

UNIVERSIDADE DO EXTREMO SUL CATARINENSE – UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

VANDERVAL BORGES DE SOUZA

**PRÉ-PROCESSAMENTO DE IMAGENS PARA O RECONHECIMENTO ÓTICO DE
CARACTERES**

CRICIÚMA, JULHO DE 2011

VANDERVAL BORGES DE SOUZA

**PRÉ-PROCESSAMENTO DE IMAGENS PARA O RECONHECIMENTO ÓTICO DE
CARACTERES**

Trabalho de Conclusão de Curso apresentado
para obtenção do Grau de Bacharel em Ciência
da Computação da Universidade do Extremo Sul
Catarinense.

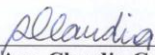
Orientador: Prof. M.Eng. Evânio Ramos Nicoleit

CRICIÚMA, JULHO DE 2011

VANDERVAL BORGES DE SOUZA


**Pré-Processamento de Imagens para o Reconhecimento Ótico de
Caracteres**

Submetido ao corpo docente do Curso de Ciência da Computação da
Universidade do Extremo Sul Catarinense como um dos requisitos para obtenção do grau
de Bacharel em Ciência da Computação.

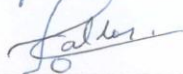


Profa. MSc. Ana Claudia Garcia Barbosa
Coordenadora do Curso de Ciência da Computação


Banca Examinadora:



Prof. M.Eng. Evânio Ramos Nicoleit (UNESC)
Orientador



Prof. M.Sc. Paracelso de Oliveira Caldas (UNESC)



Prof. Esp. Luciano Antunes (UNESC)

RESUMO

Este trabalho de conclusão de curso descreve o desenvolvimento de uma aplicação em Java com o intuito de aplicar os algoritmos de pré-processamento de imagem para a leitura ótica de caracteres. São aplicadas em imagens digitalizadas as técnicas de aprimoramento descritas na fundamentação teórica, mesmo que seja uma abordagem já conhecida a sua aplicação tende a contribuir cientificamente para o desenvolvimento de novas técnicas de implementação dos algoritmos. A aplicação desenvolvida lê uma imagem definida pelo usuário e a carrega em uma área visível, após isso a aplicação disponibiliza uma série de filtros, para que o usuário possa executá-los e obter os resultados esperados, sendo que o usuário pode definir a melhor seqüência de filtros a serem aplicados, depois disto a imagem errada é gravada na área pré-definida pelo usuário. A partir deste momento a imagem gerada esta pronta para ser submetidas ao leitor ótico de caracteres, que irá gerar um arquivo de texto, caso seja necessário o arquivo de texto gerado poderá ser submetido à métrica da distância de Levenshtein para que o usuário possa analisar os índices de eficiência dos pré-processamentos realizados.

Palavras chave: Pré-processamento de imagens, filtros, aplicação em Java, leitor ótico de caracteres, métrica da distância de Levenshtein.

ABSTRACT

This study demonstrates the development of a Java's application and has the purpose to apply the images preprocessing algorithms to read optic of characters. The techniques of improvement are applied in scanned images described in this theory fundamentation, even it's a known subject its application tends to contribute scientifically to the development of new techniques of algorithms implementation. The developed application read an user defined image and load it in a visible area, after this the application provides several of filters, to users can run them and obtain the waited results, and that the user can define the best filters to be applied, after this the wrong image is saved in a user's defined area. From this moment the created image is done to be submitted to the optic reader and it will create a text file, case be necessary, the text file created can be submitted to Levenshtein distance so the user can analyze the efficiency ratio of performed preprocessing.

Key Words: preprocessing of filters, filters, Java's application, optic reader of characters, metric de distance de Levenshtein.

LISTA DE ILUSTRAÇÕES

Figura 1. Função matemática $f(x, y)$	16
Figura 2. Cores primárias RGB e CMYK.....	18
Quadro 1. Comparativo das capacidades dos formatos de imagens.....	19
Figura 3. Imagem digitalizada do trecho de um livro.....	22
Figura 4. Scanner HP Scanjet G2410.....	28
Figura 5. Imagem embaçada da Lenna.....	33
Figura 6. Histogramas correspondentes aos quatros tipos básicos de imagens.....	34
Figura 7. Função de transformação dos níveis de cinza.....	35
Figura 8. Pontos de amarração.....	39
Figura 9. Transformação espacial.....	40
Figura 10. Sumário dos resultados morfológicos.....	41
Figura 11. Imagem de referencia 1.....	47
Figura 12. Imagem de referencia 2.....	48
Figura 13. Imagem de referencia 3.....	49
Figura 14. Imagem de referencia 4.....	50
Figura 15. Interface do programa desenvolvido.....	51
Figura 16a. Filtragem passa-alta 3X3.....	52
Figura 16b. Filtragem passa-alta 5X5.....	52
Figura 17a. Filtragem passa-baixa 3X3.....	52
Figura 17a. Filtragem passa-baixa 5X5.....	52
Figura 17a. Filtragem passa-baixa 7X7.....	52
Figura 18. Filtragem mediana.....	53
Figura 19. Filtro Gaussiano.....	53

Figura 20. Filtragem Prewitt.....	54
Figura 21. Filtragem Kirsch.....	54
Figura 22a. Filtragem Robinson com três níveis.....	55
Figura 22b. Filtragem Robinson com cinco níveis.....	55
Figura 23. Rotação de imagem.....	55
Figura 24. Algoritmo de limiarização aplicada a Java.....	56
Figura 25. Imagem de referencia 1.....	57
Figura 26. Imagem de referencia 1 corrigida.....	58
Figura 27. Imagem de referencia 2.....	58
Figura 28. Imagem de referencia 2 corrigida.....	59
Figura 29. Imagem de referencia 3.....	60
Figura 30. Imagem de referencia 3 corrigida.....	60
Figura 31. Imagem de referencia 4.....	61
Figura 32. Imagem de referencia 4 corrigida.....	62

LISTA DE TABELAS

Tabela 1. Aplicação da métrica de Levenshtein.....	64
--	----

LISTA DE ABREVEATURAS E SIGLAS

ASCII	American Standart Code for Information Interchange
BMP	Mapa de Bits
CBA	Consultores Biométricos Associados
CMYK	Cyan Magenta Yellow Black
DIB	Device Independent Bitmap
EDUSP	Editora da Universidade de São Paulo
FPAI	Filtro Passa Altas Ideal
GIF	Grafics Interchange Format
GNU	General Public License
HP	Hewlett Pacard
HTM	Hyper Text Markup
JPEG	Joint Fotografic Experts Group
LZW	Empel Ziv Welch
OCR	Reconhecimento óptico de caracteres
PBM	Portable Bit Map
PDF	Portable Document Format
PNG	Portable Network Graphics
RGB	Red Green Blue
RTF	Rich Text Format
TIFF	Tagged Image File Format
TXT	Text File
WWW	World Wid Web

SUMÁRIO

1 INTRODUÇÃO.....	11
1.1 OBJETIVO GERAL.....	12
1.2 OBJETIVOS ESPECÍFICOS.....	13
1.3 JUSTIFICATIVA.....	13
1.4 ESTRUTURA DO TRABALHO.....	15
2 IMAGENS DIGITAIS.....	16
2.1 DEFINIÇÃO E ESTRUTURA.....	16
2.2 FORMATOS.....	17
2 REPRESENTAÇÃO VISUAL.....	20
3 RECONHECIMENTO ÓPTICO DE CARACTERES.....	21
3.1 DEFINIÇÃO E CARACTERIZAÇÃO.....	21
3.2 TÉCNICAS.....	22
3.2.1 Etapas do Processo de Reconhecimento Óptico de Caracteres.....	22
3.3 PRINCIPAIS APLICAÇÕES NO MERCADO.....	24
3.3.1 Reconhecimento Óptico de Caracteres.....	24
3.3.2 Pré Processamento.....	25
3.4 SOFTWARE.....	26
3.4.1 Aplicações De Código Aberto e/ou Licença Livre.....	27
3.5 HARDWARE.....	28
3.6 PROCESSAMENTO DIGITAL DE IMAGENS.....	29
3.6.1 Métodos para processamento de imagens digitais.....	30
3.6.2 Aplicações.....	31
3.6.3 Problemas Encontrados.....	32
3.6.3.1 Inclinação da Imagem.....	32

3.6.3.2 Embaçamento.....	33
3.6.4 Principais Técnicas.....	34
3.6.4.1 Equalização de Histograma.....	34
3.6.4.2 Filtragem.....	35
3.6.4.3 Limiarização.....	37
3.6.4.4 Transformações Geométricas.....	38
3.6.4.5 Operações Morfológicas.....	40
3.6.5 Métricas.....	42
3.6.5.1 Métrica de Junker-Hoch-Dengel.....	42
3.6.5.2 Distância de Levenshtein.....	43
4 TRABALHOS CORRELATOS.....	44
5 PRÉ-PROCESSAMENTO DE IMAGENS PARA A LEITURA OCR.....	46
5.1 METODOLOGIA.....	46
5.2 DESENVOLVIMENTO DO SISTEMA.....	46
5.2.1 Imagens de Referência.....	47
5.2.2 Desenvolvimento do Aplicado para o Pré-processamento de Imagens.....	50
5.2.3 Algoritmos para o Pré-processamento de Imagens.....	51
5.2.3.1 Algoritmo para a Filtragem.....	52
5.2.3.2 Transformações Geométricas – Rotação.....	55
5.2.3.3 Algoritmo de Limiarização.....	56
5.2.4 Resultados Obtidos.....	57
5.2.4.1 Resultados Obtidos das Métricas da Distância de Levenshtein.....	62
CONCLUSÃO.....	65
REFERÊNCIAS.....	67
APÊNDICE.....	70

1 INTRODUÇÃO

O processo de OCR consiste em realizar o reconhecimento de caracteres presentes em um arquivo de imagem digital, distinguindo-os de figuras, ruídos ou outros de dados que possam estar presentes e transpô-los para caracteres em padrão *American Standard Code for Information Interchange* (ASCII). Os sistemas de OCR classificam-se em on-line e off-line.

O OCR envolve as seguintes etapas, em seqüência: aquisição de imagens; pré-processamento; segmentação; reconhecimento e pós-processamento (KUBIÇA, 1999).

A aquisição consiste em capturar os documentos e convertê-los em imagens digitais. A etapa de pré-processamento possibilita corrigir e preparar as imagens para os processos subsequentes. A segmentação identifica, dentro de uma imagem, blocos de texto, figuras, gráficos, dados, entre outros, para a etapa de reconhecimento.

A fase de reconhecimento visa classificar objetos (caracteres) nos blocos segmentados e selecionados em padrões por meio das características envolvidas. Esta etapa é altamente sensível a efeitos de deformações geométricas, tonalidades, texturas, sombras, ruídos, dentre outros. O pós-processamento realiza a análise do contexto visando eliminar ambigüidades no processo de reconhecimento, por exemplo, diferenciando o dígito zero “0” da letra O.

Neste Trabalho de Conclusão de Curso será abordada a etapa de Pré-Processamento que visa corrigir efeitos de degradação oriundos da aquisição e realçar dados relevantes (bem como eliminar dados que podem dificultar o processo) de forma a prover melhorias das respostas de OCR dos documentos. Exemplos de efeitos de degradação envolvem inclinação, ruído, falhas, dentre outros.

Podem ser usadas técnicas baseadas em equalização de histograma, filtragem, limiarização, transformações geométricas, eliminação (redução) de ruídos, afinamento ou

esqueletização, operações morfológicas, dentre outras, sendo que estas abordagens podem ser híbridas, ou seja, elas podem estar combinadas, com distintos parâmetros de utilização bem como sintonias associadas, conduzindo a distintas imagens pré-processadas e, conseqüentemente, quando submetidas a um sistema de OCR, produzindo diferentes arquivos de texto (MELLO, 2003).

Este projeto irá investigar e comparar as principais técnicas de processamento de imagens e computação gráfica aplicadas ao pré-processamento no OCR, por meio de um estudo teórico-experimental sobre cada uma delas.

O desempenho de sintonia e parâmetros utilizados será analisado por meio de métricas objetivas e ferramentas automáticas para avaliar as técnicas isoladamente e em conjunto em termos de acertos, erros e rejeições na transcrição por meio do processo de OCR sobre os textos contidos nas imagens originais.

