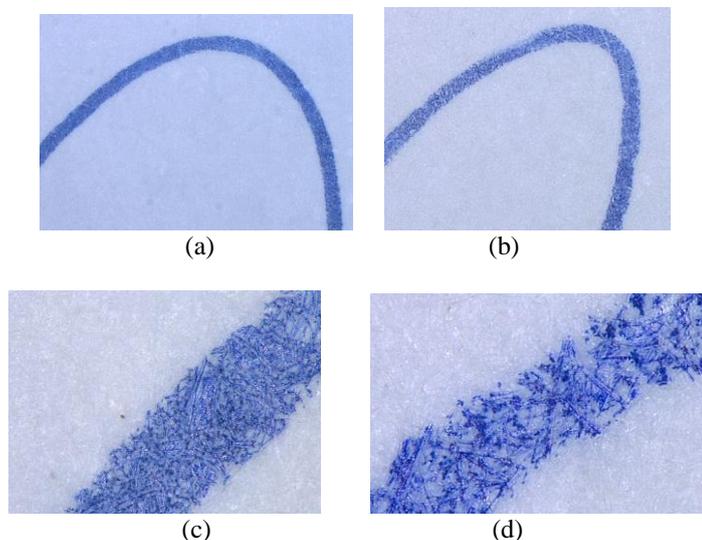


Figura 4.4: Micrografia de exemplos de traços feitos com caneta Bic. (a) e (c) são traços recentes enquanto que (b) e (d) possuem um ano. Os aumentos foram de 30x para (a) e (b) e 200x para (c) e (d).

De forma geral, estas alterações são benéficas para a verificação da autenticidade de documentos, pois podem identificar casos em que um texto foi colocado tempos depois do documento ter sido confeccionado. Mesmo que a mesma caneta tenha sido utilizada para inserção de texto, assume-se que a tinta não envelhece no cartucho de armazenagem, somente após a mesma ser depositada no papel há o esvanecimento dos pigmentos, evaporação dos solventes e polimerização das resinas [37].

4.2. Estudo sobre o efeito do envelhecimento na cor de papel branco

O foco desta seção é determinar a idade de certidões de nascimento e de casamento emitidos no Brasil antes de 1990 e depois de 1960. Este intervalo de tempo é de interesse legal, porque eles são usados por pessoas para provar um número de situações, tais como: relações de trabalho, residência, formação, para reivindicar benefícios legais, as ligações matrimoniais, herança, entre outros [93].

Documentos com mais de meio século são de menor interesse burocrático, uma vez que essas questões tendem a ser resolvidas, quando as pessoas têm menos de 50 a 70 anos de idade. A Figura 4.5 traz um exemplo de um documento de interesse forense: uma certidão de nascimento, na qual se pode observar que o documento foi escrito em forma de papel comum impresso, que originalmente era branco e recebeu um número de selos, carimbos e anotações.

Embora o Brasil, desde 1943, siga o padrão internacional de tamanhos de papel ISO 216 [40], com base na norma alemã DIN 476 [41], o tamanho de certificados oficiais (nascimento, casamento,

morte, etc.) é "ofício" (216 milímetros x 330 mm), que é aproximado, mas diferente do tamanho "legal" (216 mm x 356 milímetros) adotado nos EUA. A densidade do papel varia, mas tende a ser de 90 g/m² ou mais elevado, de tal forma que a interferência frente e verso [2] (ou sangramento) é observada em documentos.

Registada no livro de emolumentos
sob o n.º 115

Ano 1910-1911
N.º 7
Folhas 4
Administração

Registo Civil da República Portuguesa

Certidão de Nascimento

LUIZ LOPES DA SILVEIRA
Ajudante do Conservador do Registo
Civil de Setúbal

Narrativa

Ficha n.º *115 e 116*
Certidão que dos livros de registo de nascimento arquivados nesta Conservatória referentes ao ano de 1910 consta que no dia *quinze e cinco* do mês de *Maio* do ano de *mil novecentos e sete* na freguesia de *Santo André, de Mossamedes* nasceu um indivíduo do sexo *masculino* a quem foi dado o nome de *Antonio Aniceto* filho de *Antonio Ribeiro Monteiro, natural de Lisboa, e de Maria Joana Lino de Figueiredo e Silva Monteiro, natural também de Lisboa*

CONTA

N.º 45-Emol.™	2800
Selo	2800
Soma	7800
N.º 51	1800
Total	10800

E por ser verdade se passou esta certidão que conferi e assino.
Conservatória do Registo Civil de Setúbal
seis de Janeiro mil novecentos e trinta e nove

O Ajudante d O Conservador do Registo Civil,

Mod. 30 de Coelho & Silva, Lda., Porto

Figura 4.5: Exemplo de documento de interesse forense.

Documentos mais recentes já estão sendo gerados em formato digital e geralmente são armazenados em bancos de dados do governo. Assim, eles podem ser acessados on-line ou por órgãos governamentais ou pelo próprio cidadão. Além disso, na sua versão impressa, como se pode ver no documento na Figura 4.6, o documento é impresso em um papel com um fundo texturizado e existe um selo numerado. Essa numeração é única e o selo tem um holograma e é produzido pela Casa da Moeda do Brasil, fazendo que falsificações sejam mais difíceis de serem forjadas do que em documentos mais antigos.

Mesmo após pesquisa intensiva pelo autor desta tese e por especialistas de agências governamentais (consultados neste trabalho para tentar encontrar métodos de datação de documentos no espaço de tempo entre 1950 e 1990), não foi encontrado muito sobre determinação de idade documento. A literatura concentra-se em outros aspectos do envelhecimento do papel, tais como as alterações de:

- parâmetros mecânicos: resistência a dobras [42] e propriedades de tração [43] [44];
- parâmetros químicos: determinação do grau de polimerização [45] e determinação de PH [46];
- identificação do tipo de fibra utilizado no processo de produção de papel por meio de microscopia [47].

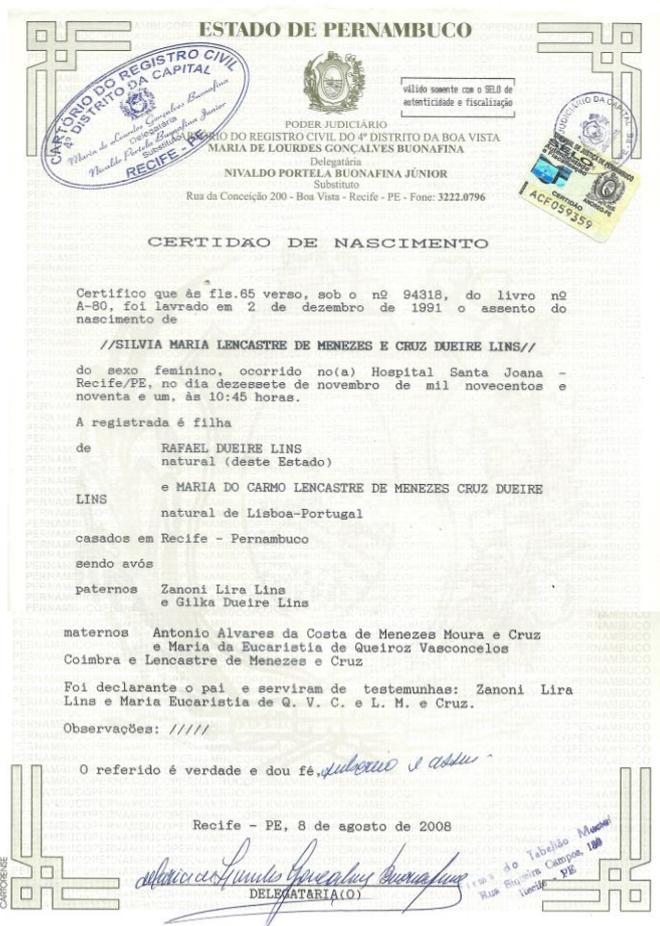


Figura 4.6: Exemplo de uma certidão de nascimento a partir de 1990, impresso em papel com "marca d'água" e selo numerado com holograma emitido pela Casa da Moeda do Brasil.

Existem mais trabalhos voltados para a análise do processo de envelhecimento da tinta em si, mas a maioria dos documentos burocráticos de interesse foram digitados com um dispositivo mecânico em um formulário em branco, sem quaisquer partes escritas à mão nem sequer uma assinatura, como alguns daqueles documentos que fazem uso de um selo legal. Assim, foi encontrado que a determinação do padrão de envelhecimento tinta possui pouco uso.

Um trabalho interessante é apresentado por Richard L. Brunelle¹ e Erich J. Speckin [48] onde é utilizado envelhecimento acelerado de tinta para estimar a idade de tintas esferográficas em documentos questionados. Neste trabalho é afirmado que 50 dos resultados descritos foram admitidos como prova em processos judiciais dos Estados Unidos, nos últimos dez anos. O método é altamente sofisticado e baseia-se na forma como a tinta de caneta esferográfica envelhece.

4.2.1. Características dos documentos analisados

Foram analisadas para este trabalho 30 amostras de documentos por ano de idade. As idades dos documentos variaram de 1 ano até 52 anos de idade, compondo um total de 1.530 amostras.

Os documentos analisados são de uso comum em cartórios brasileiros, possuindo como características: gramatura de 75 a 90 g/m², não revestido por camadas vedantes em nenhum lado e branco em sua cor original.

Os documentos estavam armazenados no escuro, em depósito protegido do clima, sem condicionador de ar e dentro de caixas de papelão com 2 furos e separadas por ano ao qual foram armazenados, na cidade de Manaus – Amazonas. Somente foram digitalizados documentos com data impressa no mesmo. Foi utilizado um scanner de mesa da HP Modelo G2410, configurado para adquirir imagens a 300 dpi e os arquivos foram salvos com o formato TIFF que armazena as imagens sem perdas.

Cada amostra analisada foi retirada do centro do papel, em área livre de manchas, buracos e texto. Observamos que em vários documentos as áreas mais próximas as bordas dos papéis analisados apresentavam tons mais escuros do que as áreas do centro do mesmo. Como várias características do papel dependem do local onde os mesmos foram armazenados [49], um maior contato com umidade e poluição aceleram os processos de alteração do papel e conseqüentemente a cor dos mesmos.

4.2.2. Resultados

Na Figura 4.7 apresentamos 2 resultados típicos de histogramas normalizados das componentes RGB. Observamos que há uma tendência a geração de uma curva normal nos histogramas da Figura 4.7, encontramos o mesmo comportamento no restante das amostras.

Foram extraídas médias e desvios-padrões das componentes RGB, conforme o ano da amostra e com elas traçamos os gráficos da Figura 4.8. Em ambos os gráficos encontramos que a componente azul foi a mais afetada entre as três. À medida que o papel envelhece os pigmentos azuis esvaem-se e um tom amarelo-alaranjado se torna mais visível. De fato, a ausência do azul resulta na cor amarela pura. Nota-se, no entanto, que foram necessários 25 anos de envelhecimento natural para que possamos perceber alterações na cor do papel.

Após o início das alterações perceptíveis encontramos que há um decaimento de intensidade da componente azul e em seguida da componente verde com os documentos com mais de 40 anos de idade. A componente vermelha só apresentou alteração na média nos documentos com 52 anos de idade. Em todos os casos o decaimento da intensidade representa um escurecimento do papel com a idade.

Já em relação ao desvio padrão o que encontramos foi um aumento do mesmo na mesma ordem apresentada pela média, primeiro há alteração da componente azul, depois da verde e por último e menos susceptível a alteração encontramos a componente vermelha.

Verificamos também a consistência dos gráficos gerados com documentos de outro arquivo de documentos localizado na mesma cidade de Manaus - Amazonas. A Figura 4.9 mostra uma comparação entre média da componente azul e de amostras individuais por ano desta nova fonte. Verificamos que mesmo sendo de depósitos diferentes, o segundo conjunto de amostras segue o mesmo padrão de envelhecimento. Porém cabe notar que em ambos os casos as condições de armazenagem e as características dos papéis eram semelhantes. Alterações em algum destes parâmetros poderia alterar o resultado final da curva.