A partir das análises efetuadas, será apresentada uma proposta do sistema de pré-processamento com os parâmetros avaliados, bem como a implementação de um protótipo de sistema com desempenho compatível ou superior aos sistemas propostos atualmente na literatura.

1.1 OBJETIVO GERAL

Estudar e desenvolver um protótipo de sistema que implemente os algoritmos de técnicas de pré-processamento aplicados ao OCR.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos desta pesquisa são:

- a) compreender o reconhecimento ótico de caracteres (OCR);
- b) compreender os fundamentos de imagens digitais;
- c) pesquisar e obter imagens de referência para testes;
- d) analisar características das imagens para reconhecimento ótico de caracteres;
- e) implementar os algoritmos de pré-processamento de imagens envolvendo transformações geométricas, computação gráfica, filtragens, dentre outras;
- f) estabelecer métricas para comparação do processo de OCR, com seus parâmetros de utilização bem como sintonia associada ao pré-processamento de imagens em OCR;
- g) realizar os testes e a validação o sistema;
- h) relatar o resultado dos testes de desempenho do sistema;
- i) avaliar as combinações de algoritmos e técnicas visando obter um eficiente processo de OCR;

1.3 JUSTIFICATIVA

Nos dias de hoje, o mercado mundial está cada vez mais voltado à eliminação e a redução na utilização de documentos em papel, devido ao processo de utilização de volumes cada vez maiores de informação com redução de espaço de armazenamento. A velocidade da localização e recuperação de um documento digitalizado propicia ganhos de eficiência e a possibilidade de vários acessos simultâneos à mesma informação, eliminando a necessidade

de fotocópias e viabilizando o processamento em paralelo das informações (OLIVEIRA, 2009).

Com o crescente aumento das demandas por OCR de documentos, surge também a necessidade de aperfeiçoar as etapas do processo, dentre elas, a etapa de pré-processamento. O pré-processamento tem como objetivo evitar que haja reconhecimento equivocado e perda de informações, tornando-o mais eficiente.

Atualmente, com as atuais tecnologias de captura de imagem envolvendo tipicamente scanners, multifuncionais e câmeras digitais, detalhes tais como inclinação, baixo nível de luminosidade, variação de cores, ruído, dentre outros fatores, prejudicam o reconhecimento, podendo gerar acertos, erros e rejeições na transcrição. Muito há ainda por pesquisar em OCR e em suas etapas. A etapa de pré-processamento de imagens é de especial interesse nesta proposta. Imagens que são capturadas por scanners ou câmeras digitais sofrem deformações que resultam em perda de desempenho no processo de OCR.

Para que os erros de reconhecimento sejam reduzidos devemos utilizar um conjunto de algoritmos na etapa de pré-processamento para que as imagens conduzam a boas taxas de acertos na transcrição dos textos contidos nas imagens originais. Não há tipicamente uma técnica de pré-processamento de imagens estabelecido que atenda a todas as necessidades de OCR, mas sim um conjunto de técnicas que podem ser incorporadas e com seus parâmetros modificados para cada tipo de arquivo a ser pré-processado. Ou seja, dentre um conjunto de técnicas existentes e bem-conhecidas, o usuário pode escolher as mais adequadas, lineares ou não, bem como a seqüência de técnicas adotadas e os parâmetros a serem utilizados.

Neste Trabalho de Conclusão de Curso busca-se pesquisar as técnicas de pré-processamento que conduzam a um protótipo de sistema com desempenho compatível ou superior aos sistemas disponíveis atualmente.

1.4 ESTRUTURA DO TRABALHO

No Capítulo 1 é apresentada a introdução do trabalho com seus respectivos objetivos, justificativas e a estrutura do trabalho.

No Capítulo 2 é apresentado as definições, estruturas, formatos e representações visuais de uma imagem digital, onde, podemos focalizar as suas principais características e funcionalidades.

O Capítulo 3 é voltado para o reconhecimento ótico de caracteres, também conhecido como OCR, neste capítulo serão apresentados as definições, técnicas, aplicações e os diversos processos que envolvem a estrutura de reconhecimento ótico de caracteres.

No Capítulo 4 serão apresentados os trabalhos correlatos, para assim justificar a aplicação de alguns algoritmos e com isso dar uma maior consistência ao conteúdo descrito no trabalho de conclusão de curso.

No Capítulo 5 é apresentada a metodologia aplicada, o desenvolvimento do sistema, a aplicação dos algoritmos e os resultados obtidos através da aplicação da métrica estabelecida.

Nas conclusões serão discutidos os pontos decisivos obtidos com a aplicação do sistema de pré-processamento de imagens aplicadas a um OCR.

2 IMAGENS DIGITAIS

Imagens são produzidas por meio de uma variedade de dispositivos físicos, como câmeras, equipamentos de radiografia, de ultrassom, radares entre outros, as utilizações destas imagens possuem diversos objetivos que vão do puro entretenimento até as aplicações sócio econômicas (MELLO, 2003).

2.1 DEFINIÇÃO E ESTRUTURA

Uma imagem digital é basicamente uma matriz onde são definidos todos os pontos ou pixels¹, onde cada ponto possui uma escala de cor. Na Figura 1 pode-se perceber melhor como a função matemática trabalha.

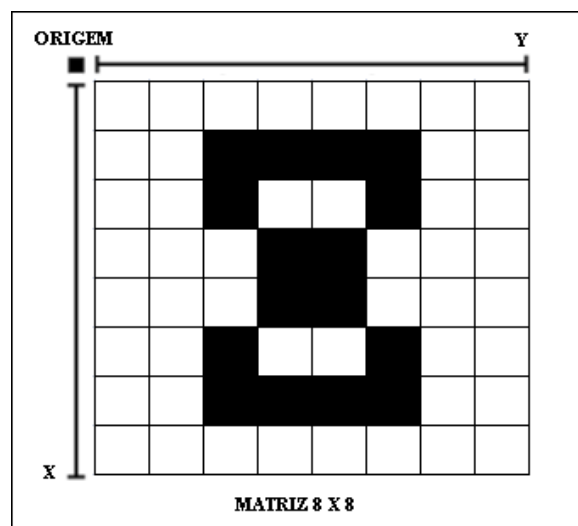


Figura 1. Função matemática $f(x, y)$

Através da função matemática $f(x, y)$ é possível encontrar a sua localização e o valor que representa a sua cor e assim manipulá-la, sendo, que sempre partiremos de um

¹ O pixel é constituído de coordenadas e informações de cor relacionadas diretamente com suas dimensões e resolução de cor da imagem (MELLO, 2003).

ponto de origem considerando ele como o número zero até a sua extremidade, onde pode haver inúmeras posições.

Para uma melhor definição podemos dizer que uma imagem digital é uma imagem $f(x, y)$ discretizada tanto em coordenadas espaciais quanto em brilho (GONZALES; WOODS, 2000), com isso podemos definir que para se obter uma imagem digital basta apenas termos uma matriz e preencher as suas possíveis posições com os níveis de cores desejados e com isso o olho humano irá conseguir interpretá-la.

2.2 FORMATOS

No formato das imagens digitais, mesmo todas as imagens utilizando a função matemática $f(x, y)$ para a formação da sua matriz, o que diferencia os vários formatos existentes é a quantidade de cores com que cada formato trabalha, a maneira de manipular estas cores e a forma como essas informações são armazenadas.

O formato *Joint Photographic Experts Group* (JPEG) é muito conhecido atualmente, pois, é o padrão mais utilizado no *WORLD WID WEB* (WWW). Este formato de imagem possui o limite de 33554432 cores e possui uma ótima taxa de compactação (LOPES, 2008).

Além do formato JPEG também existem outros formatos como PBM, DIB, GIF, PNG, GE Sygna, entre outros. O formato PBM engloba três formatos de imagens sendo eles o preto e branco chamado de *portable bitmap*, escala de tons cinzentos chamado de *portable graymap* e a colorido chamado de *portable pixmap*, todos eles sem compressão, sendo que a principal função deste formato era o de permitir a transmissão de imagens digitais.

O formato DIB é mais conhecido pela extensão de arquivo BMP que é de propriedade da Microsoft e por isso suas variantes são compatíveis com o sistema operacional Windows (LOPES, 2008).

Muito utilizada para a transmissão de imagens na rede mundial de computadores o formato GIF trabalha com mapas de cores, onde, cada pixel possui um índice que esta representada em um mapa de cores podendo ter até 256 cores.

O formato PNG possui as mesmas características do formato GIF, porem não possui limitações técnicas e restrições legais, porem, esse formato não é muito utilizado nos dias de hoje, pois utiliza apenas 256 cores e os novos hardwares conseguem suportar quantidades muito superiores a isso.

Utilizado em aplicações de medicina e um dos muitos formatos utilizado em equipamentos de TAC e IRM da General Electric o formato GE Sygna trabalha com um grupo de 65536 níveis de intensidade ou 16 bits de tons de cinza para cada pixel e isso com ou sem compressão (LOPES, 2008).

Para os formatos descritos acima existem dois tipos de modelos de cores, sendo eles o modelo RGB e o modelo CMYK, na Figura2 temos a representação gráfica dos dois modelos de cores.

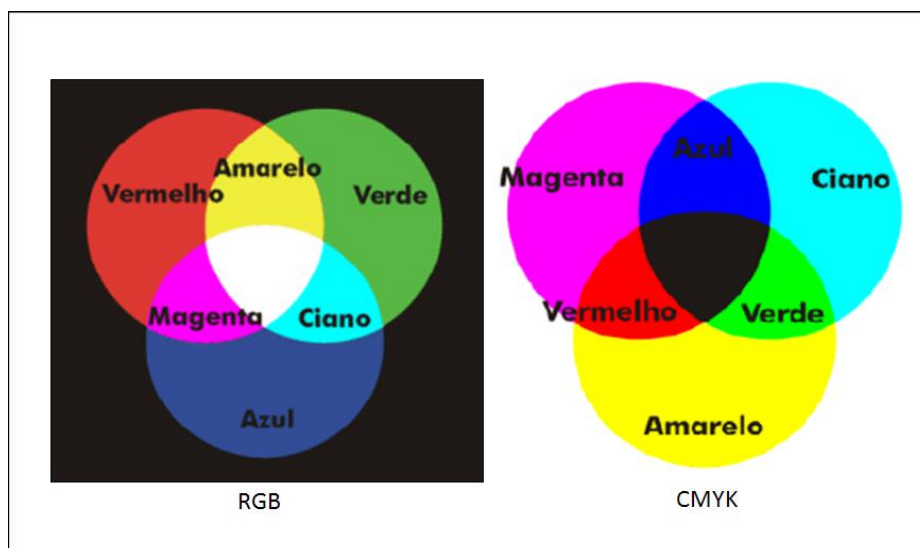


Figura 2. À esquerda as cores primárias RGB, à direita as cores primárias CMYK
Fonte: GONZALEZ, R. C.; WOODS, R. E. (2000)

Onde vemos que o modelo de cor RGB possui três cores primárias sendo elas o verde, o azul e o vermelho. O modelo de cor RGB baseia-se nas propriedades fundamentais da luz: as cores podem ser criadas a partir do vermelho, do verde e do azul (Demetrio 2003). Este modelo baseia-se num sistema de coordenadas cartesianas, enquanto o modelo de cor CMYK possui as cores primárias ciano, o magenta, o amarelo e o preto, comparando a uma impressão a cor branca recebe 0% destas cores enquanto a cor preta recebe 100%.

Em imagens digitais alguns formatos possuem uma taxa de compressão, dentre estas taxas de podemos citar a compressão com perdas e sem perdas. Entre os formatos que utilizam a taxa de compressão com perdas podemos citar o formato JPEG e um exemplo de formato que utiliza uma taxa de compressão sem perdas e que utiliza o algoritmo *Lempel-Ziv Welch* (LZW) podemos citar o formato GIF.

No Quadro 1 podemos melhor comparar as diferentes estruturas em cada um dos formatos de imagens que descrevemos acima.

	PBM	DIB	GIF	PNG	JPEG
Números de cores suportadas	2, 16.777.216	2, 16, 256, 16.777.216	2, 4, 8, 16, 32, 64, 128, 256	2, 4, 16, 56, 16.777.216, 2 ⁴⁸	33.554.432
Tipo de cor	RGB	RGB	RGB	RGB	YCBCR
Compressão	Não	RLE a 4 e 8 bits/pixel	LZW (s/ perda)	LZ77 mod. (s/ perda)	JPEG (C/ perda)
Mapa de cores	Não	só c/ 2, 16 e 256 cores	obrigatório em todos os tipos	só c/ 2, 4, 16, 256 cores	Não
Entrelaçamento	Não	Não	opcional, por linhas	opcional, por pixel	Não
Varrimento vertical	Descendente	Ascendente	Descendente	Descendente	Blocos
Transparência	Não	Não	Sim, 1 cor	Sim, plano	Não
Extensões convencionais	.PBM, .PGM, .PPM	.BMP	.GIF	.PNG	.JPG

Quadro 1. Comparativo das capacidades dos formatos de imagens
Fonte: LOPES, J. M. B. (2008)

2.3 REPRESENTAÇÃO VISUAL

Dentre as diversas formas de se visualizar uma imagem digital podemos citar como mais importantes à revelação digital em papel fotográfico, a impressão em papel fotográfico com impressora jato de tinta amadora, a tela de computador e o projetor de imagens (MACHADO; SOUKI; MAZZIEIRO, 2006).

No caso da revelação digital em papel fotográfico é utilizado um equipamento que converte a imagem digital em código binário, e depois, através de um dispositivo laser, o papel fotográfico é quimicamente alterado. Na impressão em papel fotográfico com impressora jato de tinta amadora, a impressora também lê a imagem digital e utilizando o cartucho de tinta realiza a impressão.

No computador a imagem digital pode ser visualizada sem a realização de um processamento da imagem, já que a imagem esta no formato binário, onde é necessário apenas um software para converter os arquivos binários em imagens visíveis ao olho humano. O projetor de imagens tem as mesmas características que o método de visualização de um computador, a única diferença é que a sua visualização é feita com a utilização da imagem digital sendo refletida em uma superfície branca.

3 RECONHECIMENTO ÓPTICO DE CARACTERES

O reconhecimento óptico de caracteres é a simples conversão de uma imagem em uma representação numérica adequada para o processamento digital subsequente. A aquisição destas imagens compreende dois elementos principais, o primeiro é um dispositivo físico sensível a uma faixa de energia no espectro eletromagnético o segundo converter o sinal elétrico analógico em informação digital, podendo ser representada através dos bits 0 e 1 (FILHO; VIEIRA NETO, 1999).

3.1 DEFINIÇÃO E CARACTERIZAÇÃO

O processo chamado de Reconhecimento Óptico de Caracteres (*Optical Character Recognition* - OCR) é um sistema automático que deve diferenciar os caracteres no formato ASCII das imagens e de outros dados para que com isso se obtenha apenas a parte textual da imagem.

Uma imagem ocupa muito espaço em disco e por isso a sua transposição para o formato de texto é importante, já que, um arquivo de texto é menor que uma imagem, e em um arquivo de texto é possível realizar a busca por determinadas palavras, porém é inviável a realização desta transposição de uma imagem por meio manual, isso devido aos altos custos envolvidos (MELLO, 2003).

Para trabalharmos com o OCR precisamos escolher quais os processos para a digitalização da imagem e o melhor método para realizar a transposição de imagem para texto, podendo assim ter um resultado eficiente ou ineficiente.

3.2 TÉCNICAS

A primeira parte das técnicas de reconhecimento óptico de caracteres é a aquisição da imagem que pode ser realizada através de um scanner que lê a imagem de um documento e converte os elementos escuros da página em uma matriz de pixels.

Com essa matriz gerada programa OCR identifica os espaços em branco, tornando assim fácil definir quais áreas possuem pixels que deverão ser interpretados ou não (ALVES, 2003), na Figura 3 podemos ver que o programa OCR possibilita separar os parágrafos, títulos, blocos, partes gráficas e espaços em branco.

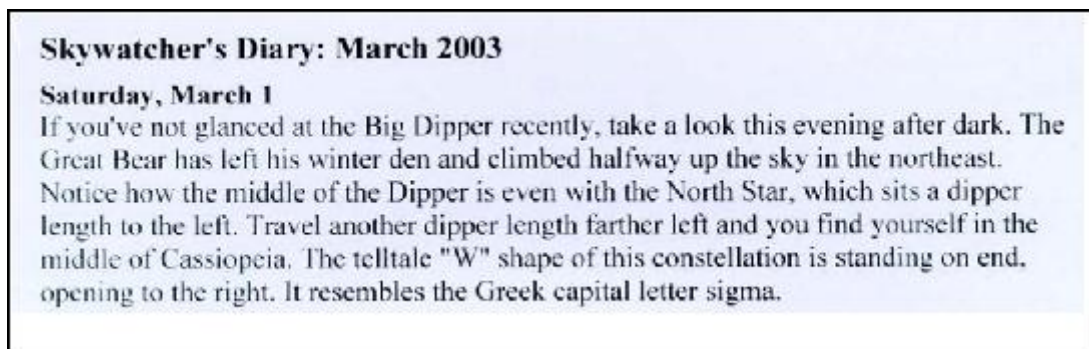


Figura 3. Imagem digitalizada do trecho de um livro
Fonte: ALVES, N. F. (2003)

Pois com o mapa da matriz de pixels o programa OCR percorre cada um desses pixels e tenta fazer uma associação entre um modelo de caracteres que o programa guarda na memória e os pixels encontrados.

3.2.1 Etapas do Processo de Reconhecimento Óptico de Caracteres

O processo de reconhecimento de caracteres possui quatro etapas distintas sendo elas, a aquisição de dados, o processamento de imagem, a análise de características e a geração do arquivo de saída.

No caso da primeira etapa a aquisição dos dados é feita com a utilização de um digitalizador, como por exemplo, um scanner ou uma câmera digital. No segundo passo o processamento da imagem é a parte mais importante para se obter um ótimo resultado, pois com ela, devemos realizar as análises de rotação, redução de ruídos, detecção de junção de caracteres e segmentação da imagem, nesta etapa é onde devemos aplicar métodos para a correção destas imperfeições. A análise de características é a etapa realizada com o propósito de identificar nos caracteres da imagem o tamanho das fontes e gerar a diagramação da imagem. Na última etapa o programa OCR gera o arquivo de saída.

Para um melhor entendimento podemos dizer que basicamente um scanner lê a imagem e converte os elementos escuros (texto e parte gráfica) num mapa de bits e com isso é gerada uma matriz de pixels quadrados que podem estar ativos ou inativos.

O programa OCR lê o mapa de bits gerado pelo scanner e pondera as áreas de pixels ativos e inativos da página, com isso podemos dizer que o programa trabalha mapeando o espaço em branco da página, com isso é possível:

- a) separar em blocos os parágrafos, colunas, títulos e partes gráficas;
- b) definir a base de cada linha através do espaço em branco entre as linhas.

Depois de separar a imagem em blocos é possível definir quantos e quais caracteres existem na imagem. Para isso o programa tenta reconhecer cada caractere através de uma comparação pixel a pixel com o modelo de caracteres que o programa guarda na memória. Os modelos são compostos de conjuntos completos: números, letras, pontuação e caracteres especiais (ALVES, 2003).

3.3 PRINCIPAIS APLICAÇÕES NO MERCADO

Nos dias de hoje, o mercado mundial está cada vez mais voltada à redução na utilização de documentos em papel, devido ao processo de utilização de volumes cada vez maiores de informação e devido a redução de espaço de armazenamento, a utilização de tal tecnologia é vital para a sobrevivência de alguns negócios.

A velocidade da localização e recuperação de um documento digitalizado propicia ganhos de eficiência e a possibilidade de vários acessos simultâneos à mesma informação, eliminando a necessidade de fotocópias e viabilizando o processamento em paralelo das informações (OLIVEIRA, 2009).

3.3.1 Reconhecimento Ótico de Caracteres

Entre as principais aplicações no mercado podemos citar aqui uma lista de serviços que processam a tecnologia OCR de entrada de dados.

Processo de fatura e movimentações em corporações financeiras: onde o programa OCR identifica a qual cliente o documento pertence e encaminha os dados capturados para as suas centrais de dados, temos, por exemplo, as empresas, Price water house Coopers Auditores Independentes, Milan e Milan Advogados Associados, Renato Oliveira Advogados;

Exploração de formas e tradutores: utilizado para reconhecer manuscritos antigos, facilitando a sua tradução automática, proporcionando assim um retorno mais ágil e preciso, para este formato temos como exemplo as empresas, Editora da Universidade de São Paulo Edusp;

Processo de reivindicações e pesquisa: ao preencher um formulário padronizado o programa OCR capta os dados e alimenta uma base de conhecimento com estes dados, com

isso o curto espaço de tempo e a grande quantidade de informações são facilmente absorvidas, exemplo de empresas que utilizam este programa, Associação Comercial de São Paulo, Sociedade Brasileira de Cirurgia Dermatológica, Associação Brasileira de Metalurgia e Materiais;

Conversão de livros para arquivos: com a racionalização de espaço e a grande quantidade de papel que um livro consome a sua conversão para um documento de texto é extremamente importante e a maneira mais prática de se realizar esta tarefa é através de um programa OCR.

3.3.2 Pré-Processamento

Como nos dias atuais a exigência por uma maior agilidade na captura e conversão de imagens em texto é cada vez maior, o pré-processamento da imagem torna-se fundamental, pois quanto mais refinada a imagem estiver melhor será o resultado, dando assim uma maior credibilidade ao programa OCR.

Sensoriamento Remoto e Sistemas de Informação Geográfica: como a maioria destas imagens é gerada através de satélites e equipamentos de sensoriamento as imagens obtidas acabam distorcidas, com isso o programa OCR executa o pré-processamento desta imagem antes de aplicar os métodos de reconhecimento.

Aplicações em pesquisas voltadas à inteligência artificial: neste caso podemos citar a pesquisa sobre a visão artificial baseada em redes neurais, onde o sistema capta uma imagem e aplica os métodos de pré-processamento para que a imagem fique totalmente adequada as necessidades da pesquisa.

Aplicação em pesquisas na área da medicina: assim como nas pesquisas voltadas a inteligência artificial, as pesquisas voltadas a área da medicina precisam ter o máximo de perfeição nas imagens.

Essas imagens são obtidas com a utilização de aparelhos, como por exemplo, ultrassom, raios-X, ultrassonografia, ressonância magnética, entre outros, o problema é que nem sempre a imagem obtida por esses aparelhos esta bem definida e com isso acaba atrapalhando o resultado final do trabalho, para haver uma redução do índice de falhas o programa de pré-processamento de imagem captura esta imagem e converte o seu dado na forma desejada pelo pesquisador.

Uma das aplicações práticas para este tipo de pré-processamento é a aplicação em pesquisas com o intuito de obter melhores resultados em imagens de ressonância magnética (SEBASTIANI, 2009).

3.4 SOFTWARE

Existem no mercado mundial, várias aplicações com o intuito de converter imagens em arquivos de texto, essas aplicações são chamadas de OCR, atualmente para o uso da grande maioria destas aplicações é necessário comprar uma cópia para poder utilizá-la, porem ainda existe uma pequena fatia no mercado de aplicações OCR que possuem o código aberto possibilitando o desenvolvimento de novas aplicações e assim distribuí-las. Com a utilização deste tipo de aplicação é possível converter dados que estejam em diferentes tipos de linguagem, bastando apenas possuir um dicionário de dados interno onde o software faz as comparações e exhibe os resultados.

3.4.1 Aplicações de Código Aberto e/ou Licença Livre

Com as aplicações de código aberto podemos abri-la, analisar a sua estrutura e a forma com que converte uma imagem em arquivo de texto, com isso podemos decidir qual será a melhor forma de tratar a imagem com métodos de pré-processamento antes de submetê-la a aplicação OCR e assim obter os melhores resultados possíveis.

Atualmente podemos dizer que existe as seguintes aplicações OCR com o código aberto e/ou a sua licença livre, Tesseract, GOCR, TopOCR, Ocropus, OCRAD, entre outros (HENG, 2008), abaixo temos uma descrição resumida de cada aplicação OCR destacada acima.