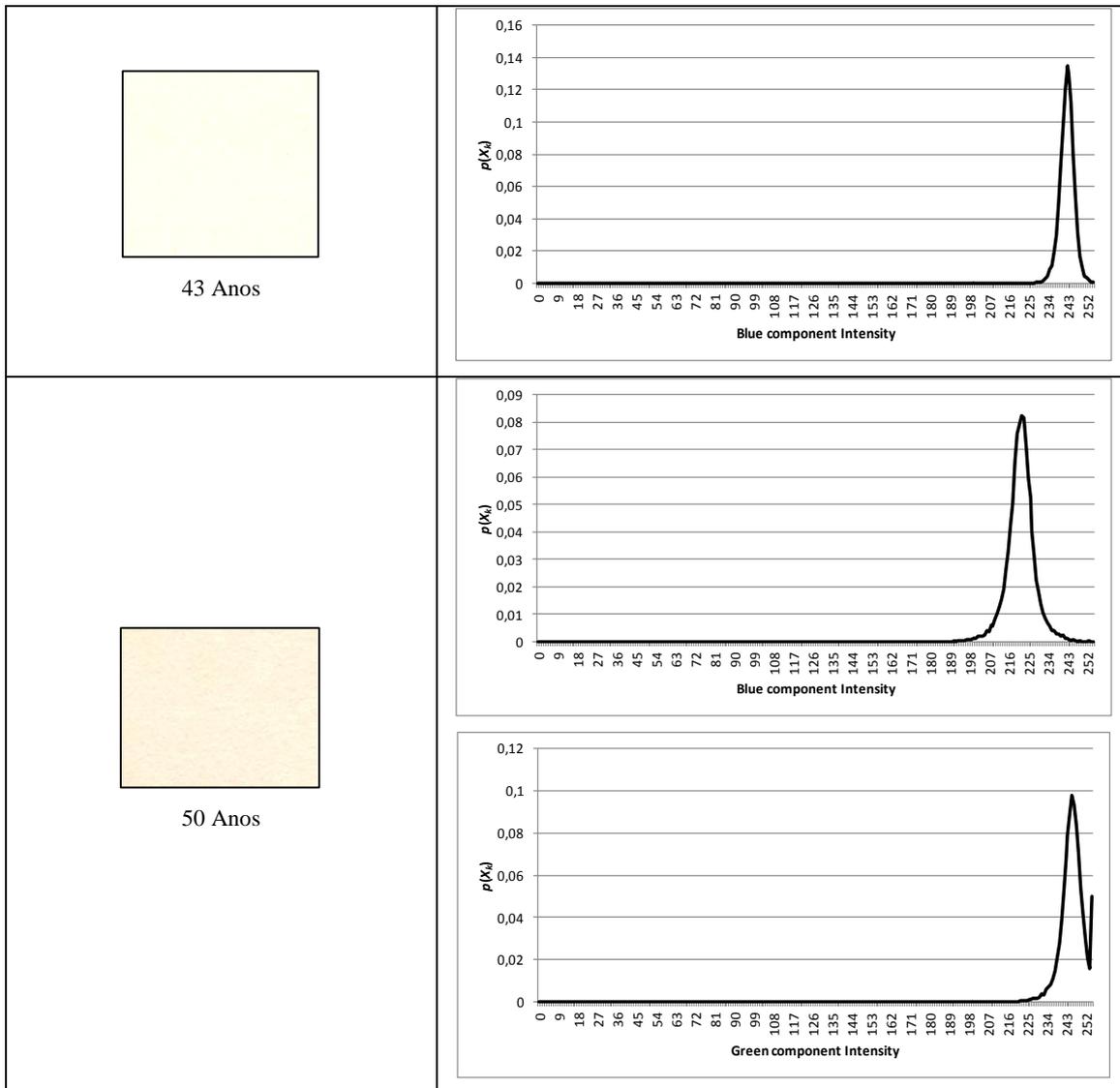
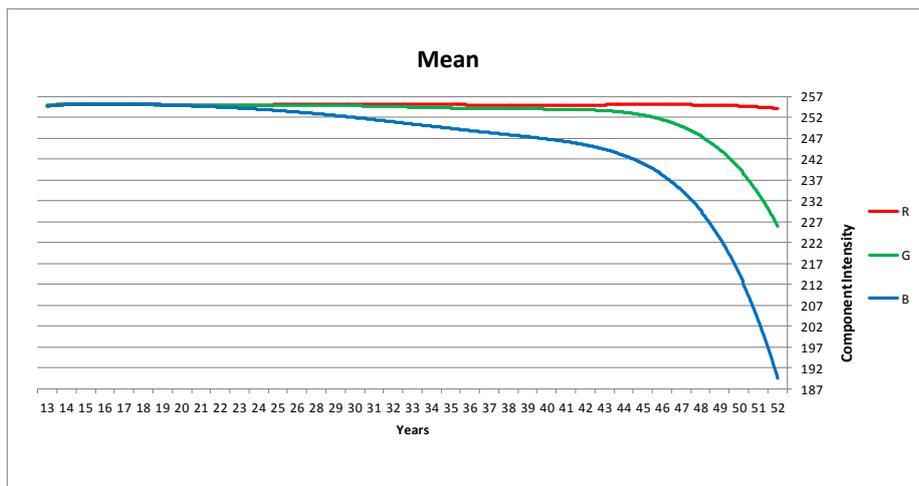
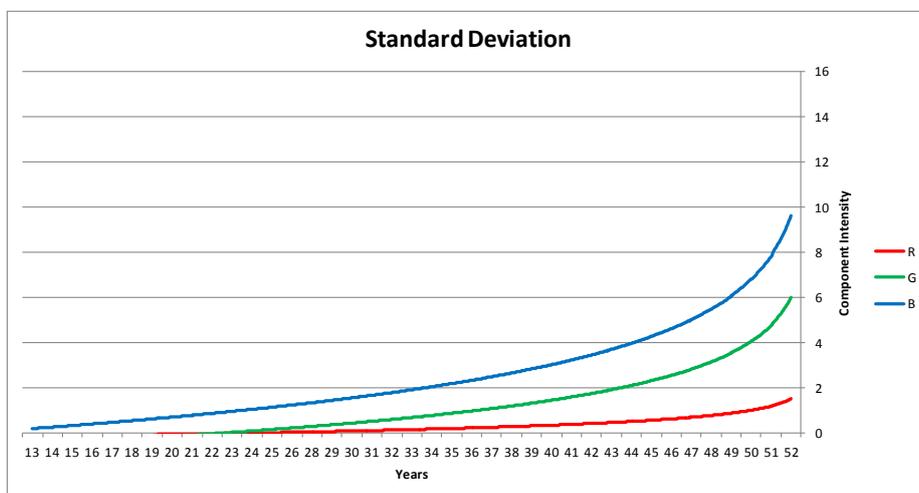


Figura 4.7: Exemplos de amostras retiradas em áreas próximas ao centro dos documentos analisados. Na imagem de 43 anos os histogramas das componentes verde e vermelhas são impulsos próximos a 255, já na imagem de 50 a normal da componente azul se deslocou a esquerda e o impulso da componente verde começa a tomar forma de uma normal.



(a)



(b)

Figura 4.8: Gráficos de Média (a) e Desvio Padrão (b) das amostras analisadas em relação a suas idades em anos.

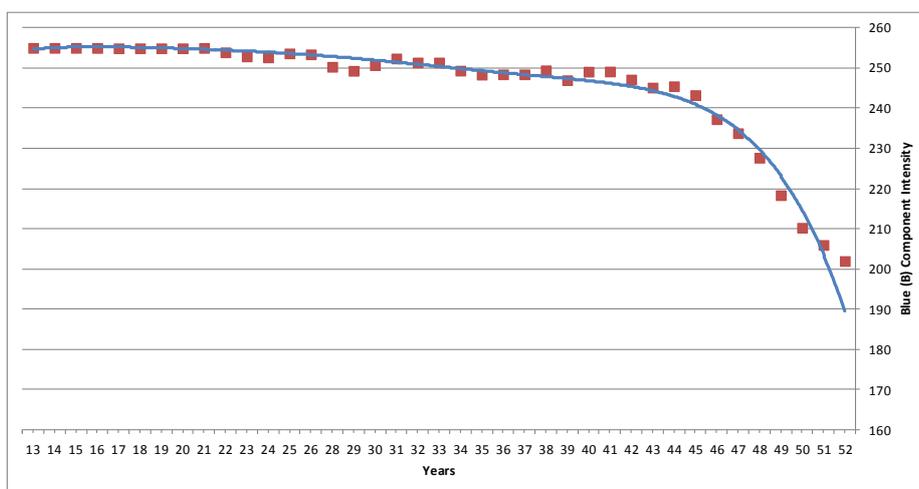


Figura 4.9: Comparação da média da intensidade da componente Azul entre duas fontes de documentos. Os quadrados representam amostras individuais de uma segunda fonte de documentos.

4.2.3. Sintetização de Texturas de Documentos Envelhecidos

Com as informações dos gráficos da Figura 4.8 podemos reconstruir o fundo de documentos recentemente envelhecidos dispondo de um gerador com distribuição uniforme.

Uma distribuição normal de média μ e variância σ^2 , $\sim N(\mu, \sigma^2)$, pode ser obtida por:

$$x = \mu + \sigma \sqrt{-2 \log_2 u_1} \cos(2\pi u_2) \quad (4.5)$$

Onde u_1 e u_2 são variáveis aleatórias de uma distribuição uniforme. Esta transformação é conhecida como transformação de Box-Muller [50].

Utilizando a fórmula (4.5) em conjunto com o gráfico de média e desvio-padrão versus idade da Figura 4.8, podemos gerar uma textura de papel envelhecido tendo como parâmetro somente a idade do mesmo.

A Figura 4.10 mostra um exemplo de textura gerada tendo como parâmetro 40 anos de idade uma das amostras de mesma idade para comparação. Comparando ambas verificamos ambas são visualmente idênticas.

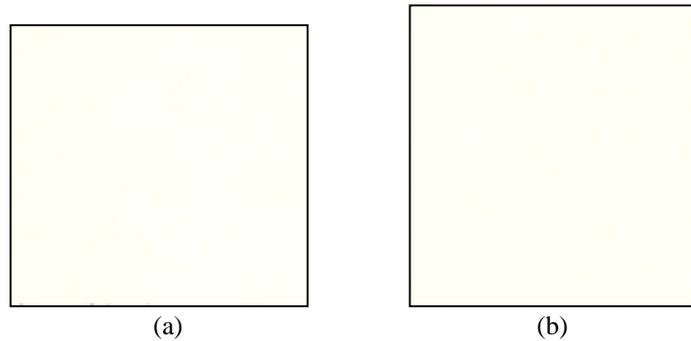


Figura 4.10: Geração de textura tendo como parâmetro de entrada somente a idade do papel. No caso idade = 40 anos. (a) Amostra real, (b) imagem sintetizada.

4.3. Conclusão

A técnica de identificação de canetas apresentado possui baixo custo comparado com técnicas forenses como o de cromatografia sendo que foi testada em uma amostra real de fraude com sucesso.

A distância de mahalanobis se mostrou mais adequada que a distância euclidiana sendo menos sensível a variações de luminosidade da fonte de aquisição das imagens.

Quanto à questão do envelhecimento de papel branco de forma natural, as curvas obtidas mostraram que há uma tendência nos tons da cor do papel à medida que o mesmo envelhece de forma natural.

5. IDENTIFICAÇÃO DE GESTOS MANUAIS PARA MANIPULAÇÃO DE INTERFACES

Turing believes machines think

Turing lies with men

Therefore machines do not think.

Alan Turing em uma carta à
Norman Routledge em 1952.

Usuários da nova *Interactive TV Digital* (IDTV) muitas vezes são confrontados com o dispositivo de interação da IDTV, o controle remoto, que a cada inovação inserida na TV reflete no crescimento do número de opções que podem ser selecionadas, seja por acesso direto (um único botão) ou pelo acesso indireto (combinação de botões), tornando a atividade de manipular a IDTV bastante difícil, principalmente a aqueles que possuem um baixo grau de conhecimento digital.

Neste cenário, a manipulação da interface através de gestos manuais surge como um modo útil e mais natural para fornecer a interação do usuário com a IDTV, diminuindo a curva de aprendizado e aumentando a usabilidade da mesma.

Neste capítulo é descrito algoritmos que fornecem a interação entre o usuário e a IDTV através de reconhecimento de gestos manuais usando o *Hidden Markov Model* (HMM), implementado

através da ferramenta Adaboost [55]. Os gestos manuais utilizados para o algoritmo de reconhecimento foram definidos por experimentos com grupo de usuários aplicando regras da engenharia da usabilidade [56, 57]. A abordagem descrita trata dos desafios que foram enfrentados nas áreas de engenharia da usabilidade, IDTV e visão computacional.

5.1. Cenário de aplicação do reconhecimento de gestos

Com a mudança do paradigma do analógico para o digital, a televisão toma forma de computador, herdando deste uma série de funcionalidades que hoje são disponibilizadas pela Web, com especial foco na interatividade, sem perder o objetivo e funções inerentes à TV convencional, tornando-se mais funcional que sua antecessora, a TV convencional, porém muito mais complexa [58]. Do ponto de vista dos usuários, as novas tecnologias acabam impondo necessidades diferenciadas para a interação do homem com o artefato computacional, sua forma de utilização e como acessá-lo de modo a torná-lo útil.