Tesseract é uma aplicação OCR de qualidade comercial desenvolvida originalmente na HP, entre 1985 e 1995, atualmente patrocinada pela Google esta aplicação funciona nos sistemas operacionais Windows e Linux e ela pode reconhecer cerca de seis idiomas (TESSERACT, 2011).

GOCR é uma aplicação disponível para os sistemas operacionais Linux, Windows e OS/2, é liberado sob o código aberto General Public License (GNU) e a sua forma de trabalhar é por linha de comando (JOCR, 2011).

TopOCR foi projetado exclusivamente para câmeras digitais e smartphones, porem também pode ser utilizado em imagens obtidas através de qualquer outro tipo de equipamento. A suas características são que ele pode reconhecer o texto a partir das imagens de pior qualidade, suporta 11 idiomas e ele é capaz de obter imagens diretamente do scanner ou da câmera e suporta os formatos JPEG, TIFF, GIF e BMP, é uma aplicação de licença livre mas é utilizada apenas para o sistema operacional Windows (TOPOCR, 2011).

Ocropus é outro software só que para utilizar este aplicativo é necessário compilá-lo, pois ele é distribuído em forma de código fonte e apenas para o sistema operacional Linux.

Baseado em plug-ins e no motor Tesseract para o funcionamento do seu motor de reconhecimento de caracteres, possui uma análise do layout, linguagem de modelagem estatística natural, capacidade multilíngue (OCROPUS, 2011).

OCRAD está disponível para Linux e apenas na forma de código fonte é um utilitário de linha de comando OCR que aceita arquivos no formato de PBM e PGM, é capaz de lidar com vários textos de coluna ou blocos de texto (OCRAD, 2011).

3.5 HARDWARE

Como todos os softwares necessitam de hardware o reconhecimento ótico de caracteres não é diferente, além dos equipamentos convencionais também são utilizados os hardwares responsáveis pela captura de imagens como, por exemplo, um scanner, uma câmera digital, um fax, entre outros.

Para o reconhecimento ótico de caracteres a qualidade da imagem obtida é essencial e para isso devemos analisar a capacidade que o hardware possui antes de adquiri-lo. Para analisarmos a capacidade devemos levar em consideração a sua resolução, quantidade de cores, e armazenamento.

Na Figura 4 vemos um scanner de ótima qualidade e ideal para a sua utilização no reconhecimento ótico de caracteres.



Figura 4. Scanner HP Scanjet G2410
Fonte: Hewlett-Packard (2009)

Segundo o fabricante de hardware Hewlett-Packard (HP) o Scanner HP Scanjet G2410 possui uma resolução de 1200 dpi com 48 bits, suporta os sistemas operacionais Windows, Linux e Macintosh. Os formatos suportados pelo scanner são: BMP, JPEG, GIF, TIFF, PNG, PDF, RTF, HTM, TXT.

3.6 PROCESSAMENTO DIGITAL DE IMAGENS

Definido como o processo de análise, manipulação e síntese obtidas através de técnicas matemáticas e de forma algorítmica, com a exceção da aquisição e da exibição da imagem todo o processamento é feito por meio da utilização de um software (GONZALEZ; WOODS, 2000).

O processamento de imagens é basicamente um conjunto de técnicas e ferramentas que constituem um sistema computacional. A lógica do processamento de imagens é simples, se parte de uma imagem de origem e utiliza os métodos de processamento chega-se a um resultado.

Dentre as aplicações, podem ser citadas: auxílio ao diagnóstico clínico, avaliação e acompanhamento de procedimentos cirúrgicos e radioterapêuticos (MAINTZ, 1998).

Outro ponto importante citado é que o processamento de imagens é extremamente específico, ou seja, cada solução técnica é responsável apenas para o seu objetivo, por isso se um tipo específico de processamento de imagens for aplicado de forma inadequada pode danificar a imagem ou reduzir a sua qualidade (GONZALEZ; WOODS, 2000).

3.6.1 Métodos para Processamento de Imagens Digitais

Entre os métodos utilizados para a execução do processamento digital de imagens podemos citar a codificação, compressão segmentação, análise de características e filtragem.

Na codificação é utilizado um método que utiliza o mapa de pixels e uma série de técnicas para que com isso possa definir a melhor seqüência de informação e tornar a imagem menor.

A compressão de uma imagem é responsável por eliminar os pontos redundantes, tornando assim menor o tamanho da imagem (GONZALEZ; WOODS, 2000).

Segmentação é um método que consiste em dividir a imagem em partes diferenciadas entre si, melhorando o processo de reconhecimento, para isso são utilizadas duas técnicas chamadas de similaridade e descontinuidade entre pixels. Com isso podemos dizer que a segmentação é um processo empírico e adaptativo que procura adequar se às características particulares de cada tipo de imagem e aos objetivos que pretende alcançar (MELLO, 2003).

Na análise de características o processamento da imagem procura identificar o tamanho das fontes e a análise de diagramação facilitando assim o processo de reconhecimento dos caracteres da imagem.

Em filtragem o método consiste na remoção de características indesejáveis deixando a imagem mais nítida e com um melhor realce para que o reconhecimento da imagem se torne o mais preciso possível.

3.6.2 Aplicações

Desde muito tempo o processamento de imagens esta presente em praticamente todos os ramos de nossa vida (MARQUES; VIEIRA, 1999).

Com essa afirmação podemos destacar as principais aplicações no mercado atual como sendo a meteorologia, transmissão digital de sinais de televisão ou fax, análise de imagens biomédicas, ultrassom, tomografia computadorizada, automação industrial, sensores visuais robóticos.

Na área da meteorologia nós encontramos a aplicação chamada de METMAN, onde é uma aplicação comercial que é responsável por gerenciar dados meteorológicos através de imagens obtidas por sensores, podendo assim exibir as imagens em tempo real. Na mesma área também é possível encontrar uma aplicação gratuita voltada para o sistema operacional LINUX, essa aplicação foi escrita por um suíço chamado Andreas Muller e tem o nome de METEO.

Na transmissão digital de sinais de televisão foi encontrada a aplicação chamada de Ginga, muito conhecida no Brasil por ter o seu código aberto permitindo o desenvolvimento de aplicações NCL interativas para a TV Digital de forma independente da plataforma de hardware dos fabricantes de terminais de acesso.

Para as análises de imagens biométricas é possível citar o sistema de passaportes BR PASS desenvolvido pelos Consultores Biométricos Associados (CBA), mesmo sendo uma aplicação comercial é possível ver quanto o processamento de imagem é importante para este tipo de ramo de atividade, a mesma empresa possui outra aplicação chamada de estação biométrica universal que segundo a CBA é capaz de capturar e armazenar informações biográficas, impressões digitais, impressões manuais, fotografias e assinaturas, além de enviá-las e/ou imprimir relatórios com as informações armazenadas.

Em todos os ramos de atividade destacados anteriormente é possível encontrar aplicações voltadas para o processamento de imagens, mesmo sendo nível comercial ou de licença gratuita.

3.6.3 Problemas encontrados

Devido a sua má utilização ou as condições do equipamento utilizado para o processamento da imagem é muito comum encontrar problemas, em uma imagem digitalizada é possível visualizar vários problemas entre eles podemos citar as imagens inclinadas, embaçamento, foco distorcido, entre outros problemas.

A utilização dos equipamentos corretos e da padronização correta para se obter uma imagem torna o seu processamento mais otimizado e com isso o reconhecimento dos caracteres da imagem será mais confiável.

3.6.3.1 Inclinação da Imagem

A inclinação de uma imagem em relação ao seu eixo é gerada devido a um mal-posicionado de um documento sobre a mesa de digitalização, e com isso gerar erros no processo de reconhecimento de caracteres.

Nos métodos de correção para este problema podemos citar principalmente o uso da Transformada de Hough, e o sistema de detecção de grau de rotação. Para realizar os cálculos responsáveis pela correção deste problema podemos dizer que consiste em copiar os pixels que estão organizados em linhas e reordenando-os em colunas tendo assim uma imagem correta para as imagens com ângulos de 90° (GONZALEZ; WOODS, 2000).

Para as imagens que possuem uma rotação por ângulos diferentes a rotação de cada ponto (X, Y) de uma imagem por um ângulo arbitrário Ang , mapeará este ponto na localidade de coordenadas (X', Y') , onde X' e Y' são calculados pelas equações:

$$X' = X \cos(Ang) + Y \sin(Ang)$$

$$Y' = Y \cos(Ang) - X \sin(Ang)$$

Com a aplicação destes métodos e equações matemáticas é possível deixar uma imagem que está inclinada ou fora do ângulo correto na sua posição correta e assim obter um melhor resultado no reconhecimento dos caracteres (ALVES, 2003).

3.6.3.2 Embaçamento

Geralmente o grande responsável pelo embaçamento das imagens digitalizadas é o ajuste de resolução ou até mesmo à distância. A Figura 5 é uma típica representação de uma imagem embaçada.

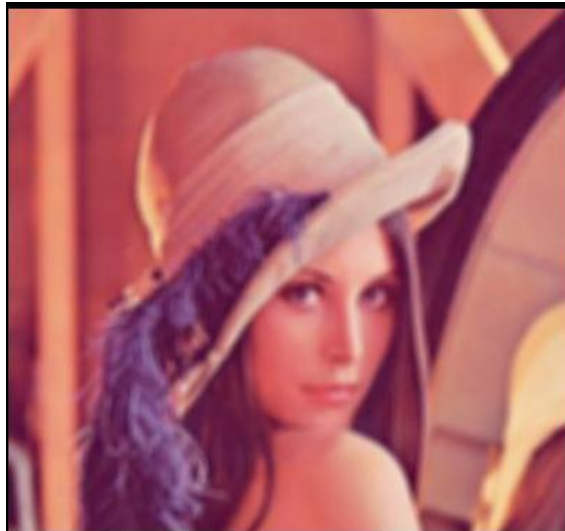


Figura 5. Imagem embaçada da Lenna
Fonte: FREEONLINEPHOTOEDITOR (2009)

Esse embaçamento ocorre por causa do sensor calcular a entrada de luz e não o embaçamento em si, isso ocorre provavelmente por terem sido digitalizadas com uma

distância grande e utilizar uma resolução alta demais e sem a utilização de uma ferramenta de nitidez essa imagem ficou embaçada.

3.6.4 Principais Técnicas

O principal objetivo com a correção das imagens é que o resultado seja mais apropriado que a imagem original, porém cada correção é tratada especificamente para um determinado problema, sendo assim é possível afirmar que a correção aplicada a imagem com o propósito de realçar imagens de raios-X pode ou não funcionar em imagens com outras finalidades (GONZALEZ; WOODS, 2000).

3.6.4.1 Equalização de histograma

A equalização de histograma é capaz de gerar uma estimativa da probabilidade de ocorrência do nível de cinza, onde $p(r, k)$ é a forma de realizar tal estimativa. Na Figura 6 temos os histogramas correspondentes aos quatro tipos básicos de imagens.

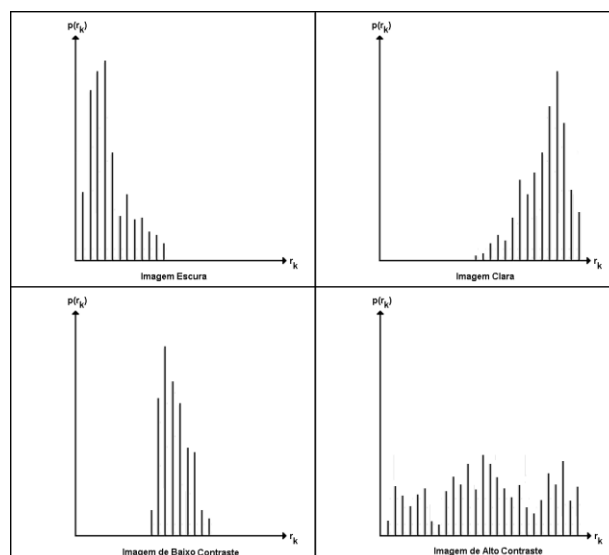


Figura 6. Histogramas correspondentes aos quatro tipos básicos de imagens.
Fonte: GONZALEZ, R. C.; WOODS, R. E. (2000).