Assim, tornou-se objeto de estudo dos profissionais da área de interação Homem-Computador (IHC) um melhor entendimento para o relacionamento entre humanos e seus dispositivos eletrônicos, dando maior ênfase ao design da interface, porém sempre chegando a resultados criticados por parte dos usuários, assim como por parte da comunidade científica [56]. Desta forma, o IHC deve direcionar esforços para a criação dos softwares para a IDTV sob um conjunto de aspectos que normalmente são aplicados na criação de softwares para a Web no tocante às questões de interação como usabilidade e acessibilidade, além de observar as questões culturais, sociais e educacionais que ficam mais evidentes quando observados em populações de países subdesenvolvidos, onde há um grande contingente de pessoas com baixo nível de alfabetização digital [59]. Portanto, requisitos de ergonomia, usabilidade e de acessibilidade que venham a contribuir para um uso mais eficaz, eficiente e agradável das aplicações que comporão a programação interativa disponibilizada na IDTV devem considerar esse perfil populacional, permitindo a avaliação dos riscos positivos e negativos da nova tecnologia [56].

Este capítulo irá descrever os esforços para desenvolver um modo de reconhecimento de gestos manuais para ser aplicado ao ambiente de IDTV e assim possibilitar a interação com as opções ambiente de IDTV, como a manipulação do volume e dos canais, para fornecer um ambiente amigável ao usuário.

Este campo emergente de pesquisa é influenciado por várias abordagens em Engenharia da Usabilidade [56, 57], IDTV [57], visão computacional [55], além da integração das técnicas em um único produto [60].

5.2. Engenharia de Usabilidade

Quando se desenvolve um projeto para a TV digital deve-se pensar na convivência do usuário com o seu televisor e a interatividade realizada através da interface do controle remoto, que deve oferecer opções de abordagem simples, fácil e com alta taxa de autoaprendizagem. Além disso, ainda é necessário observar as diversas características das TVs, seja do ponto de vista de hardware, de software, além das atitudes do usuário de TV para criar uma interface útil e agradável.

Desta forma, a engenharia da usabilidade se apresenta como a etapa de um projeto que visa a criação de um produto com usabilidade onde esta deve ser compreendida pelo usuário, seja ele leigo ou não, além de objetivar a criação de um ambiente de interface confortável, eficiente e eficaz.

Assim, a metodologia adotada para dar consistência às interfaces da IDTV com o usuário busca melhorias baseadas em catalogação de informações sobre o perfil do usuário, suas tarefas e seu contexto de uso.

A análise comparativa é uma estratégia que procura observar os diversos enfoques adotados para resolver questões de design de interfaces em plataformas já implementadas [61]. Esta análise permite perceber as vantagens e desvantagens dos enfoques utilizados e serve como referência para a etapa de coordenação de interface. A etapa de coordenação de interface verifica a consistência entre as diversas partes de um sistema por meio de uma inspeção de consistência utilizando as recomendações extraídas das etapas iniciais da investigação. O produto final é a elaboração de novas propostas para os casos em que o modo existente não seja considerado adequado. A Figura 5.1 mostra o resumo das tarefas da metodologia da engenharia da usabilidade aplicada na proposta de um novo modo de interação com a TV através de gestos manuais.

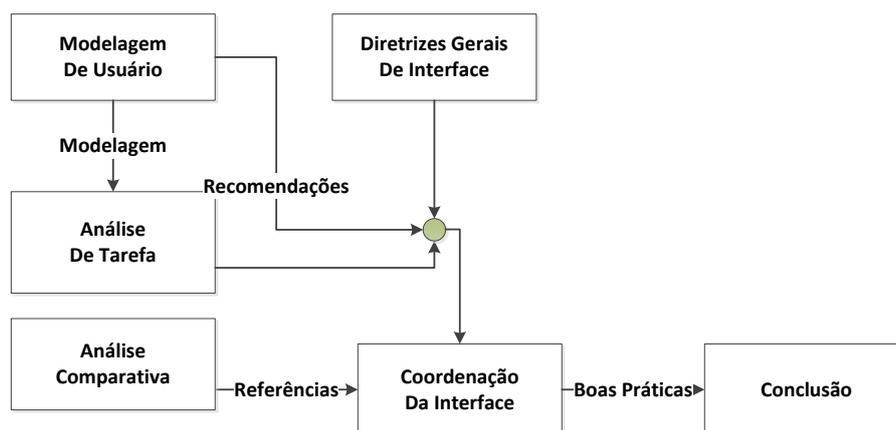


Figura 5.1: Diagrama resumido da metodologia adotada para o desenvolvimento dos algoritmos deste capítulo.

Para desenvolver a linguagem corporal, um questionário foi submetido a um grupo heterogêneo de pessoas, composto por 20 pessoas, com perfis divididos entre pessoas com experiência e pessoas

sem nenhuma experiência em utilização de equipamentos com interação gestual. As perguntas realizadas aos usuários foram:

Questão 1: Qual gesto representa a ação com a televisão: Aumentar/diminuir volume?

Questão 2: Qual gesto representa a ação com a televisão: trocar canal?

Questão 3: Qual gesto representa a ação com a televisão: movimentar cursor livremente pela tela?

Questão 4: Qual gesto representa a ação com a televisão: selecionar uma opção para ser executada?

Ao final das catalogações dos questionários, os gestos sugeridos pelo grupo mais indicados estão catalogados na Figura 5.2.

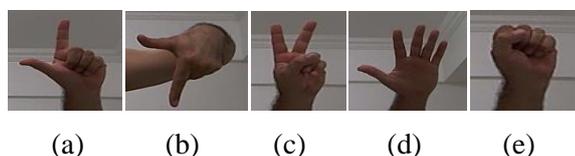


Figura 5.2: (a) Aumenta volume, (b) Diminui volume, (c) troca canal, (d) Movimenta cursor e (e) seleciona opções na IDTV.

Alguns desses gestos coincidem com gestos já utilizados em outros produtos como a *Smart TV* da Samsung LED ES8000® [83] e o controle de gestos do console Xbox, o *Kinect* [62]. Embora a *Smart TV* da Samsung já esteja no mercado, dois problemas se destacam: a câmera só reconhece o usuário se este estiver posicionado em determinado ângulo em frente à TV e a baixa qualidade dos modelos dos gestos que em muitos casos obriga o usuário a utilizar o controle remoto para realizar sua tarefa.

No final do questionário foi incluído um texto explicando com linguagem bem simples que imagens consomem muitos recursos de memória e processador, pois metade do grupo foi composto por pessoas que não dominavam conceitos de computação, e foi feito um pedido para que o entrevistado fizesse uma combinação entre os gestos sugeridos na Figura 5.2. O resultado deste questionário resultou no esquema utilizado neste capítulo.

Operação 1: Alterar volume: Pode ser executada de duas formas: o gesto (a) para aumentar ou o (b) para diminuir pode ser executado isoladamente ou exibido o gesto (c) para mostrar uma tela com os botões de volume e canal para então executar o gesto (d) para movimentar o cursor e (e) para executar a operação.

Operação 2: trocar canal: Deve ser executada da seguinte forma: o gesto (c) deve ser exibido para que uma tela de manipulação dos botões de canal e volume seja exibida. Deste ponto em diante, o modo de alteração dos canais deve ser executado da mesma forma que a Operação 1.

Após definidos os gestos e o modo de uso dos mesmos pelo grupo, os classificadores foram construídos a fim de validar a metodologia escolhida. A Figura 5.3 exibe respectivamente a mão direita aberta e a mão esquerda em punho sendo detectadas.

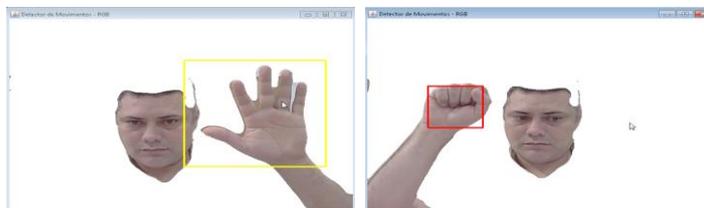


Figura 5.3: Detecção da mão direita aberta e da mão esquerda em punho após as etapas de *skin detection* e de *motion detection*.

Após detectado o gesto da mão direita aberta, sua coordenada deve ser repassada ao cursor do mouse através da classe Robot. Essa classe é usada para gerar eventos de entrada do sistema para fins de automação de teste, demos de execução automática, e outras aplicações onde é necessário o controle do mouse e teclado. O principal objetivo da classe é facilitar testes automatizados de implementações da plataforma Java [63].

5.3. TV Digital Interativa

Middleware é um termo normalmente utilizado para descrever uma camada de software que atua como um mediador entre dois programas independentes. Ou seja, sua função é trazer independência para as aplicações em relação ao hardware, permitindo que os códigos funcionem em diferentes equipamentos. Assim, os desenvolvedores de aplicações deixam de se preocupar com o hardware em si e com os protocolos existentes nas camadas inferiores do sistema de transmissão e direcionam os esforços nas interfaces para a construção dos aplicativos.

Ainda não existe um padrão de middleware universal, porém três grupos formaram padrões abertos, são eles: a Europa com o sistema DVB [64] com o *middleware* MHP, os Estados Unidos com ATSC [65] com o *middleware* DASE e o Japão com o padrão ISDB [66] e seu *middleware* ARIB. Este último serviu de base para a construção de um modelo Brasileiro que recebeu o nome GINGA [67].

Um middleware MHP é composto basicamente por uma máquina virtual Java, capaz de executar bytecode dessa linguagem, e por um conjunto de bibliotecas encarregadas de fornecer diversas funções às aplicações específicas de um ambiente de televisão e uma das mais importantes APIs do MHP é a JavaTV que introduziu o conceito de Xlet que trata-se de uma aplicação Java para TV Digital.

Como se trata de software embarcado pode surgir uma série de dificuldades inerentes a execução em um *set-top box*. Por esta razão foram construídos emuladores do ambiente de IDTV para que os aplicativos sejam testados no ambiente de PC pelas organizações de padronização da IDTV. Esta situação é comum a todos os aplicativos que são embarcados, onde uma série de adequações é necessária para relacionar a programação com a mecânica ou a característica eletrônica do hardware [68].

O *XletView* [69] é um projeto de software livre sob licença GNU GPL, desenvolvido em Java para ter sua execução independente do sistema operacional, simula uma TV Digital em um desktop baseado na *middleware* MHP executando Xlets em um ambiente para PC. Portanto, o *XletView* nos permite desenvolver e testar essas aplicações interativas (Xlets) em um ambiente de um PC antes de se preocupar com marca de set-top box a ser utilizado.

Neste projeto, foi desenvolvida uma aplicação utilizando a linguagem Java (JDK versão 7.5) e a API JavaTV (versão 1.1), para o MHP. Para sua codificação e compilação foi utilizada a IDE (*Integrated Development Environment*) Eclipse (versão 4.0) e para execução o emulador *XletView* (versão 0.3.6).

O principal elemento na construção da interface gráfica é representado pela classe *HScene*. Esta classe se assemelha ao *Frame* do pacote AWT ou *Swing* do Java e é utilizado como container principal. Esta classe apresenta algumas restrições como, por exemplo, a exibição de uma única instância durante qualquer momento da existência da aplicação.

Para realizar as operações pertinentes ao ambiente de TV foram utilizados os componentes *HAVi* (*Home Audio Video Interoperability*) [70]. A utilização deste pacote é essencial, pois existem algumas operações do ambiente de TV que não são realizáveis com os componentes nativos.

5.4. Visão Computacional

A visão computacional pode ser descrita como um complemento da visão biológica. Na visão biológica, a percepção visual dos humanos, resultando em modelos que descrevem como tais sistemas operam em termos de processos fisiológicos. Por outro lado, a visão computacional estuda e descreve sistemas de visão artificial implementados por hardware ou software.

5.4.1. Criação dos Classificadores de Gestos

Como o processo de visão computacional envolve duas etapas [55], a de criação de um modelo e a do algoritmo (que submete imagens a serem comparadas com o modelo e informa se estas foram reconhecidas), este projeto de visão computacional foi executado também desta forma.

Como modelo para manipulação das imagens, optou-se pelo modo ótico [71] em que não se utilizam marcadores, e como estratégia utilizou-se o método de Viola-Jones [72].

Como técnica para buscar as formas dos gestos em uma imagem, Haar (1910) propôs uma função, chamada transformada de Haar, que é uma transformada matemática discreta usada no processamento e análise de sinais e na compressão de dados.

Esta função proposta por Haar é um caso particular da transformada discreta de wavelet [95]. A wavelet é uma função capaz de decompor e descrever ou representar outra função (ou uma série de dados), de forma a permitir que estes possam ser analisados em diferentes escalas e sua representação gráfica é dada em forma de um pulso quadrado. Na Figura 5.4 é ilustrada a wavelet de Haar.

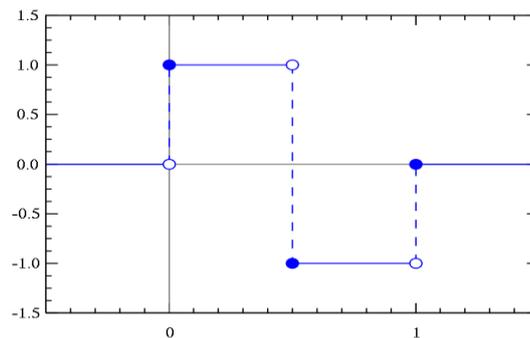


Figura 5.4 - Wavelet de Haar

Dadas as similaridades do wavelet com algumas características especiais em imagens digitais, estas foram denominadas de *Haar-Like features*. Assim, Jones *et al.* [72] propôs o uso da *Haar-Like* para codificar as características relacionadas a forma de um objeto e a existência de contrastes oriundos entre regiões em uma imagem. Um conjunto deles pode ser usado para codificar contrastes de um rosto humano, uma mão, um carro, etc.