Quando os níveis de cinza estão concentrados nas extremidades escuras podemos dizer que a imagem é praticamente escura, do contrário a imagem é clara, porém, se indicar uma escala dinâmica pequena, então essa imagem possui um baixo contraste e se a imagem possuir um grande número de espalhamento então é uma imagem de alto contraste. Na Figura 7 temos a representação de uma transformação de níveis de cinza.

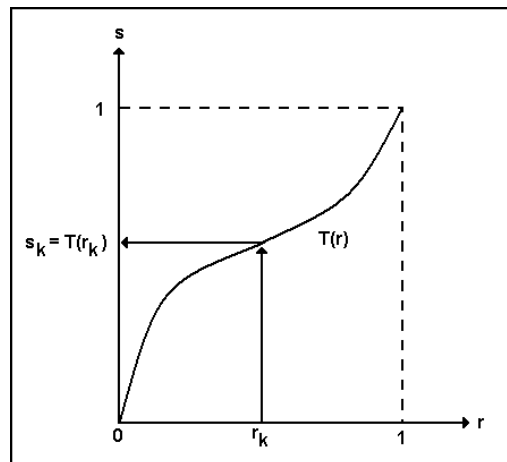


Figura 7. Função de transformação dos níveis de cinza
Fonte: GONZALEZ, R. C.; WOODS, R. E. (2000).

A função $T(r)$ que é de uma variação única e crescente no intervalo entre $0 \leq r \leq 1$ e com isso ela preserva a ordem de preto para o branco em escala de cinza, enquanto a função $0 \leq T(r) \leq 1$ para $0 \leq r \leq 1$ é responsável por garantir que um mapeamento consistente com o intervalo permitido de valores de pixel.

3.6.4.2 Filtragem

Dentre as filtragens utilizadas podemos citar a filtragem passa-baixas que utiliza as bordas nos níveis de uma imagem para a definição do conteúdo de alta frequência, com isso é possível aplicar a suavização e assim alcançar através da atenuação de um intervalo de

específico de componentes de alta frequência na transformada de uma dada imagem (GONZALEZ; WOODS, 2000).

A fórmula abaixo representa um filtro ideal para um filtro passa-baixas bidimensional.

$$H(u, v) = \begin{cases} 1 & \text{se } D(u, v) \leq D_0 \\ 0 & \text{se } D(u, v) > D_0 \end{cases}$$

Onde é aquele que cuja função de transferência satisfaz a relação em que D_0 é um valor não negativo específico, e $D(u, v)$ é a distância do ponto (u, v) à origem do plano da frequência, ou melhor ainda definido pela fórmula abaixo.

$$D(u, v) = \sqrt{u^2 + v^2}$$

Para compreender o processo de borramento em um domínio espacial utilizamos a função $h(x, y)$, onde com a utilização desta função, e de uma imagem de fundo escuro com um ponto branco no meio o filtro passa-baixa bidimensional (FPBI) pode aproximar em dois impulsos o ponto branco, isso, dependendo da clareza desta imagem. Estes conceitos são estendidos para imagens mais complexas considerando cada pixel como um impulso de intensidade proporcional ao nível de cinza daquele pixel (GONZALEZ; WOODS, 2000).

No filtro de passa-baixas também existe o filtro de Butterworth que é uma função de transferência.

$$H(u, v) = \frac{1}{1 + \left[\frac{D(u, v)}{D_0}\right]^{2n}}$$

Esse filtro é de ordem n e com posição da frequência de corte a uma distância D_0 da origem é definida pela relação descrita acima.

Outro tipo de filtragem comum é a passa-altas, que para alcançar a agudização da imagem utiliza os componentes dos níveis de cinza de alta frequência. Isso ocorre porque os componentes de baixa frequência são atenuados sem influenciar nas informações de alta

frequência. O filtro ideal para passa-altas bidimensional (FPAI) deve satisfazer a relação entre 0 e 1, conforme abaixo.

$$H(u, v) = \begin{cases} 1 & \text{se } D(u, v) \leq D_0 \\ 0 & \text{se } D(u, v) > D_0 \end{cases}$$

O filtro Butterworth para a filtragem de passa-altas (FPAB) é definido por uma relação onde a ordem n e o corte posicionado a uma distância D_0 da origem é definida pela relação.

$$H(u, v) = \frac{1}{1 + \left[\frac{D_0}{D(u, v)}\right]^{2n}}$$

O ultimo filtro a ser descrito é chamado de homomórfica, que utilizando a compressão do intervalo de brilho é utilizado para melhorar a aparência de uma imagem. Esse tipo de compressão é realizado com auxílio da função abaixo que identifica os componentes de iluminação e refletância.

$$f(x, y) = i(x, y)r(x, y)$$

Porem essa função não pode ser aplicada de forma direta, pois a transformada de Fourier do produto de duas funções não é separável, por isso para se obter a imagem desejada é formada através do logaritmo da imagem original $f(x, y)$ uma inversão e assim produzindo uma imagem realçada $g(x, y)$.

3.6.4.3 Limiarização

Tendo a limiarização como uma das mais importantes abordagens no que se refere a correções de imagens (GONZALEZ; WOODS, 2000), a limiarização pode ser vista como uma operação que envolve teste de uma função T da forma.

$$T = T[x, y, p(x, y), f(x, y)]$$

Em que $f(x, y)$ é o nível de cinza do ponto (x, y) e $p(x, y)$ denota alguma propriedade local desse ponto, por exemplo, o nível de cinza médio em uma vizinhança centrada em (x, y) . Uma imagem limiarizada $g(x, y)$ é definida como:

$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) > T \\ 0 & \text{se } f(x, y) \leq T \end{cases}$$

O papel da iluminação na limiarização de uma imagem baseada na função $f(x, y)$ pode ser descrita como um produto de um componente de refletância $r(x, y)$ por um componente de iluminação calculado como $i(x, y)$ e com isso analisar os efeitos da iluminação na segmentação da imagem.

Dentre a limiarização as técnicas mais simples é a do particionamento do histograma da imagem por um limiar único T , onde a segmentação é efetuada utilizando-se um sistema que varre a imagem pixel a pixel e se cada um for menor que o limiar único T , então ele pode ser definido se pertence ao fundo ou o objeto da imagem.

Na limiarização ótima podemos supor que uma imagem possui apenas dois tons de brilho, com isso temos uma estimativa da função da densidade de probabilidade do brilho $p(z)$. Para a região clara e para a região escura é feita uma soma ou mistura de duas densidades uni modais e com isso se obtém as áreas proporcionais para cada tipo de brilho da imagem.

Alem das técnicas descritas acima podemos citar a seleção limiar baseada nas características da fronteira, onde uma maneira de melhorar o formato dos histogramas é considerar apenas aqueles pixels que estejam localizados sobre ou próximo das fronteiras entre os objetos e o fundo (GONZALEZ; WOODS, 2000).

3.6.4.4 Transformações geométricas

Sendo mais um dos recursos utilizados para a restauração de imagens, em transformações geométricas encontramos duas maneiras de realizá-las, sendo elas as transformações espaciais e a interpolação entre os níveis de cinza.

Por possuir um conjunto de regras de atuação na imagem de forma a esticar esta imagem, podemos chamá-la de transformações folha de borracha (GONZALEZ; WOODS, 2000).

As regras denominadas de transformações espaciais cuidam do processo contínuo de pixel a pixel e linha a linha até que uma matriz seja obtida, sendo que ela não deve ultrapassá-la. Podemos ver na equação abaixo que

$$\hat{X} = r(x,y)$$

$$\hat{Y} = s(x,y)$$

Representam as transformações espaciais e conforme a Figura 8 é possível se obter a imagem geometricamente distorcida.

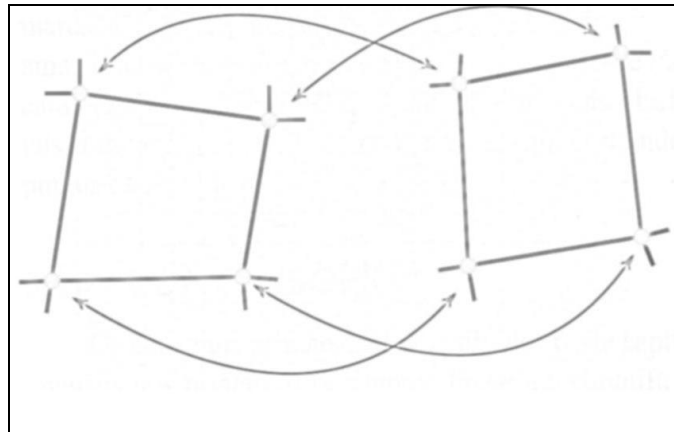


Figura 8. Pontos de amarração.

Fonte: GONZALEZ, R. C.; WOODS, R. E. (2000)

Como na Figura 8 está aparecendo regiões distorcidas, definimos que são vértices quadriláteros que estão correspondentes entre si e com isso podemos utilizar as equações de transformações espaciais para corrigir este problema.

Com a interpolação de níveis de cinza é abordado uma técnica apelidada de “vizinho mais próximo” que tem a finalidade de restaurar uma imagem que ao ser obtida acabou assumindo uma forma arredondada, ou seja, como se estivesse no formato de um barril, assim podemos dizer que, conforme a Figura 9 vemos os mapeamentos das coordenadas inteiras.

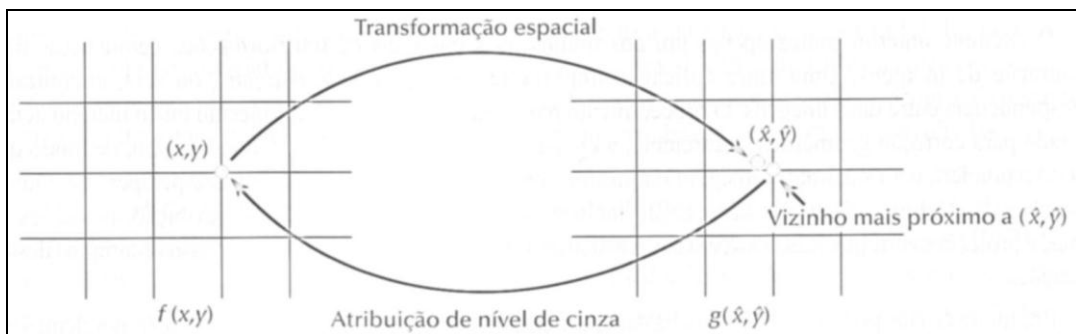


Figura 9. Transformação espacial.

Fonte: GONZALEZ, R. C.; WOODS, R. E. (2000)

Assim podemos dizer que as coordenadas inteiras são representadas por (x, y) e suas coordenadas fracionárias por (\hat{x}, \hat{y}) . A principal desvantagem deste tipo de transformação geométrica é a produção de artefatos indesejáveis, gerando assim distorções nas bordas retas.

3.6.4.5 Operações Morfológicas

Ferramenta baseada na teoria dos conjuntos que é utilizada para se obter determinados componentes de uma imagem, por isso ela é fundamental para o processamento de imagens digitais (GONZALEZ; WOODS, 2000). Entre os conceitos de morfologia voltada para a correção de imagens digitais, podemos definir como importantes a dilatação, erosão, abertura, fechamento e transformada *hit-or-miss*. Na Figura 10 temos um sumário dos resultados morfológicos e suas propriedades básicas, resumindo assim os tipos básicos de elementos.