As características da *Haar-Like* consistem em regiões retangulares, como apresentadas na Figura 5.5, aplicadas ao longo da janela de detecção. Para cada região, calcula-se a soma das intensidades de cada um dos subgrupos (branco e preto), efetuando-se a diferença entre as partes opostas. A diferença entre as partes é usada para categorizar subseções da imagem.

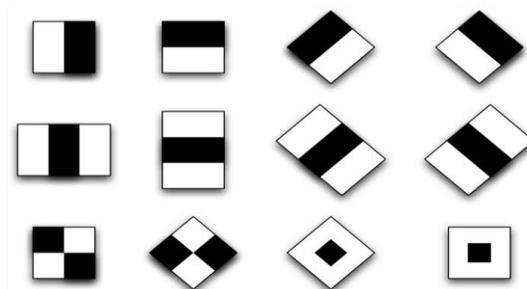


Figura 5.5 - Haar-Like Features utilizadas neste trabalho

Uma vez que as características dos gestos são mapeadas, estas compõem uma classe de gestos representando as variações alcançadas em processo de codificação. Esta classe de gestos é utilizada como modelo para buscar representantes desta classe em uma imagem.

Muitos algoritmos utilizam árvores de decisão para buscar as características dos gestos nas imagens, onde cada nó é associado a uma medida que representa a relação entre a quantidade de cada classe de objeto no nó da árvore. Neste caso, a árvore representa a classe de gestos e cada nó um objeto mapeado ou uma variação deste. Esse modo de busca de características tem como desvantagem ser sensível ao *overfitting*. O *overfitting* ocorre quando um modelo estatístico descreve o erro aleatório ou ruído em vez do objeto ou gesto desejado.

Um modo de aumentar a eficiência de uma árvore de decisão é transformá-la em um *Boosting*. Um *Boosting* é uma árvore de decisão modificada pela inserção de quebras (splits), que são efetuadas no conjunto de dados a partir de restrições nos valores de algumas das características. Deste modo, cada nó é convertido numa pequena árvore de classificação com o objetivo de diminuir a combinação entre diferentes classes num mesmo nó.

Um *Boosting* baseia-se na construção de um classificador forte a partir de uma série de classificadores fracos. O termo classificador fraco trata de uma medida de desempenho ao redor de 50% +1. Ou seja, cada classificador fraco tem apenas a obrigação de ser melhor que o acaso, porém, ao se combinar os resultados de cada classificador fraco, o resultado final é um classificador forte. Em relação ao *Boosting*, pode-se citar em particular o *AdaBoost - Adaptive Boosting*, que estendeu o *Boosting* original para torná-lo adaptativo [96].

Na prática, apenas uma característica não é suficiente para realizar uma detecção com eficiência. Por isso o *AdaBoost* extrai características e as combina em pequenas árvores de decisão, representando em cada nível um classificador fraco. À medida que o algoritmo avança, os classificadores fracos se concentram nos pontos que os anteriores tiveram os piores resultados incrementalmente melhorando a qualidade da resposta final.

Na prática, na primeira etapa do método de Viola-Jones deve-se escolher dois grupos de imagens, as positivas que contêm o objeto que se deseja mapear e as negativas, que devem conter qualquer tipo de objeto, menos o que se deseja mapear. Após definidos estes dois grupos de imagens, passa-se a utilizar 3 algoritmos fornecidos pelo conjunto de bibliotecas do OpenCV. São eles: Objectmarker, responsável por marcar nas imagens positivas os objetos de interesse, criando um arquivo contendo o nome da imagem e as coordenadas da área de marcação. Este arquivo será convertido em um vetor através da ferramenta CreateSamples, que ao mesmo tempo padroniza brilho, iluminação, e dimensiona um tamanho de janela para as imagens recortadas do grupo de imagens positivas. O tamanho padrão escolhido para as imagens deste projeto é 20 por 20 pixels.

Quanto maior for o número de imagens e as variações nestas em relação à iluminação, reflexão, fundos, escala e rotação nesta etapa, mais preciso será o classificador no final.

Viola-Jones [72] definiu que cada estágio da cascata deve ser independente dos demais, o que permite que se possa fazer a criação de uma árvore e à medida que se deseja aumentar a precisão da mesma, basta ir adicionando mais imagens e mais estágios à árvore; porém, vários trabalhos, como [73] [74] [75] apontam que para se chegar a um classificador preciso são necessárias cerca de 10.000 imagens. Para este projeto foram utilizadas 2.000 imagens, adquiridas através de um software de captura de imagens escrito em Java. Esta quantidade de imagens foi assim definida a partir de dois parâmetros: o empírico – através de experimentação na construção da árvore de características, que iniciou com 500 imagens, e onde se percebeu que à medida que se aumentava a quantidade de imagens, cada estágio ficava mais forte, melhorando o classificador no final. Outra característica observada foi voltada à qualidade das imagens utilizadas. Enquanto a literatura indica o uso de imagens com dimensões entre 640x480, este trabalho utilizou imagens com resolução de 320x240 com o objetivo de diminuir o tempo de processamento. As imagens foram obtidas a partir de câmera com resolução nativa de 12 *megapixels*. As imagens oriundas do dispositivo com qualidade superior à de webcams, aumentou consideravelmente o número de características percebidas em cada etapa, com desempenho superior ao obtido com as imagens de 640x480.

Finalmente, após estas duas etapas, o vetor de imagens positivas e a pasta contendo as imagens negativas foram submetidas ao algoritmo que realiza o treinamento e criação da cascata: o Traincascade. Este algoritmo confronta as imagens positivas com as imagens negativas, utilizadas como plano de fundo, e tenta definir bordas e outras características [76]. Esta é a etapa que leva mais tempo para ser executada, por isso foi importante acompanhar as estimativas que são exibidas na tela, e perceber se o classificador será eficiente ou não baseado nos sucessos e nas taxas de falso alarme de cada etapa. Viola-Jones indica que são necessárias, no mínimo, 14 etapas para iniciar a processo de reconhecimento de algum objeto.

O Traincascade, além de treinar as amostras submetidas a ele, gera a cascata utilizando características *Haar-Like*. Apesar da importância da determinação da textura, a detecção da forma de um objeto é um problema recorrente em sistemas de visão, por isso, [72] propôs o uso de características retangulares, ao invés das intensidades de cor para melhorar a inferência da forma de um objeto e aumentar a precisão do classificador a partir de um conceito chamado imagem integral. A partir da imagem integral é possível calcular a soma dos valores numa região retangular em tempo constante, simplificando a extração de características e acelerando o processamento da imagem.

O modo de funcionamento de uma imagem integral se dá da seguinte maneira: dada uma imagem $N \times M$, pode-se criar a imagem integral somando-se para cada pixel da imagem o valor do pixel à esquerda e acima do pixel dado.

Por motivos computacionais, a imagem integral possui dimensões $N+1 \times M+1$, com a primeira linha e a primeira coluna com valores zero. Para deixar mais claro o conceito da imagem integral, a Figura 5.6 mostra do seu lado esquerdo uma matriz representando uma Figura de dimensões 5x5, com a indicação das coordenadas adotadas e do seu lado direito a imagem integral correspondente.

Para exemplificar o modo de uso da imagem integral, será encontrada a soma dos pixels dentro do retângulo da imagem representada na matriz esquerda da Figura 5.6. Para se realizar esta tarefa é necessário somar um a um os valores da região, obtendo assim: $RESULTADO = 3+1+4+9+2+3 = 22$.

5	8	1	2	9
10	8	7	6	2
4	3	1	4	5
8	9	2	3	5
6	8	7	1	3

(a)

0	0	0	0	0	0
0	5	13	14	16	25
0	15	31	39	47	58
0	19	38	47	59	75
0	27	55	66	81	102
0	33	69	87	103	127

(b)

Figura 5.6: (a) Imagem Original. (b) Adição de linha e coluna para cálculo da imagem integral.

5	8	1	2	9
10	8	7	6	2
4	3	1	4	5
8	9	2	3	5
6	8	7	1	3

Figura 5.7: Soma dos pixels utilizando os valores diretamente na imagem de interesse.

Esta operação tem um custo computacional muito alto, inviabilizando o uso desta técnica em aplicativos de tempo real. Ao se utilizar a imagem integral, o cálculo resume-se ao acesso de quatro pontos extremos a área desejada (Figura 5.8). Assim, o cálculo resume-se a: $RESULTADO = A+C - (B+D)$. Desta forma, buscando os valores da mesma região na imagem integral obtém-se os seguintes valores: $RESULTADO = 15+81 - (27+47) = 22$. Como era esperado, foi encontrado o mesmo valor do resultado da soma dos pixels da imagem real, porém com um esforço computacional muito inferior. Mesmo que se deseje executar operações em regiões ainda maiores que o exemplo dado, o custo computacional continuaria o mesmo.

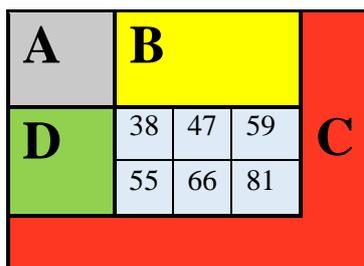


Figura 5.8: Definição das regiões em uma imagem integral.

5.4.2. Uso dos Classificadores

Foi necessário escrever um software de reconhecimento de gestos que contém todo o código necessário para ler o arquivo classificador, criado em formato *Extensible Markup Language* (XML), e inicializar as variáveis utilizadas na detecção, além de ativar a câmera e manipular o cursor do mouse mediante os resultados obtidos. Este software foi construído para lidar com múltiplos classificadores, havendo necessidade de utilização de threads para paralelizar os processos. Devido ao uso dos threads também foi dada a capacidade do suporte a multiusuários.

Vários métodos de tratamento de imagens foram utilizados para reduzir ao máximo o nível de ruído encontrado nas imagens. Entre estes citam-se: transformação morfológica [14], filtro gaussiano (*smooth*) [14] e detector de pele (*skin detector*) [79]. Neste último os principais requisitos para se obter um modelo de cor de pele estão listados a seguir.

Existem muitos objetos no ambiente que têm a mesma cor da pele. Nesses casos, mesmo um observador humano não pode identificar se uma determinada cor foi obtida a partir de uma região da pele ou de um objeto que está na imagem sem levar em conta informações contextuais. Portanto, não é uma tarefa simples construir um modelo de cor de pele que funcione em todas as condições de iluminação possíveis. No entanto, um bom modelo de cor da pele deve apresentar alguma robustez a variações das condições de iluminação. Um modelo de cor de pele humana requer um algoritmo de classificação de cor e um espaço de cores em que todos os objetos estejam representados. Existem diversos algoritmos de classificação, incluindo perceptrons multicamadas [77], mapas de auto-organização, limites lineares de decisão [78], e probabilísticos baseados na estimativa da densidade. A escolha de espaço de cores também é variada: RGB [79], YCbCr [79], HSV [79], CIE Luv [79], Farnsworth UCS [77]. O modelo de cor de pele humana adotado neste trabalho é o HSV, pois este formato acentua as tonalidades de tons de pele [80].

Os passos do algoritmo proposto para detecção dos gestos estão descritos abaixo. Os passos 1-2 são para a detecção de cor da pele, passos 3-4 são para a segmentação da pele usando borda e cor, passo 5 é pós-processamento.

Passo 1: Converter a imagem do padrão RGB para o HSV. Aplicar um filtro gaussiano de tamanho 3×3 para suavizar e tornar o mais homogêneo possível a tonalidade de pele presente na imagem.

Passo 2: Aplicar um limiar (pixels no range de 25 a 180) de tons de pele, contendo um valor mínimo e um máximo para cada camada de cor. Este limiar deve prever todos os tipos possíveis de tons de pele, mesmo que ainda assim restem alguns objetos na imagem que não são pele.

Passo 3: Aplicar detectores de borda Canny e Sobel sobre os canais de cores da imagem de entrada para encontrar os pixels de borda. Os dois filtros de borda são necessários, pois o detector de bordas Canny é adequado para detectar bordas fortes entre regiões homogêneas, enquanto que o Sobel é melhor na detecção de blocos não-homogêneos dentro de uma região da cor da pele.

Passo 4: Remover as regiões que são menores do que 1% do maior região, e regiões cuja área é reduzida para menos de 5% depois de uma operação de erosão morfológica.

Passo 5: reescrever o resultado destes algoritmos na imagem original, eliminando os objetos que não se encaixam neste padrão.