Operação	Equação	Comentários
Translação	$(A)_x = \{c c = a + x, \text{ para } a \in A\}$	Translada a origem de A para o ponto x .
Reflexão	$\hat{B} = \{x x = -b, \text{ para } b \in B\}$	Reflete todos os elementos de B em torno da origem desse conjunto.
Complemento	$A^c = \{x x \notin A\}$	Conjunto de pontos que não pertencem a A .
Diferença	$A - B = \{x x \in A, x \notin B\} = A \cap B^c$	Conjunto de pontos que pertencem a A mas não a B .
Dilatação	$A \oplus B = \{x (\hat{B})_x \cap A \neq \emptyset\}$	“Expande” a fronteira de A . (I)
Erosão	$A \ominus B = \{x (B)_x \subseteq A\}$	“Contraí” a fronteira de A . (I)
Abertura	$A \circ B = (A \ominus B) \oplus B$	Suaviza contornos, quebra istmos estreitos e elimina pequenas ilhas e picos agudos. (I)
Fechamento	$A \bullet B = (A \oplus B) \ominus B$	Suaviza contornos, funde pequenas quebras e golfos finos e compridos e elimina pequenos buracos. (I)
Transformada “Hit-or-Miss”	$A \odot B = (A \ominus B_1) \cap (A^c \ominus B_2)$ $= (A \ominus B_1) - (A \oplus B_2)$	O conjunto de pontos (coordenadas) nas quais B_1 encontra um “casamento” (“match”) em A (“hit”) e B_2 em A^c .
Extração de fronteiras	$\beta(A) = A - (A \ominus B)$	Conjunto de pontos na fronteira do conjunto A . (I)
Preenchimento de regiões	$X_k = (X_{k-1} \oplus B) \cap A^c; X_0 = p \text{ e } k = 1, 2, 3, \dots$	Preenche uma região em A , dado um ponto p naquela região. (II)
Componentes conectados	$X_k = (X_{k-1} \oplus B) \cap A; X_0 = p \text{ e } k = 1, 2, 3, \dots$	Encontra um componente conexo Y em A , dado um ponto p em Y . (I)
Fecho convexo	$X_k^i = (X_{k-1}^i \oplus B^i) \cup A; i = 1, 2, 3, 4$ $k = 1, 2, 3, \dots, X_0^i = \Lambda, \text{ e } D^i = X_{conv}^i$ $C(A) = \bigcap_{i=1}^4 D^i$	Encontra o fecho convexo $C(A)$ de um conjunto A , sendo que “conv” indica convergência no sentido que $X_k^i = X_{k-1}^i$. (III)
Afinamento	$A \otimes B = A - (A \ominus B)$ $= A \cap (A \ominus B)^c$ $A \otimes \{B\} = ((\dots((A \otimes B^1) \otimes B^2)\dots) \otimes B^n)$ $\{B\} = \{B^1, B^2, B^3, \dots, B^n\}$	Afina um conjunto A . As duas primeiras equações fornecem a definição de afinamento. As duas últimas equações denotam o afinamento por uma seqüência de elementos estruturantes. Esse método é normalmente usado na prática. (IV)
Espessamento	$A \odot B = A \oplus (A \otimes B)$ $A \odot \{B\} = ((\dots((A \odot B^1) \odot B^2)\dots) \odot B^n)$	Faz um espessamento do conjunto A . (veja os comentários precedentes sobre a seqüência de elementos estruturantes). Usa (IV) com 0's e 1's invertidos.
Esqueletos	$S(A) = \bigcup_{k=0}^K S_k(A)$ $S_k(A) = \bigcup_{k=0}^K \{(A \ominus kB) - [(A \ominus kB) \odot B]\}$ $A = \bigcup_{k=0}^K (S_k(A) \oplus kB)$	Encontra o esqueleto $S(A)$ do conjunto A . A última equação indica que A pode ser reconstruído a partir dos subconjuntos de esqueleto $S_k(A)$. Em todas as três equações, K é o valor do passo iterativo depois do qual A é erodido a um conjunto vazio. A notação $(A \ominus kB)$ denota a k -ésima iteração de erosão sucessiva. (I)

Figura 10. Sumário dos resultados morfológicos
Fonte: GONZALEZ, R. C.; WOODS, R. E. (2000)

Nesta figura encontramos a descrição de todas as operações básicas, onde, podemos analisar as equações responsáveis pela obtenção dos resultados desejados, qual a função que cada uma destas operações e a sua influência na correção das imagens digitais.

3.6.5 Métricas

As métricas podem ser definidas para medir as distâncias entre os pontos de uns conjuntos, ou melhor, com as métricas são possíveis realizar em comparações entre as transcrições de texto, definindo assim se as alterações realizada são ou não viáveis.

As métricas mais conhecidas para que seja possível comparar os resultados são chamadas de Junker Hoch Dengel e Distância de Levenshtein segundo ALVES (2003).

3.6.5.1 Métrica de Junker-Hoch-Dengel

Para analisar a eficiência do reconhecimento de caracteres a métrica de Junker-Hoch-Dengel utiliza cálculos baseados em *recall*, precisão e exatidão, segundo (JUNKER; HOCH; DENGEL, 1999), esses cálculos são baseados nas fórmulas:

$$\text{Recall} = \frac{\#(\text{character/word x correctly recognised})}{\#(\text{character/word x occurs})}$$

$$\text{Precisão} = \frac{\#(\text{character/word x correctly recognised})}{\#(\text{character/word x recognised})}$$

$$\text{Exatidão} = \frac{\#(\text{character/word x correctly recognised})}{\#(\text{all characters/words})}$$

Por serem métricas que possuem maiores probabilidades de encontrar prováveis caracteres desconhecidos, existe uma garantia de que iremos obter um maior índice de acerto,

conseguindo assim resultados cuja flutuação em torno de um valor médio e sua discrepância em relação ao valor verdadeiro será muito pequena.

3.6.5.2 Distância de Levenshtein

A Distância de Levenshtein é o menor número de caracteres que devem ser inseridos, apagados ou substituídos na comparação entre duas palavras.

Para demonstrar melhor vamos utilizar uma palavra originária da língua Tupi Guarani chamada de Paranambouc e compará-la com a sua tradução na língua portuguesa, conhecida como Pernambuco (ALVES, 2003), com essas duas palavras podemos realizar as seguintes alterações para obter a distância de levenshtein:

- a) Paranambouc: Palavra Original
- b) Parnambouc: 1 exclusão “a”
- c) Pernambouc: 1 substituição “a” por “e”
- d) Pernambuc: 1 exclusão “o”
- e) Pernambuco: 1 inserção “o”

Com os dados demonstrados acima podemos dizer que com duas exclusões, uma troca e uma inserção a distância de Livenshtein é de quatro e com isso é possível notar a formação de uma métrica, pois:

$$\mathcal{L}(A, A) = 0, \mathcal{L}(A, B) = \mathcal{L}(B, A)$$

E para a inequação triangular é utilizada a métrica:

$$\mathcal{L}(A, C) < \mathcal{L}(A, B) + \mathcal{L}(B, C)$$

Onde A, B e C são caracteres e L (X, Z) são a Distância de Levenshtein entre os caracteres X e Z segundo (ALVES, 2003).

4 TRABALHOS CORRELATOS

Este capítulo tem como característica demonstrar outros trabalhos relacionados ao trabalho apresentado neste documento e com isso melhorar a avaliação do mesmo.

O primeiro é um trabalho de conclusão de curso apresentado ao departamento de ciência da computação da Universidade do Extremo Sul Catarinense pela acadêmica Tatiany Demetrio orientada pelo professor Evânio Ramos Nicoleit e com o título “Protótipo para Visualização, Processamento e Análise de Imagens Médicas”.

Este trabalho de conclusão de curso descreve estudo e desenvolvimento de um Protótipo para visualização, processamento e análise de imagens médicas digitais. São abordadas e implementadas técnicas que permitem a obtenção de imagens convenientes de auxílio ao diagnóstico de patologias.

O segundo trabalho de conclusão de curso apresentado ao departamento de ciência da computação da Universidade do Extremo Sul Catarinense pela acadêmica Vanessa Camerini Parise orientada pelo professor Evânio Ramos Nicoleit e Co-orientada pelo professor Márcio Arcângelo Zaccaron. Possui o seguinte título “Sistema de Auxílio à Identificação e Classificação de micro calcificações em Mamogramas Digitais via Processamento Digital de imagens e Redes Neurais”.

Neste trabalho foi desenvolvido um Sistema para Auxílio à Identificação e Classificação de Microcalcificações em Mamografias Digitais via Processamento Digital de Imagens e Redes Neurais, denominado *BreastSystem*. Este sistema visa auxiliar mastologistas, radiologistas, imaginologistas e ginecologistas, no diagnóstico de mamografias digitais por meio do processamento das imagens e o uso das redes neurais para classificação das microcalcificações mamárias encontradas.

O terceiro trabalho correlato trata-se de um artigo descrito para a revista Odontologia da faculdade PUCRS pelos acadêmicos André Wilson Machado, Bernardo Quiroga Souki, Ênio Tonani Mazzeiro e que possui o título “Avaliação de Quatro Métodos de visualização de Imagens Digitais em Odontologia”.

O objetivo deste trabalho foi comparar a qualidade de quatro métodos de visualização de imagens em Odontologia. Foram selecionados quatro slides 35 mm de pacientes da Faculdade de Odontologia da Pontifícia Universidade Católica de Minas Gerais. Os diapositivos foram digitalizados por quatro recursos e, posteriormente, as imagens foram processadas para serem visualizadas por quatro métodos distintos.

O quarto trabalho correlato é uma dissertação de mestrado da faculdade federal de Pernambuco, escrita por Neide Ferreira Alves e orientada pelo professor Rafael Dueire Lins com o título de “Estratégias para Melhoria do Desempenho de Ferramentas Comerciais de Reconhecimento Óptico de Caracteres”.

Neste trabalho é apresentada uma nova métrica para avaliar transcrições de OCR's: através da aplicação de técnicas de filtragem (brilho, contraste, resolução, rotação, etc.) na imagem original, para que as mudanças mínimas gerem inúmeras imagens, as quais serão submetidas ao OCR e resultarão em textos distintos.

5 PRÉ-PROCESSAMENTO DE IMAGENS PARA A LEITURA OCR

Este capítulo serve para demonstrar quais as imagens, algoritmos, procedimentos, softwares e métricas foram utilizados para que fosse possível demonstrar claramente os resultados obtidos.

5.1 METODOLOGIA

Foi necessário realizar a revisão bibliográfica para obter um bom nível de referências, sendo que essa revisão bibliográfica será baseada em artigos, TCCs, teses de mestrado entre outras fontes.

Foi necessário realizar a identificação de imagens de referência para termos condições de aplicar os algoritmos de pré-processamento de imagens.

Estabelecido uma métrica para a comparação de reconhecimento de caracteres, para que com isso o embasamento do TCC seja o mais fiel possível.

Implementado os algoritmos de pré-processamento, avaliando seus respectivos parâmetros de utilização bem como sintonia associada às combinações de algoritmos no processo de OCR.

Realizados os testes e a validação do sistema, para que assim aplicação implementada poderá ser avaliada de forma coerente.

5.2 DESENVOLVIMENTO DO SISTEMA

Nesta fase serão apresentados os recursos utilizados, o desenvolvimento, avaliação e os resultados obtidos através dos algoritmos, métricas e métodos utilizados para o pré-processamento de imagens para a leitura OCR.

5.2.1 Imagens De Referência

Para a aplicação dos algoritmos utilizados para a execução das correções e para obtermos os resultados da métrica de avaliação, adquirimos algumas imagens por meio da digitalização de alguns documentos utilizando um scanner e utilizando imagens de texto conhecidas. Na Figura 11 temos a digitalização de uma página de um livro onde temos problemas com inclinação e ruídos e iremos chamá-lo de referencia 1.