A técnica de detecção de movimento utilizada neste capítulo consistiu em realizar uma subtração de fundo, que é o método de remover os pixels que não se moveram, da seguinte forma:

- Capturar dois frames;
- Comparar as cores dos pixels em cada quadro;
- Se a cor for a mesma, substituir pela cor preta, senão manter o novo pixel;

Devido à velocidade do método em relação ao uso de qualquer classificador, a técnica de bordas foi também utilizada para ativar o sistema de reconhecimento de gestos. O modo de funcionamento ficou assim descrito: ao se movimentar a mão durante 4 segundos, o sistema ativa o algoritmo que realiza o reconhecimento dos gestos.

A Figura 5.9 mostra como o processo de detecção de pele associado ao detector de movimentos é trabalhado neste capítulo, entregando como resultado uma imagem mais simples para ser submetida ao processo de reconhecimento.

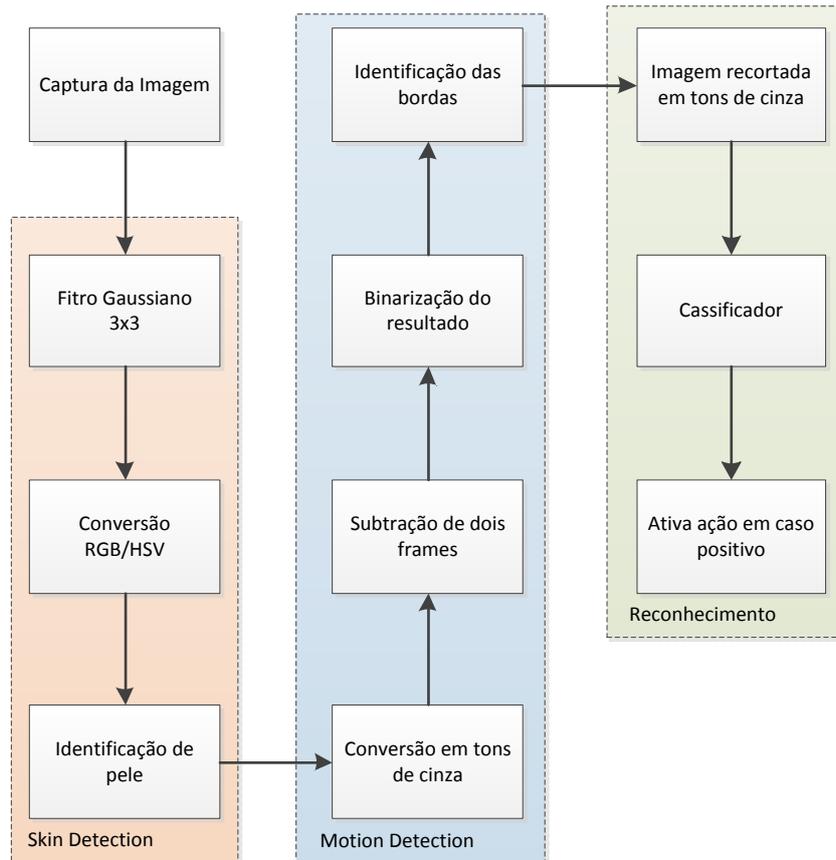


Figura 5.9: Processo de detecção da tonalidade de pele associado ao *motion detection*.

5.5. iGesture

O iGesture é a aplicação resultante da união dos projetos de usabilidade, IDTV e visão computacional, escrita com o uso da API JavaTV (um Xlet) que proporciona ao telespectador interatividade com ambiente de televisão que está sendo transmitido. Este aplicativo interage com o ambiente através de gestos manuais, que fará a execução das tarefas básicas já existentes em um controle remoto. O termo interação aqui utilizado serve para descrever o contato do telespectador com seu televisor através de uma câmera comum ou infravermelho ligada ao *set-top box*, e não o do conteúdo em si, que é um produto que depende da estação da TV (servidor de geração de conteúdo).

Um cenário foi construído para demonstrar o uso das técnicas descritas neste trabalho. Este cenário tem como características:

- Controle através de gestos com tempo de resposta rápido (cerca de 25 frames por segundo).
- Reconhecimento de gestos múltiplos.
- Utilizar webcams RGB de diversas resoluções.

A aplicação de visão computacional utiliza o componente `IplImage`, baseado na biblioteca de processamento de Imagens (IPL) da Intel® e pertencente a biblioteca `JavaCV`, que é responsável em receber e exibir as imagens obtidas da webcam. Este componente é incompatível com os componentes do `AWT`, `Swing`, `HAVi`, `JavaTV` que tratam de imagem. Por isso, a técnica aqui utilizada para exibir no ambiente do MHP as imagens adquiridas por este componente foi dividida em três etapas: (1) recebe-se o `Buffer` do `IplImage` e enviá-lo a um componente `BufferedImage`, (2) um componente `Image` recebe o conteúdo do `BufferedImage` e (3) envia o `Image` a um `JLabel`, componente pertencente ao pacote `Swing` responsável em exibir texto e imagem no Java.

Este processo mostrou-se extremamente leve e eficiente, o que justificou a reescrita da aplicação de visão computacional, substituindo os componentes da biblioteca `JavaCV` por componentes nativos do Java pertencentes as bibliotecas `AWT` e `Swing`.

Outro motivo para este modo de desenvolvimento é deixar o aplicativo de visão computacional menos dependente de bibliotecas do `JAVATV` e `JAVACV` e poder se anexado a outros aplicativos em PCs e dispositivos móveis. A classe `Reconhecer_gestos` é uma classe que implementa um `thread` e permite que o algoritmo de reconhecimento de gestos possa ser utilizado no ambiente de TV, realizando a conversão da imagem exibida no componente `IplImage` para o componente `BufferedImage`.

Com a chamada da classe que trata do reconhecimento dos gestos, colocada no `startXlet`, a aplicação MHP travava enquanto a classe de visão computacional estivesse funcionando, pois, este método do `Xlet` foi projetado para executar somente uma tarefa por vez. O `Xlet` foi modificado, incorporando ao método `startXlet` um `thread` responsável pela execução em paralelo da classe de visão computacional, eliminando assim a concorrência ao recurso de hardware.

Na `thread`, um laço monitora a webcam, capturando em média 20 frames por segundo, com resolução definida a partir da configuração da câmera utilizada. Para efeito comparativo, as câmeras atuais capturam até 30 frames por segundo, mostrando que o aplicativo, em relação ao consumo de processador, tem perda insignificante de frames e não compromete no processo de detectar o gesto e executar a ação relacionada a ele.

O layout constitui-se de uma barra de botões com as seguintes opções: fotos, música, vídeo, TV, jogos e interatividade via gestos manuais, conforme a Figura 5.10, o sistema, ao ser acionado, permite que a barra de botões possa ser controlada via controle remoto e quando ativada a opção de reconhecimento de gestos manuais, permite os dois modos de interação.

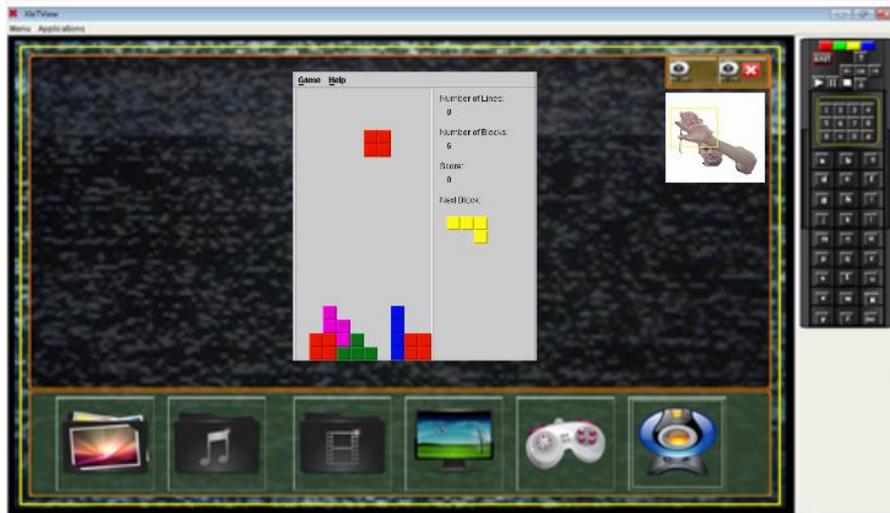


Figura 5.10: Barra de navegação do iGesture com a opção Jogo ativada.

O algoritmo construído para realizar o reconhecimento de gestos, utilizando uma webcam comum, consegue trabalhar dentro de uma margem de mais ou menos 3,5 metros. Esta distância é suficiente para atender a maioria das residências em que a distância do telespectador para o seu televisor encontra-se dentro destes limites.

Para avaliar o desempenho do método adotado neste trabalho, um banco de imagens, com 2000 arquivos próprios para cada gestos e 3000 arquivos de cenas diversas para serem utilizadas como plano de fundo, foi submetido ao algoritmo de reconhecimento.

Por se tratar de reconhecimento de gestos executados por pessoas, as imagens abrangiam diferentes raças e iluminações diversas. Os resultados detectados são mostrados na Tabela 5.1. Foi utilizado para efeito comparativo o algoritmo que realizou somente a detecção de tons de pele [81].

Nota-se pelo resultado que a grande dificuldade no método de *skin detector* está em segmentar a imagem, separando regiões de pele. O limiar não pode distinguir a diferença entre planos de fundo e a pele, realizando o corte de forma errônea. Outra vantagem percebida no método adotado neste trabalho está relacionada ao tempo de reconhecimento. No *skin detector* padrão, é necessário observar a imagem inteira para realizar a segmentação, enquanto que no método adotado, realiza-se primeiramente o *motion detector*, reduzindo drasticamente a quantidade de pixels que serão submetidos ao processo de *skin detector*.

Tabela 5.1: Comparativo de resultados entre o método deste trabalho e o *skin detector* padrão.

Resultados				
	Método 1 - Skin + Motion Este Trabalho		Método 2 – Skin Detection [25]	
	<i>True positive</i>	<i>False positive</i>	<i>True positive</i>	<i>False positive</i>
Gesto a	80%	20%	73%	27%
Gesto b	78%	22%	68%	32%
Gesto c	89%	11%	85%	15%
Gesto d	91%	9%	84%	16%
Gesto e	93%	7%	82%	18%

5.6. Otimização do algoritmo de acompanhamento de objetos através do CamShift

O CamShift é um método de rastreamento de objetos, que é uma modificação do método de controle MeanShift. A média de deslocamento em si é uma técnica não-paramétrica robusta utilizada para encontrar o modo em uma distribuição de probabilidade [74]. No CamShift, o algoritmo de MeanShift é modificado para que ele possa lidar com a mudança da dinâmica de distribuição de probabilidade da cor que é tomada a partir das imagens submetidas ao processo.

O CamShift é um algoritmo que começa com a seleção de uma região-alvo definida manualmente por um usuário, sem muito critério ou certeza de que esta área é a melhor a ser utilizada para o rastreamento de um gesto ou objeto.

Esta incerteza no momento de definir o retângulo para o objeto ou gesto pode ocasionar em erros ou na diminuição da robustez do método, pois se na seleção da região forem incluídas muitas informações do plano de fundo, estas também serão rastreadas pelo processo de busca. O algoritmo CamShift pode ser resumido através das seguintes etapas:

1. Definição da região de interesse inicial, que contém o objeto que se deseja acompanhar.
2. Criação de um histograma de cores da região contendo o objeto.
3. Fazer uma distribuição de probabilidade do quadro usando o histograma de cores.

4. Com base na distribuição de probabilidade encontrar o centro de massa da janela de busca usando o método MeanShift.
5. Centralizar a janela de busca para o ponto de tomada a partir do passo 4 e a realização de loops do passo 4 até a convergência.
6. Processar o próximo quadro com a posição da janela de pesquisa do passo 5.

Como o CamShift necessita acompanhar o gesto a partir de uma imagem colorida, o sistema de cor adotado é o HSV, retirando o componente de matiz para construir o histograma.

O CamShift será reforçado com uma região definida a partir do classificador Haar e o HSV foi ajustado para aumentar o contraste dos objetos, obtendo-se uma detecção probabilística mais pura, ou seja, realizar o processamento apenas nos pixels mais relevantes.

A adição do CamShift ao projeto, retirando a tarefa de rastreamento do Haar apresentou uma performance de 26 frames por segundo com taxa de detecção de 94%, ou seja, um ganho de performance de 23% e de 5,6% de eficiência sobre o classificador Haar.

O diagrama da Figura 5.11 mostra o fluxo completo da técnica de processamento de imagens aplicada, desde a captura da imagem até o acompanhamento do gesto realizado pelo CamShift.