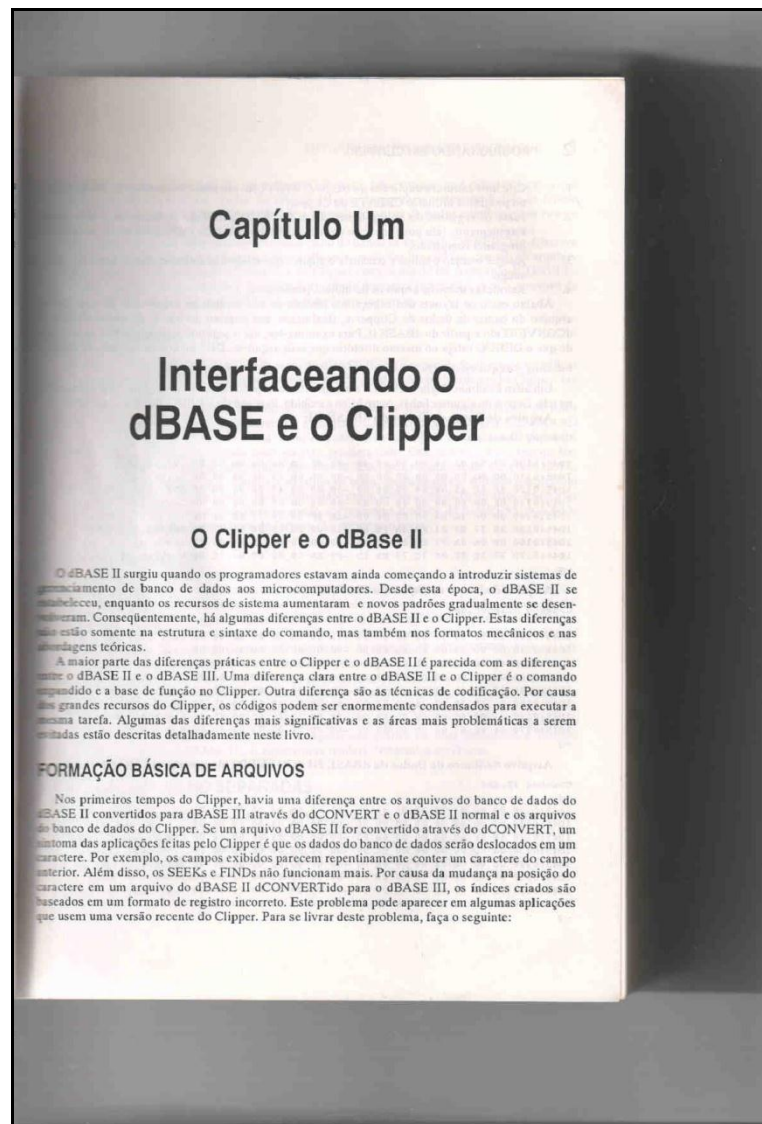


Figura 11. Imagem de referencia 1
Fonte: STRALEY, S. J. (1999)

Com a Figura 11 podemos verificar que o texto esta desalinhado e que possui vários ruídos gerados pela digitalização do livro. Na Figura 12 vemos uma imagem descrita como referencia 2 utilizada para efetuar testes de eficiência OCR.

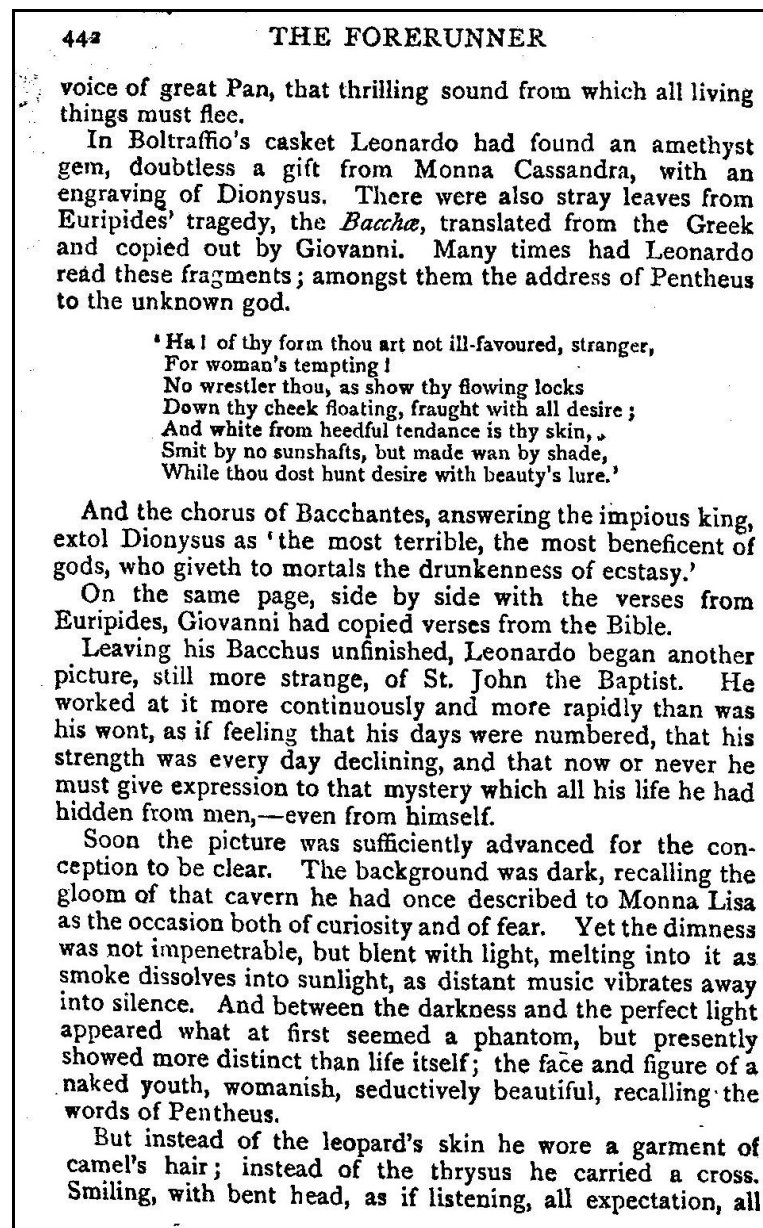


Figura 12. Imagem de referencia 2
Fonte: BLUELEAF (2011)

No erro de digitalização da Figura 12, temos um texto com seus caracteres fora do seu alinhamento e com alguns ruídos, com esses erros o software OCR encontrará maiores

dificuldades para identificar os caracteres contidos no texto. Na Figura 13 temos uma imagem retirada de um livro denominada por referencia 3.

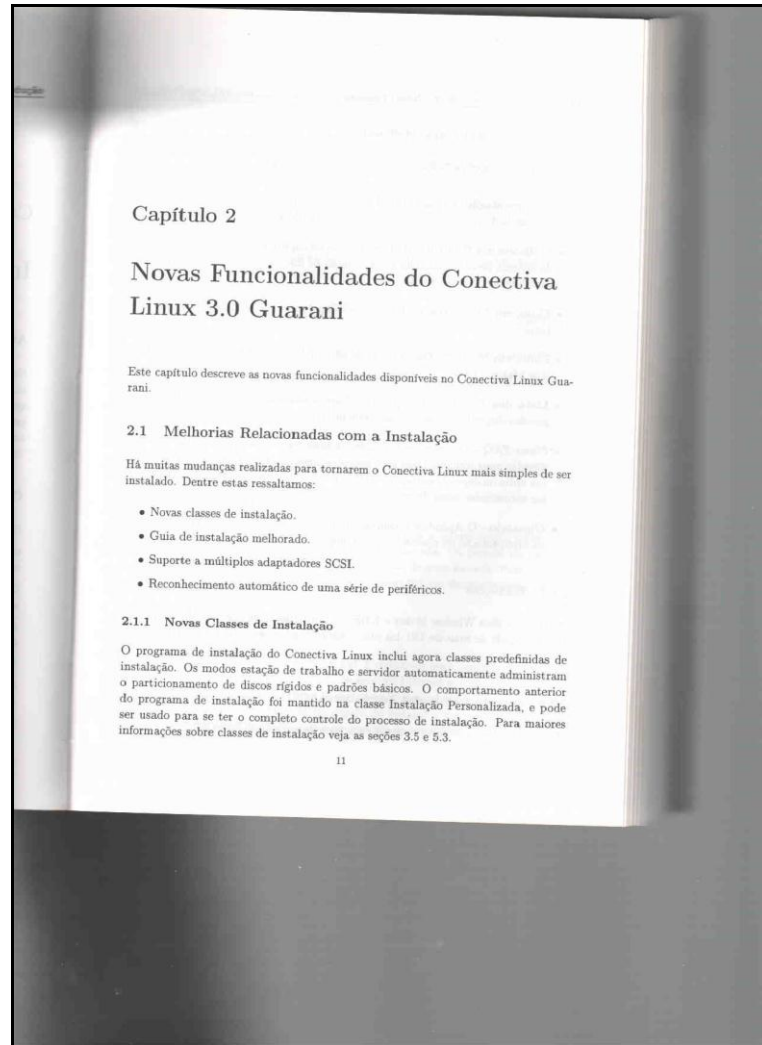


Figura 13. Imagem da referencia 3
Fonte: CONECTIVA (1999)

Como o software OCR utiliza padrões para reconhecer os caracteres de uma imagem, os erros gerados pelo digitalizador na Figura 13, podemos afirmar que esses padrões ficaram muito complicados de serem identificados pelo software OCR. A imagem digitalizada da Figura 14 com nome de referencia 4 possui um resultado mais complicado da digitalização, pois nela são encontrados vários erros.

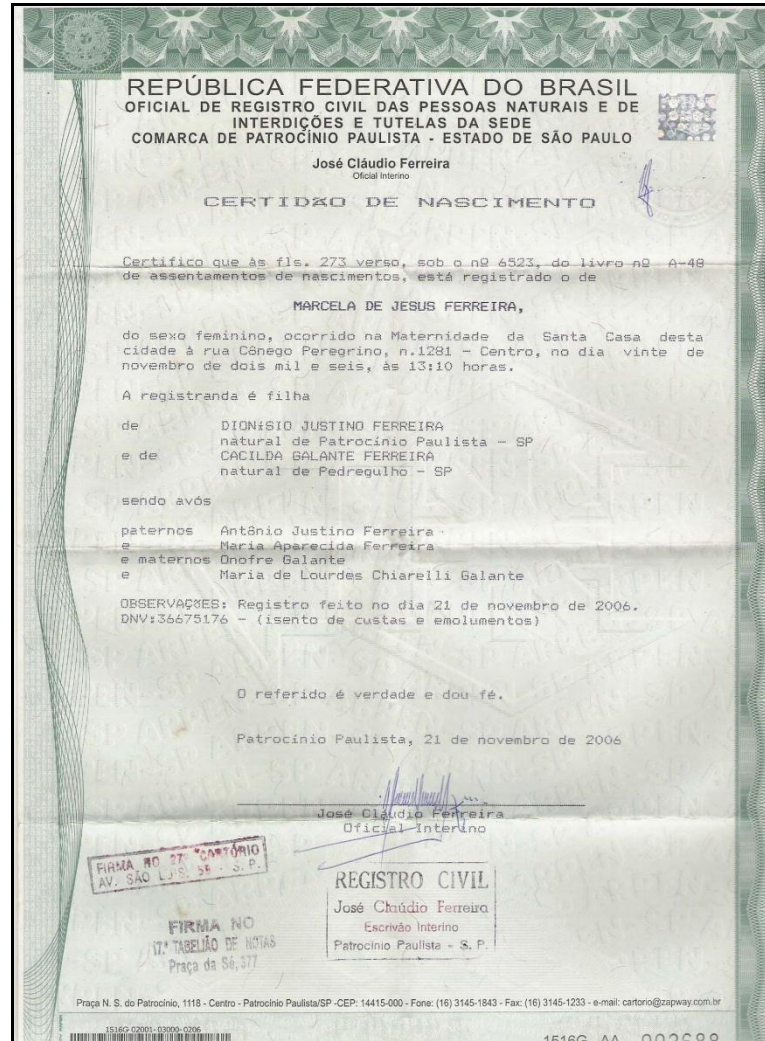


Figura 14. Imagem de referencia 4

Na Figura 14 temos um texto com vários tipos de ruídos, como dobras e sombras geradas pelas condições físicas do documento digitalizado.

5.2.2 Desenvolvimento do Aplicativo Para o Pré-processamento de imagens

Na Figura 15 apresenta-se a interface do aplicativo, onde com a utilização dos algoritmos abordados, a aplicação foi desenvolvida em JAVA utilizando o software livre Netbeans 6.9.1 e as suas bibliotecas ImageIO, Image, Color e Graphics2D, para assim obtermos os resultados ao aplicarmos a métrica de análise de eficiência OCR.