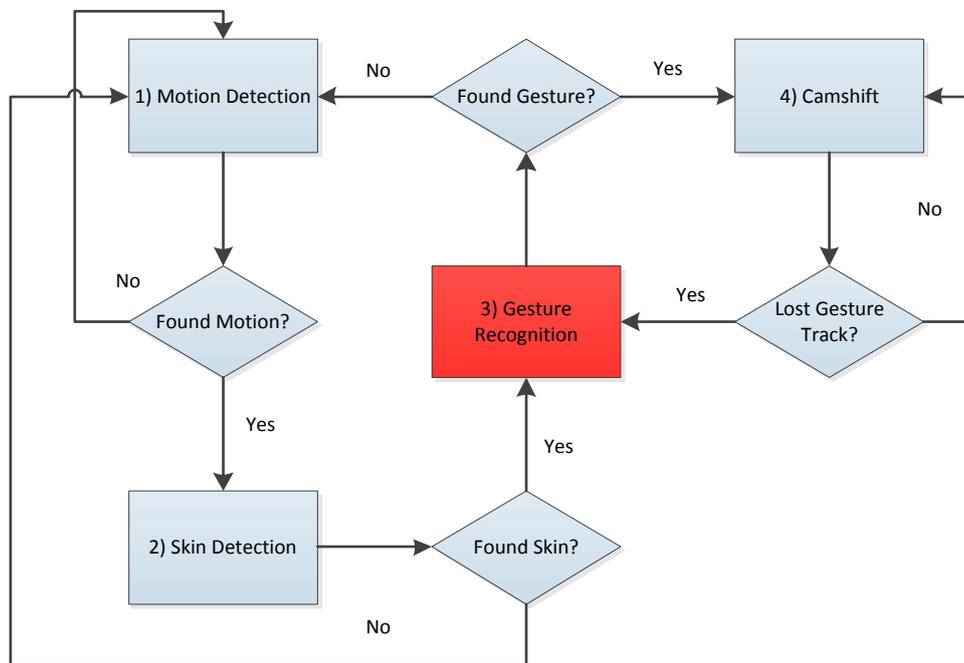


Figura 5.11: Diagrama do processo de reconhecimento de gestos utilizando motion, skin, Haar cascade e CamShift.

Para avaliar o desempenho e a eficácia do método adotado neste trabalho, foi utilizado como referência o método descrito seção anterior. O modo adotado para o desenvolvimento do projeto

possibilitou o levantamento da performance a cada adição de técnica, partindo desde a verificação da performance da câmera e a performance de construção das imagens capturadas na tela do software até a adição do método CamShift que substituiu o classificador Haar na tarefa de rastreamento, facilitando assim a construção do gráfico demonstrado na Figura 5.12.

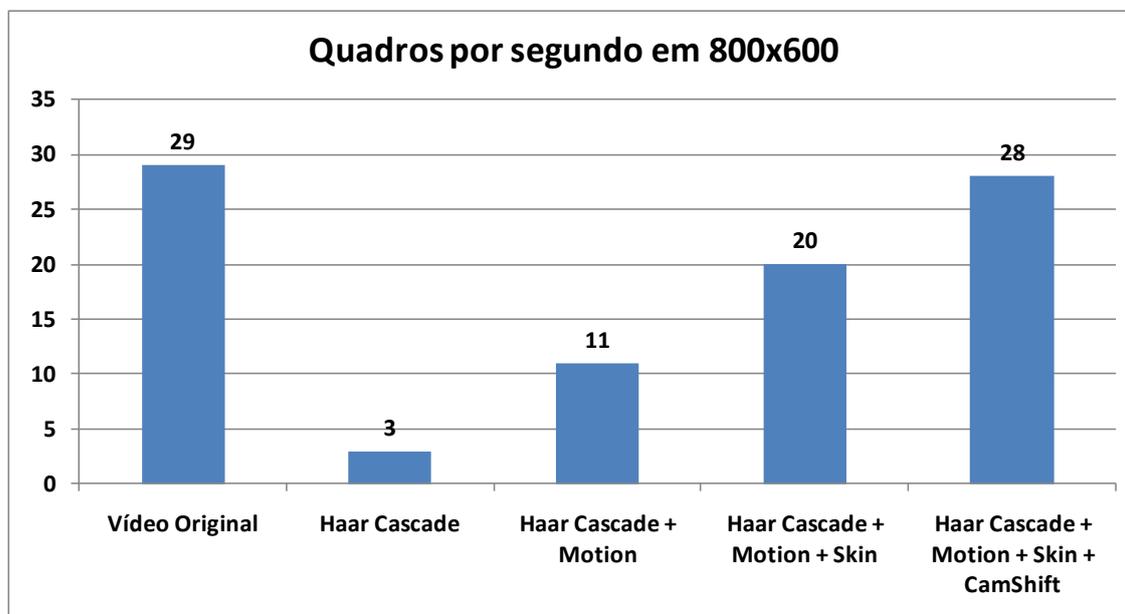


Figura 5.12: Diagrama comparativo das performances dos métodos adotados.

Nota-se pelo resultado que o método Haar tem um consumo bastante elevado de processamento quando realiza a tarefa de rastreamento do gesto mapeado no classificador. Esta percepção foi explicitada pela contagem de frames por segundo que puderam ser processados por cada método.

Este esforço é diminuído com a associação das técnicas de *Motion* e *Skin Detection* que diminuem a quantidade de bits que é submetida ao processo de comparação com o classificador, mas que ainda fica bastante distante do desempenho oferecido pela câmera. Além do fator desempenho, o classificador também é prejudicado com as mudanças de iluminação, de mudança de ângulos e rotações dos gestos apresentados. Como o valor de desempenho buscado em processamento de imagens de tempo real é o apresentado pelo dispositivo de captura, foi aplicada a técnica de CamShift após o classificador Haar para realizar o rastreamento dos gestos mapeados nos classificadores.

O CamShift é uma aplicação bem simples e computacionalmente eficiente para rastrear objetos baseados em seus padrões de cor, que mesmo que seus valores sejam alterados com as mudanças de iluminação, estes continuam a ser percebidos pelo método uma vez que o CamShift rastreia bem em ambientes ruidosos e não leva em conta a correlação cor, crescimento de regiões, considerações de contorno, suavização e predição, que são operações pesadas para o processamento computacional e que são vitais para que o Haar encontre os objetos definidos em seus classificadores.

O desempenho individual das etapas utilizadas para o reconhecimento de gestos pode ser visualizado na Figura 5.13. Nela pode-se verificar os passos que visam a eliminação de objetos irrelevantes ao reconhecimento possuem baixa demanda de processamento.

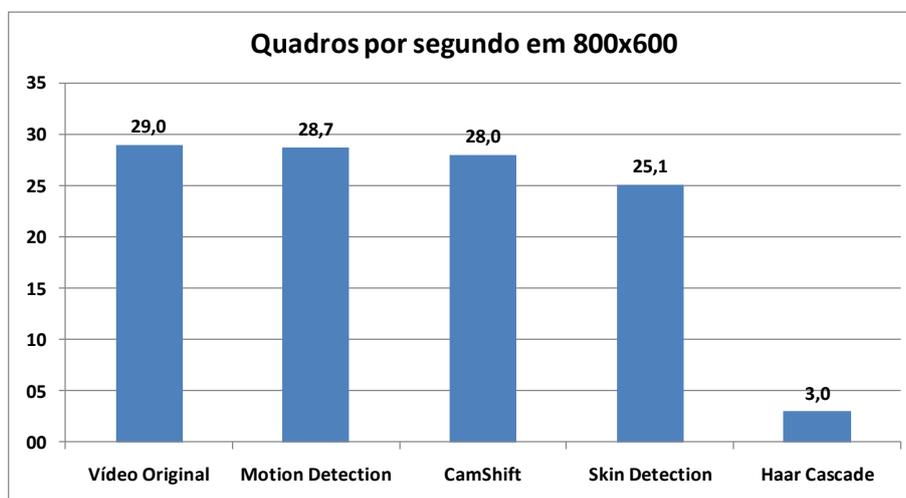


Figura 5.13: Diagrama comparativo das etapas utilizadas para o reconhecimento de gestos.

5.7. Conclusão

O reconhecimento de gestos em tempo real é uma tarefa que demanda muito recurso computacional e é vital que se busque cada vez mais se aproximar o desempenho dos algoritmos ao desempenho apresentado pelo equipamento utilizado.

A associação do *motion detection*, *skin detection* e o uso de diversas operações de processamento de imagens são importantes para melhorar a percepção dos gestos que serão entregues ao classificador, mas não são suficientes para garantir um bom desempenho.

O classificador Haar apresenta um fator que pode ser modificado para o ajuste de eficácia e desempenho, porém tratam estes dois fatores como grandezas inversamente ligadas. Ou seja, se aumentar a eficiência, diminui-se o desempenho e vice-versa, cabendo ao projetista descobrir qual a melhor relação deste fator para ser utilizado.

O uso do CamShift, apesar de ter apresentado em alguns momentos dificuldade em localizar com eficiência os gestos que tivessem grande quantidade de tonalidade ou quando este foi executado em um ambiente que tivesse exatamente a mesma cor (matiz, saturação, valor) do gesto, mostrou ser interessante na busca de gestos em tempo real pois o método é computacionalmente eficiente no processo de rastreamento de objetos em imagens coloridas, mesmo em ambientes contendo ruídos.

6. CONCLUSÕES E TRABALHOS FUTUROS

Stay hungry, stay foolish.

Steve Jobs em *Discurso em um ato de graduação da Universidade de Stanford em 12 de junho de 2005.*

O trabalho de pesquisa desenvolvido nesta tese percorreu várias facetas do processamento digital de imagens e de visão computacional. Com o intuito de desenvolver algoritmos para remoção de ruídos, foi possível trilhar os caminhos da análise forense, geração de imagens resumos e de identificação de gestos para fins de manipulação de interfaces. Em relação ao último item, o de reconhecimento de gestos, pode-se citar que é uma área de grande interesse atual devido a dispositivos como IDTVs e consoles de jogos que estão utilizando técnicas proprietárias para o mesmo. Mesmo na literatura encontram-se trabalhos discutindo soluções parecidas à demonstrada nesta tese.

Revedo os últimos trabalhos publicados foi encontrado um que utiliza parte dos algoritmos base discutidos no capítulo 5 [96]. Naquele trabalho foram utilizados processadores gráficos CUDA para se alcançar respostas em tempo real. O trabalho aqui proposto funcionou bem em processador Intel

i5, devido à remoção de objetos indesejados ao processamento da visão computacional pelas etapas de detecção de pele (*skin detection*) e detecção de movimento (*motion detection*).

As etapas de remoção de objetos indesejados foram decisões de projetos influenciadas pelo trabalho realizado no capítulo 1, de remoção de ruído, onde pela experiência adquirida se tentou identificar somente as partes das imagens que possivelmente teriam os objetos que eram essenciais aos passos posteriores.

Outra questão importante que foi notada durante o desenvolvimento desta tese foi a utilização de ferramentas, *frameworks* e bibliotecas voltados para o processamento digital de imagens, como o ImageJ e o OpenCV.

À medida que os trabalhos iam ficando cada vez mais complexos, viu-se a necessidade de utilização destes recursos para implementação dos algoritmos. Pode-se notar que à medida que os capítulos da tese avançam há um aumento da dificuldade na materialização das ideias.

O algoritmo inicial de remoção de marcadores, por exemplo, pôde ser implementado em Delphi, ImageJ e no Apple IOS sem grandes dificuldades.

Para colocação em prática das ideias do capítulo 5, a utilização da biblioteca OpenCV permitiu o término em uma fração do tempo necessário que seria para a codificação de todas as sub rotinas.

6.1. Trabalhos futuros

Para os trabalhos desenvolvidos no capítulo 2, o desenvolvimento de um plug-in para o ImageJ seria o caminho natural, já trilhado pelo algoritmo de remoção de marcadores, para o algoritmo de retirada de sublinhados em imagens binarizadas.

No capítulo 3 a separação de imagens resumos segundo diferentes cores de marcadores seria uma interessante pesquisa. Foi iniciado um trabalho com vistas a este desenvolvimento, sendo que o principal obstáculo é a identificação dos cubóides onde se concentram os marcadores de diferentes cores.

Para o capítulo 4 faz-se necessário uma interdisciplinaridade no sentido de se trabalhar com análise multivariada em conjunto com os componentes químicos das tintas esferográficas. O objetivo seria o de acelerar os testes mais precisos e caros de identificação de tinta de canetas através de uma análise das imagens digitalizadas, diminuindo desta forma o conjunto de possíveis fontes na qual o pesquisador forense irá utilizar para identificar quimicamente as assinaturas e artefatos feitos por canetas.

Os algoritmos do capítulo 5 forneceram taxas de processamento de quadros por segundo próximos à da fonte de vídeo utilizada, desta forma, um outro parâmetro necessitaria ser analisado para melhora da solução, visto que para o caso em estudo não haveria ganho na utilização de um algoritmo de

menor complexidade. Há um espaço para trabalho nos filtros utilizados nas etapas iniciais do processo. Se o usuário pudesse fazer uma auto calibração do aparato como um todo, um conjunto menor de pixels seria encaminhado às etapas posteriores, fazendo com que a necessidade de hardware pudesse ser menor.

7. PUBLICAÇÕES

Segue abaixo a relação das publicações associadas a esta tese:

- [1] BARBOZA, R. S., LINS, R. D., MATTOS, V. S. Removing Highlighting in Paper Documents. *7th IEEE/SBrT International Telecommunications Symposium (ITS 2010)*, Manaus, 2010.
- [2] BARBOZA, R. S., LINS, R. D., PEREIRA, V. M. S. Using Readers' Highlighting on Monochromatic Documents for Automatic Text Transcription and Summarization. *11th International Conference on Document Analysis and Recognition (ICDAR 2011)*, Pequim, 2011.
- [3] BARBOZA, R. S., LINS, R. D., BARROS, D. A., LIRA, E. F. Statistically Analyzing RGB Histograms to Remove Highlighting in Aged Paper Monochromatic Documents. *9th International Workshop on Graphics Recognition (GREC2011)*, Seul, 2011. LNCS. Springer Verlag, 2011. v.7423. p.251 – 259.
- [4] BARBOZA, R. S., LINS R. D., JESUS, D. M. A Color-based Model to Determine the Age of Documents for Forensic Purposes. *12th International Conference on Document Analysis and Recognition (ICDAR 2013)*, Washington, 2013.
- [5] SIMÕES, W. C. S. S., BARBOZA, R. S., LUCENA Jr., V. F. L., LINS, R. D. Use of Hand Gestures as Interface for Interaction Between Multi-Users and the IDTV. *4th Workshop on Interactive Digital TV in Emergent Countries at EuroITV 2013*. Como, Italy, 2013.
- [6] PICANÇO, W. S., ALBUQUERQUE, W. C., SIMÕES, W. C. S. S., BARBOZA, R. S., ESTEVES, A. F. A Limited Animation Framework - Structure to Handle Still Imagens Via iTVD. *4th Workshop on Interactive Digital TV in Emergent Countries at EuroITV 2013*. Como, Italy, 2013.
- [7] BARBOZA, R. S., LINS, R. D., BALDUINO, L. W. N. Filtering out Readers' Underline in Color and Monochromatic Documents. *10th International Workshop on Graphics Recognition (GREC 2013)*, Bethlehem, USA, 2013.
- [8] SIMÕES, W. C. S. S., BARBOZA, R. S., LUCENA Jr., V. F. L., LINS, R. D. A Fast and Accurate Algorithm for Detecting and Tracking Moving Hand Gestures. *International Conference VipIMAGE 2013 – IV ECCOMAS Thematic Conference on Computational Vision and Medical Image Processing*. Madeira, Portugal, 2013.

8. REFERÊNCIAS

- [1] History of pen and writing instruments - <http://inventors.about.com/library/inventors/blpen.htm>, acessado em agosto de 2013.
- [2] R. D. Lins. A Taxonomy for Noise Detection in Images of Paper Documents - The Physical Noises. *ICIAR 2009*. LNCS v. 5627. p. 844-854, Springer Verlag, 2009.
- [3] B. Wilhelm, Mark J. *Principles of Digital Image Processing - Fundamental Techniques*. London: Springer, 2009.
- [4] B. Wilhelm, Mark J. *Principles of Digital Image Processing - Core Algorithms*. London: Springer, 2009.
- [5] ImageJ – Image Processing and Analysis in Java - <http://rsbweb.nih.gov/ij/>, acessado em agosto de 2013.
- [6] G. Leedham, S. Varma, A. Patankar, V. Govindaraju, Separating text and background in degraded document images — a comparison of global thresholding techniques for multi-stage thresholding. *Proceedings of the Eighth International Workshop on Frontiers in Handwritten Recognition*, pp. 244–249, 2002.
- [7] N. A. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Sys., Man., Cyber.* 9 (1): 62–66. 1979.
- [8] J. M. M. Silva, R. D. Lins, V. C. Rocha. Binarizing and filtering historical documents with back-to-front interference. *SAC 2006*: 853-858.
- [9] R. Nagarajan, *et al.* Automated Method for Extracting Highlighted Regions in Scanned Souce. *U.S. Patent 2007/0253620*, Nov. 1, 2007.
- [10] R. Nagarajan. Automated Method and System for Retrieving Documents Based on Highlighted Text from a Scanned Source. *U.S. Patent 2007/0253643*, No. 1, 2007.
- [11] R. S. Barboza, R. D. Lins, V. S. Mattos. Removing Highlighting in Paper Documents. *VII IEEE International Telecommunications Symposium, ITS2010*. Manaus-AM, Brazil, September 2010.
- [12] P. F. Stetson, F. Sommer, A. Macovski. Lesion contrast enhancement in medical ultrasound imaging, *IEEE Trans. Medical Imaging*, 16(4), 416–425. 1997.
- [13] J. Starck, F. Murtagh, E. J. Candès, D. L. Donoho. Gray and color image contrast enhancement by the curvelet transform. *IEEE Transactions on Image Processing*, pp.706-717, 2003.
- [14] R. C. Gonzalez, R. E. Woods. *Processamento Digital de Imagens*. 3ª Ed. São Paulo: Pearson, 2010.

- [15] D. Chen, R. Ramli. Contrast Enhancement Using Recursive MeanSeparate Histogram Equalization for Scalable Brightness Preservation. *Computer Journal of IEEE Transactions Consumer Electronics*, vol. 49, no. 4, pp. 1301-1309, 2003.
- [16] R. S. Barboza, R. D. Lins, V. M. S. Pereira. Using Readers Highlighting on Monochromatic Documents for Automatic Text Transcription and Summarization. *Eleventh International Conference on Document Analysis and Recognition (ICDAR 2011)*. Beijing, China, September, IEEE Press 2011.
- [17] K. Baclawski. *Introduction to probability with R*. Ed. Taylor & Francys Group, USA, 2008.
- [18] J. Cui, H. He, Y. Wang. An Adaptive Staff Line Removal in Music Score Images. *IEEE 10th International Conference on Signal Processing (ICSP 2010)*, Beijing, China, 24–28 October 2010.
- [19] A. R. Khan, F. Kurniawan, D. Mohamad. An Automated Approach to Remove Line From Text Bypassing Restoration Stage, in *The 2nd IEEE International Conference on Computer, Control & Communication (IEEE-IC4 2009)*.
- [20] J. Kumar, D. Doermann. Fast Rule-line Removal using Integral Images and Support Vector Machines. *ICDAR 2011*, Beijing, China, 2011.
- [21] S. Deivalakshmi, B. Harinivash, P. Palanisamy. Line Removal Technique for Document and Non Document Images. *11th International Conference on Hybrid Intelligent Systems, HIS 2011*, Melacca, Malaysia, December 5-8, 2011 IEEE 2011.
- [22] J. R. C. Pinto, L. Bandeira, J. M. C. Sousa, P. Pina. Combining Fuzzy Clustering and Morphological Methods for Old Documents Recovery. *2nd Iberian Conference on Pattern Recognition and Image Analysis, IbPRIA2005*, Estoril, Portugal, June 7-9, 2005.
- [23] Página Dinâmica para Aprendizado do Sensoriamento Remoto - <http://www.ufrgs.br/engcart/PDASR/formcor.html>, acessado em agosto de 2013.
- [24] R. S. Barboza, R. D. Lins, D. M. Jesus. A Color-based Model to Determine the Age of Documents for Forensic Purposes. *12th International Conference on Document Analysis and Recognition (ICDAR 2013)*. Agosto de 2013.
- [25] C. A. B. Melo, R. D. Lins. Generation of Images of Historical Documents by Composition. *ACM International Conference on Document Engineering*, 2002, ACM Press, 2002.
- [26] M. Gallidabino, C. Weyermann, R. Marquis. Differentiation of Blue Ballpoint Pen Inks by Positive and Negative Mode LDI-MS. *Forensic Science International*, vol. 204, no. 1, pp. 169-178, 2011.
- [27] D. Djozan, T. Baheri, G. Karimian, M. Shahidi. Forensic discrimination of blue ballpoint pen inks based on thin layer chromatography and image analysis. *Forensic Science International*, Volume 179, Issue 2 , Pages 199-205, 2008.

- [28] S. G. Reibland. Ink corrosion comparison of currently used aqueous treatments for paper objects. *Forensic Science International*. 2012.
- [29] M. Sakayanagi. Analysis of ballpoint pen inks by field desorption mass spectrometry. *Journal of Forensic Sciences*, vol. 44, issue 6, 1999.
- [30] M. Ezcurra, J. M. G. Góngorra, I. Maguregui, R. Alonso. Analytical methods for dating modern writing instrumental ink on paper. *Forensic Science International*, vol. 197, 2010.
- [31] A. A. Cantu. A study of the evaporation of a solvent from a solution application to writing ink aging. *Forensic Science International*. 2012.
- [32] S. Aly, N. Tsuruta, R. Taniguchi. Face Recognition under Varying Illumination Using Mahalanobis Self-organizing Map. *Artificial Life and Robotics* 13(1), 298- 301 (2008).
- [33] R. McGill, J. W. Tukey, W. A. Larsen. Variations of Box Plots. *The American Statistician* ,Vol. 32, No. 1 (Feb., 1978), pp. 12-16.
- [34] J. Andrasko. Changes in composition of ballpoint pen inks on aging in darkness. *J Forensic Sci*, 2001;47:324–7.
- [35] A. A. Cantu, R. S. Prough. On the Relative Aging of Ink - The Solvent Extraction Technique. *Journal of Forensic Sciences*, Vol. 32, No. 5, Sept. 1987, pp. 1151-1174.
- [36] C. Weyermann, B. Spengler. The potential of artificial aging for modelling of natural aging processes of ballpoint ink, *Forensic Science International* 180 (2008). pp. 23–31.
- [37] C. Weyermann, J. Almog, J. Bügler, A. A. Cantu. Minimum requirements for application of ink dating methods based on solvent analysis in casework. *Forensic Science International* 210(1-3), pp. 52-62, 07-2011.
- [38] Audrey Giles. Forensic document examination – the science today STEP Journal. - <http://www.stepjournal.org/default.aspx?page=2164>, acessado em agosto de 2013.
- [39] S. N. Srihari, G. Leedham. A Survey of Computer Methods for Document Examination. *11th Conference of the International Graphonomics Socceity*, pp. 278-281, November 2003.
- [40] ISO 216:2007. Writing paper and certain classes of printed matter. 2007.
- [41] Wikipedia: Tamanho de Papel - http://pt.wikipedia.org/wiki/Tamanho_de_papel, acessado em agosto de 2013.
- [42] ISO 5626:1993. Paper – Determination of folding endurance.
- [43] ISO 1924-1:1992. Paper and board – Determination of tensile properties – Part 1: Constant rate of loading method. 1992.
- [44] ISO 1924-2:1992. Paper and board – Determination of tensile properties – Part 2: Constant rate of elongation method. 1992.
- [45] D. Klemm, B. Philipp, T. Heinze, U. Heinze, W. Wagenknecht. *Comprehensive Cellulose Chemistry*, Volume 1, Fundamentals and Analytical Methods. Weinheim: Wiley – VCH, 1998.

- [46] W. Wilson, J. Harvey, J. Mandel, T. Workman. Accelerated aging of record papers compared with normal aging. *TAPPI Journal* 38 (9) (1955): 543-548.
- [47] T. Collings, D. Milner. The nature and identification of cotton papermaking fibres in paper. *The Paper Conservator* 8 (1984): 59-71.
- [48] R. L. Brunelle, E. J. Speckin. Technical Report With Case Studies on the Accelerated Aging of Ball-point Inks. <http://www.forensiccriminaldefense.com/PDF/TechnicalReportWithCaseStudies.pdf>, acessado em agosto de 2013.
- [49] I. Batterham, R. Rai. A comparison of artificial ageing with 27 years of natural ageing. 2008 AICCM Book, *Paper and Photographic Materials Symposium*, 81-89, 2008.
- [50] G. E. P. Box and Mervin E. Muller, A Note on the Generation of Random Normal Deviates, *The Annals of Mathematical Statistics* (1958), Vol. 29, No. 2 pp. 610-611.
- [51] H. Pedrini, W. R. Schwartz. *Análise de Imagens Digitais*. São Paulo: Thomson. 2008.
- [52] T. Y. Zhang, C. Y. Suen. A Fast Parallel Algorithms For Thinning Digital Patterns, *Communication of the ACM*, Vol 27, No. 3, Maret 1984, pp.236-239.
- [53] A. R. Widiarti. Comparing Hilditch, Rosenfeld, Zhang-Suen, and Nagendraprasad-Wang-Gupta Thinning. *World Academy of Science, Engineering and Technology*, vol. 78, 2011.
- [54] P. Soille, *Morphological Image Analysis: Principles and Applications*. Berlin and New York: Springer-Verlag, corrected second printing of second ed., 2004.
- [55] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, 2008.
- [56] A. Sellen, Y. Yvonne, R. Harper, T. Rodden. Reflecting human values in the digital age. *Communications of the ACM - Volume 52, Issue 3, March 2009. Being Human in the Digital Age - Pages 58-66*.
- [57] C. Quico, M. J. Damásio. Televisão Digital e Interactiva: a modelação social como variável na avaliação de usabilidade. *Oficina IHC*. Brasil. 2004.
- [58] ZUFFO, Marcelo Knörich. TV Digital Aberta no Brasil – Políticas Estruturais para um Modelo Nacional. <http://www.lsi.usp.br/~mkzuffo/repositorio/politicaspUBLICAS/tvdigital/TVDigital.pdf> , acessado em agosto de 2013.
- [59] PNAD. Pesquisa Nacional por Amostra de Domicílios do IBGE: Síntese de Indicadores 2007. Rio de Janeiro: IBGE, 2008.
- [60] W. T. Freeman, C. D. Weissman. Television Control by Hand Gestures. *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, p. 179-183, Junho 1995.
- [61] J. Nielsen. Why Consumer Products Have Inferior User Experience. <http://www.useit.com/alertbox/20040315.html> , acessado em agosto de 2013.
- [62] D. Heng, T. O. TszHang. Hand gesture recognition using Kinect. Department of Electrical and Computer Engineering, Boston University, Boston, USA. 2011.

- [63] E. Etheridge. *JAVA: Graphical User Interfaces – An Introduction to Java Programming*. Ventus Publishing ApS. ISBN 978-87-496-0., 2009.
- [64] DVB-MHP. Digital Vídeo Broadcasting Multimedia Home Platform - <http://www.mhp.org>, acessado em agosto de 2013.
- [65] ATSC – Advanced Television Systems Committee Inc. Especificação e documentação do padrão ATSC - <http://www.atsc.org/standards.html>, acessado em agosto de 2013.
- [66] ARIB – Association of Radio Industries and Businesses - Especificação e documentação do padrão ARIB - <http://www.arib.or.jp/english>, acessado em agosto de 2013.
- [67] G. L. Souza Filho, L. E. C. Leite, C. E. C. F. Batista. Ginga-J: The Procedural Middleware for the Brazilian Digital TV System. In: *Journal of the Brazilian Computer Society*.
- [68] W. S. S. S. Simões, V. Lucena Jr., E. Collins, W. Albuquerque, R. Padilla, R. Valente. Avaliação de ambientes de desenvolvimento para automação do problema do cubo mágico para o robô lego mindstorms nxt. V *CONNEPI – Congresso Norte-Nordeste de Pesquisa e Inovação*. Maceió – Alagoas – Brasil. ISBN: 978-85-64320-00-0, 2010.
- [69] XleTView - <http://www.xletview.org/>, acessado em agosto de 2013.
- [70] HAVi - Home Audio Video Interoperability - <http://www.havi.org/>, acessado em agosto de 2013.
- [71] L. C. Miranda, H. H. Hornung, M. C. C. Baranauskas. Prospecting a Gesture Based Interaction Model for iDTV. In: *IADIS International Conference on Interfaces and Human Computer Interaction (IHCI) / IADIS Multi Conference on Computer Science and Information Systems (MCCSIS)*, 2009, Algarve, Portugal. *Proceedings of the IADIS International Conference on Interfaces and Human Computer Interaction*. Lisbon, Portugal: IADIS Press, 2009. p. 19-26.
- [72] M. Jones, P. Viola. Rapid Object Detection using a Boosted Cascade of Simple Features, *IEEE CVPR*, 2001.
- [73] I. F. Ince, M. Socarras-Garzon, T. Yang. Hand mouse: Real time hand motion detection system based on analysis of finger blobs. *JDCTA* 4(2):40-56 (2010)
- [74] P. I. Wilson, J. Fernandez. Facial feature detection using Haar classifiers. *JCSC* 21, 2006.
- [75] G. Monteiro, P. Peixoto, U. Nunes. Vision-based pedestrian detection using Haar-like Features. *Robotica* 2006.
- [76] W. S. S. S. Simões, V. Lucena Jr.. Remoção do Fundo da Cena para Detecção da Silhueta da Mão Humana e Detecção de Movimentos. *I SIGES - Simpósio de Informática e Geotecnologia de Santarém-PA*, Brasil, 2011.
- [77] S. L. Phung, D. Chai, A. Bouzerdoum. A universal and Robust human skin color model using neural networks. *Proc. IJCNN'01*, July 2001, pp. 2844 -2849.
- [78] K. Sobottka, I. Pitas, A novel method for automatic face segmentation, facial feature extraction and tracking. *Sig. Process.: Image Comm.*, vol. 12, no. 3, pp. 263-281, 1998.

- [79] Tsamoura *et al.* Gradual transition detection using color coherence and other criteria in a video shot metasegmentation framework. *ICIP 2008. 15th IEEE International Conference on Image Processing*, pages 45-48, 2008.
- [80] C. Garcia, G. Tziritas. Face detection using quantized skin color regions merging and wavelet packet analysis. *IEEE Trans. on Multimedia*, vol. 1, no. 3, pp. 264-277, 1999.
- [81] W. S. S. Simões, V. Lucena Jr., J. C. Leite, C. A. S. Silva. Visión por computador para manos a base de reconocimiento de gestos para la interacción com los sistemas operativos de escritorio Windows y Linux. *XXXIII UPADI - Convención Panamericana de Ingenierías*. La Habana - Cuba, 2012.
- [82] How Much Information? - <http://www.sims.berkeley.edu/how-much-info-2003>, acessado em agosto de 2013.
- [83] TV that can be controlled by hand gesture. <http://www.zoneitech.com/2012/05/03/samsung-smart-tv-es8000-tv-that-can-be-controlled-by-hand-gesture>, acessado em agosto de 2013.
- [84] The R Project for Statistical Computing – <http://www.r-project.org/>, acessado em agosto de 2013.
- [85] J. Adler. *R in a Nutshell*. 2nd Ed. O'Reilly. Sebastopol: 2012.
- [86] B. Gatos, N. Papamarkos, C. Chamzas. Skew detection and text line position in digitized documents. *Pattern Recognition*, V 30(9): 1505-1519, 1997.
- [87] C. A. B. Mello, R. D. Lins. A Comparative Study on Commercial OCR Tools In: *Vision Interface'99*, 1999, Québec. v.1. p.224 – 232.
- [88] Cuneiform PRO OCR 6.0 - <http://en.openocr.org/>, acessado em agosto de 2013.
- [89] B. S. Agarwal, J. R. Desai, S. Saha. A Fast Haar Classifier based Gesture Recognition using camShift algorithm and Curve Fitting Method. *International Journal of Computer Science and Informatics* ISSN (PRINT): 2231 –5292, Vol-1, Iss-4, 2012.
- [90] Fiftieth Anniversary of First Digital Image Marked - http://www.nist.gov/public_affairs/releases/image_052407.cfm, acessado em agosto de 2013.
- [91] S. Simon. *O Último Teorema de Fermat*. Rio de Janeiro: Editora Record, 1998.
- [92] D. Larry; D. Lars. *Digital Forensics for Legal Professionals*. New York: Elsevier, 2012.
- [93] M. E. Gondra; G. R. Grávalos. *Análise Forense de Documentos*. Campinas: Millennium Editora, 2012.
- [94] G. V. Landon. Toward Digitizing All Forms of Documentation. *D-Lib Magazine*, no 15. 2009.
- [95] Y. Bang-Hua; Y. Guo-Zheng; Y. Rong-Guo; W. Ting. Adaptive subject-based feature extraction in brain-computer interfaces using wavelet packet best basis decomposition. *Med. Eng. & Physics.*, Vol.29, pp. 48-53, 2007.

- [96] R. Lienhart, A. Kuranov, V. Pisarevsky. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection. *MRL Technical Report*, 2002.

APÊNDICE A – CÓDIGO FONTE DO FILTRO DE REMOÇÃO DE MARCADOR EM OUTRAS PLATAFORMAS

Código fonte do filtro de removedor de marcador em Objective-c

```
- (IBAction)botaoRemovedorDeMarcador:(id)sender {
    if (NULL == imagemOriginal) {
        UIAlertView *alertaSemFoto = [[UIAlertView alloc] initWithTitle: @"Alerta" message:
@"Nenhuma imagem foi selecionada\nPrimeiro escolha uma imagem" delegate: self
cancelButtonTitle: @"Ok" otherButtonTitles: nil];

        [alertaSemFoto show];
    } else {

        // count recebe tamanho da imagem, no caso, tamanho da imagem completa para que seja
aplicado o filtro na imagem toda
        int count = (imagemOriginal.size.width * imagemOriginal.size.height);
        int xx = 0;
        int yy = 0;
        int limite = 4;
        int redAndGreen, redAndBlue, greenAndBlue;

        // Primeiro obtém a imagem em seu buffer de dados
        CGImageRef imageRef = [imagemOriginal CGImage];
        NSUInteger width = CGImageGetWidth(imageRef);
        NSUInteger height = CGImageGetHeight(imageRef);
        CGColorSpaceRef colorSpace = CGColorSpaceCreateDeviceRGB();
        unsigned char *rawData = (unsigned char*) calloc(height * width * 4, sizeof(unsigned
char));
        NSUInteger bytesPerPixel = 4;
        NSUInteger bytesPerRow = bytesPerPixel * width;
        NSUInteger bitsPerComponent = 8;
        CGContextRef context = CGContextCreate(rawData, width, height,
                                             bitsPerComponent, bytesPerRow, colorSpace,
                                             kCGImageAlphaPremultipliedLast |
kCGBitmapByteOrder32Big);

        CGContextDrawImage(context, CGRectMake(0, 0, width, height), imageRef);

        // Agora seu rawData contém dados da imagem em formato RGBA8888 no formato de pixel
        int byteIndex = (bytesPerRow * yy) + xx * bytesPerPixel;
        for (int ii = 0 ; ii < count ; ++ii)
        {
            CGFloat red    = rawData[byteIndex];
            CGFloat green  = rawData[byteIndex + 1];
            CGFloat blue   = rawData[byteIndex + 2];
            CGFloat alpha  = rawData[byteIndex + 3];

            redAndGreen = abs( red - green );
            redAndBlue  = abs( red - blue );
            greenAndBlue = abs( green - blue );

            if ( (redAndGreen > limite) || (redAndBlue > limite) || (greenAndBlue > limite) )
            {
                if ( (red >= blue) && (red >= green) ) {
                    /*RED*/   rawData[byteIndex] = red;
                    /*GREEN*/ rawData[byteIndex + 1] = red;
                    /*BLUE*/  rawData[byteIndex + 2] = red;
                    /*ALPHA*/ rawData[byteIndex + 3] = alpha;
                }

                if ( (blue >= red) && (blue >= green) ) {
                    /*RED*/   rawData[byteIndex] = blue;
                    /*GREEN*/ rawData[byteIndex + 1] = blue;
                    /*BLUE*/  rawData[byteIndex + 2] = blue;
                }
            }
        }
    }
}
```

```

        /*ALPHA*/ rawData[byteIndex + 3] = alpha;
    }

    if ( (green >= red) && (green >= blue) ) {
        /*RED*/  rawData[byteIndex] = green;
        /*GREEN*/ rawData[byteIndex + 1] = green;
        /*BLUE*/ rawData[byteIndex + 2] = green;
        /*ALPHA*/ rawData[byteIndex + 3] = alpha;
    }

}

byteIndex += 4;
}

//Criando a imagem apos a passagem do filtro e jogando na tela
context = CGContextCreate( &rawData[0], width, height,
                          bitsPerComponent, bytesPerRow, colorSpace,
                          kCGImageAlphaPremultipliedLast);

CGImageRef toCGImage = CGContextCreateImage(context);
UIImage *imagemFinal = [[UIImage alloc] initWithCGImage:toCGImage];

[telaImagemFinal setImage:imagemFinal];

free(rawData);
CGColorSpaceRelease(colorSpace);
CGContextRelease(context);
}
}

```

Programa: Código fonte Removedor de Marcador em Objective-c

Código-fonte do filtro do removedor de marcador em Delphi2009

```
procedure TSDIAppForm.Button1Click(Sender: TObject);

var i, j, rg, rb, gb : integer;
    r, g, b : byte;
    limite : integer;

begin
  image1.visible := false;
  limite := strtoint(edit2.Text);
  for i := 0 to image1.Picture.Width - 1 do begin
    for j := 0 to image1.Picture.Height - 1 do begin
      r := GetRValue(image1.Canvas.Pixels[i,j]);
      g := GetGValue(image1.Canvas.Pixels[i,j]);
      b := GetBValue(image1.Canvas.Pixels[i,j]);
      rg := Abs(r - g);
      rb := Abs(r - b);
      gb := Abs(g - b);
      if ((rg > limite) or
          (rb > limite) or
          (gb > limite)) then begin
        if (r >= b) and (r >= g) then
          image1.Canvas.Pixels[i,j] := RGB(r,r,r);
        if (b >= r) and (b >= g) then
          image1.Canvas.Pixels[i,j] := RGB(b,b,b);
        if (g >= r) and (g >= b) then
          image1.Canvas.Pixels[i,j] := RGB(g,g,g);
      end;
    end;
  end;
  image1.visible := true;
end;
```

Programa: Código fonte Removedor de Marcador em Delphi2009