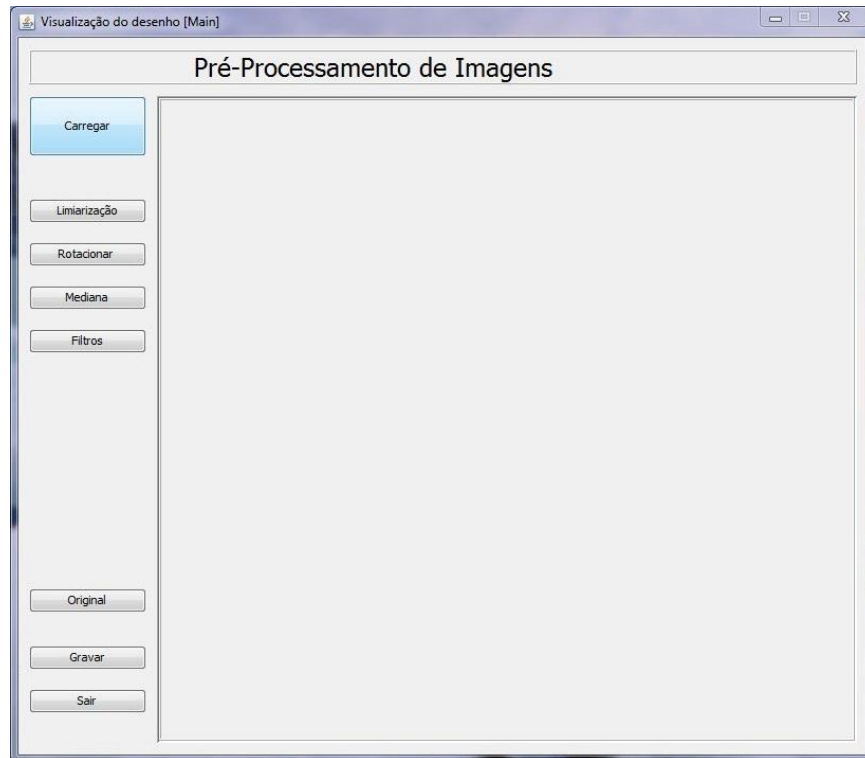


Figura 15. Interface do programa desenvolvido

Na interface é possível identificar o funcionamento do aplicativo, onde, primeiramente será carregada a imagem digitalizada e após isso selecionar o tipo de pré-processamento desejado e assim obter o resultado, que deverá ser salvo, para após isso ser possível dar continuidade ao processo de leitura OCR.

5.2.3 Algoritmos para o Pré-processamento de Imagens

Para obtermos um melhor resultado do programa OCR ao ser aplicarmos as imagens de referência encontradas, serão utilizados os algoritmos de pré-processamento, onde cada um vai executar uma função específica para minimizar os problemas encontrados.

5.2.3.1 Algoritmo Para a Filtragem

No protótipo desenvolvido foi disponibilizado os filtros passa-alta, passa-baixa, Gaussiano, Prewitt, Kirsch, Robinson e mediana sendo estes filtros considerados muito importantes para o pré-processamento de imagens (GONZALES; WOODS, 2000). Nas Figuras 16a e 16b temos os comprimentos 3X3 e 5X5 da matriz da filtragem passa-alta.

$$\begin{pmatrix} -1f/9, & -1f/9, & -1f/9, \\ -1f/9, & 8f/9, & -1f/9, \\ -1f/9, & -1f/9, & -1f/9 \end{pmatrix};$$

Figura 16a. Filtragem passa-alta 3X3
Fonte: EFFORD, N. (2000)

$$\begin{pmatrix} -1f/4, & 1f/4, & -1f/4, & 1f/4, & -1f/4, \\ 1f/4, & -2f/4, & 2f/4, & -2f/4, & 1f/4, \\ -1f/4, & 2f/4, & 4f/4, & 2f/4, & -1f/4, \\ 1f/4, & -2f/4, & 2f/4, & -2f/4, & 1f/4, \\ -1f/4, & 1f/4, & -1f/4, & 1f/4, & -1f/4 \end{pmatrix};$$

Figura 16b. Filtragem passa-alta 5X5
Fonte: EFFORD, N. (2000)

Conforme a Figura 16a e a Figura 16b a filtragem passa-alta tende a acentuar melhor os níveis de cinza de alta frequência, ignorando os níveis de cinza de baixa frequência. Na Figuras 17a, 17b e 17c vemos os comprimentos 3X3, 5X5 e 7X7 da matriz da filtragem passa-baixa.

$$\begin{pmatrix} 1f/9, & 1f/9, & 1f/9, \\ 1f/9, & 1f/9, & 1f/9, \\ 1f/9, & 1f/9, & 1f/9 \end{pmatrix};$$

Figura 17a. Filtragem passa-baixa 3X3
Fonte: EFFORD, N. (2000)

$$\begin{pmatrix} 1f/25, & 1f/25, & 1f/25, & 1f/25, & 1f/25, \\ 1f/25, & 1f/25, & 1f/25, & 1f/25, & 1f/25, \\ 1f/25, & 1f/25, & 1f/25, & 1f/25, & 1f/25, \\ 1f/25, & 1f/25, & 1f/25, & 1f/25, & 1f/25, \\ 1f/25, & 1f/25, & 1f/25, & 1f/25, & 1f/25 \end{pmatrix};$$

Figura 17b. Filtragem passa-baixa 5X5
Fonte: EFFORD, N. (2000)

$$\begin{pmatrix} 1f/49, & 1f/49, & 1f/49, & 1f/49, & 1f/49, & 1f/49, & 1f/49, \\ 1f/49, & 1f/49, & 1f/49, & 1f/49, & 1f/49, & 1f/49, & 1f/49, \\ 1f/49, & 1f/49, & 1f/49, & 1f/49, & 1f/49, & 1f/49, & 1f/49, \\ 1f/49, & 1f/49, & 1f/49, & 1f/49, & 1f/49, & 1f/49, & 1f/49, \\ 1f/49, & 1f/49, & 1f/49, & 1f/49, & 1f/49, & 1f/49, & 1f/49, \\ 1f/49, & 1f/49, & 1f/49, & 1f/49, & 1f/49, & 1f/49, & 1f/49, \\ 1f/49, & 1f/49, & 1f/49, & 1f/49, & 1f/49, & 1f/49, & 1f/49 \end{pmatrix};$$

Figura 17c. Filtragem passa-baixa 7X7
Fonte: EFFORD, N. (2000)

A tendência da filtragem passa-baixa é a de minimizar os tons de cinza de baixa frequência, dando assim mais enfoque aos tons de cinza de alta frequência. Na Figura 18 temos a aplicação da filtragem mediana.

```

BufferedImage imagem = ImageIO.read(new File("C:\\Us
Raster raster = imagem.getRaster();
WritableRaster wraster = imagem.getRaster();
double valornr, valornrg, valornb;
int[] v = new int[9];
for(int y=1; y<imagem.getHeight()-1; y++){
    for(int x=1; x<imagem.getWidth()-1; x++){
        //Aplica Filtro de Mediana 3x3
        LeJanela3x3(raster,v,x,y,0);
        valornr = CalcMediana(9,v);
        LeJanela3x3(raster,v,x,y,1);
        valornrg = CalcMediana(9,v);
        LeJanela3x3(raster,v,x,y,2);
        valornb = CalcMediana(9,v);
        wraster.setSample(x,y,0, (int) (valornr+.5));
        wraster.setSample(x,y,1, (int) (valornrg+.5));
        wraster.setSample(x,y,2, (int) (valornb+.5));
    }
}
ImageIO.write(imagem, "jpg", new File("C:\\Users\\va

```

Figura 18. Filtragem mediana
Fonte: EFFORD, N. (2000)

Com a utilização da filtragem mediana é possível reduzir a quantidade de ruídos, na Figura 18 vemos que o algoritmo busca os valores de RGB pixel a pixel, soma e divide o resultado da soma por três e assim obtêm o valor da media referente ao pixel. Na imagem 19 temos as quatro máscaras utilizadas no filtro Gaussiano.

1f, 5f, 6f, 5f, 1f,	1f, 5f, 8f, 5f, 1f,	1f, 5f, 10f, 5f, 1f,	1f, 5f, 12f, 5f, 1f,
5f, 25f, 30f, 25f, 5f,	5f, 25f, 40f, 25f, 5f,	5f, 25f, 50f, 25f, 5f,	5f, 25f, 60f, 25f, 5f,
6f, 30f, 36f, 30f, 6f,	8f, 40f, 64f, 40f, 8f,	10f, 50f, 100f, 50f, 10f,	12f, 60f, 144f, 60f, 12f,
5f, 25f, 30f, 25f, 5f,	5f, 25f, 40f, 25f, 5f,	5f, 25f, 50f, 25f, 5f,	5f, 25f, 60f, 25f, 5f,
1f, 5f, 6f, 5f, 1f));	1f, 5f, 8f, 5f, 1f));	1f, 5f, 10f, 5f, 1f));	1f, 5f, 12f, 5f, 1f));

Figura 19. Filtragem Gaussiano
Fonte: EFFORD, N. (2000)

Trata-se de um filtro que trabalha com baixas frequências e assim denominada como sendo do tipo de filtro passa-baixa, onde a aplicação do algoritmo é feita sucessivas vezes seguindo sempre o comprimento da matriz informada na Figura 19 tendendo assim a forma de uma curva. Para demonstrar o algoritmo do filtro Prewitt a Figura 20 é composta pelos comprimentos das oito matrizes que compõe o filtro.

1f, 1f, 1f, 1f, -2f, 1f, -1f, -1f, -1f));	1f, 1f, 1f, 1f, -2f, -1f, 1f, -1f, -1f));	1f, 1f, -1f, 1f, -2f, -1f, 1f, 1f, -1f));	1f, -1f, -1f, 1f, -2f, -1f, 1f, 1f, 1f));
-1f, -1f, -1f, 1f, -2f, 1f, 1f, 1f, 1f));	-1f, -1f, 1f, -1f, -2f, 1f, 1f, 1f, 1f));	-1f, 1f, 1f, -1f, -2f, 1f, -1f, 1f, 1f));	1f, -1f, 1f, -1f, -2f, 1f, -1f, -1f, 1f));

Figura 20. Filtragem Prewitt
Fonte: EFFORD, N. (2000)

Com os comprimentos das matrizes aplicadas, sendo que cada uma delas trata de uma posição da imagem definidas como direcional região norte, noroeste, oeste, sudoeste, sul, sudeste, leste e nordeste a imagem tende ao realce de altas frequências (Demetrio 2003). Na Figura 21 temos os comprimentos das matrizes do filtro Kirsch.

5f, 5f, 5f, -3f, 0f, -3f, -3f, -3f, -3f));	5f, 5f, -3f, 5f, 0f, -3f, -3f, -3f, -3f));	5f, -3f, -3f, 5f, 0f, -3f, 5f, -3f, -3f));	-3f, -3f, -3f, 5f, 0f, -3f, 5f, 5f, -3f));
-3f, -3f, -3f, -3f, 0f, -3f, 5f, 5f, 5f));	-3f, -3f, -3f, -3f, 0f, 5f, -3f, 5f, 5f));	-3f, -3f, 5f, -3f, 0f, 5f, -3f, -3f, 5f));	-3f, 5f, 5f, -3f, 0f, 5f, -3f, -3f, -3f));

Figura 21. Filtragem Kirsch
Fonte: EFFORD, N. (2000)

O filtro Kirsh assim como o filtro Prewitt trata cada uma das posições da imagem, podendo com esta estrutura definir qual região será tratada e com isso obter vários resultados. Na Figura 22a temos as matrizes referentes ao filtro Robinson de três níveis aplicada ao

algoritmo enquanto a Figura 22b possui as matrizes referentes ao filtro de Robinson de cinco níveis.

1f, 1f, 1f, 0f, 0f, 0f, -1f, -1f, -1f));	1f, 1f, 0f, 1f, 0f, -1f, 0f, -1f, -1f));	1f, 0f, -1f, 1f, 0f, -1f, 1f, 0f, -1f));	0f, -1f, -1f, 1f, 0f, -1f, 1f, 1f, 0f));
-1f, -1f, -1f, 0f, 0f, 0f, 1f, 1f, 1f));	-1f, -1f, 0f, -1f, 0f, 1f, 0f, 1f, 1f));	-1f, 0f, 1f, -1f, 0f, 1f, -1f, 0f, 1f));	0f, 1f, 1f, -1f, 0f, 1f, -1f, -1f, 0f));

Figura 22a. Filtragem Robinson com três níveis
Fonte: EFFORD, N. (2000)

1f, 2f, 1f, 0f, 0f, 0f, -1f, -2f, -1f));	2f, 1f, 0f, 1f, 0f, -1f, 0f, -1f, -2f));	1f, 0f, -1f, 2f, 0f, -2f, 1f, 0f, -1f));	0f, -1f, -2f, 1f, 0f, -1f, 2f, 1f, 0f));
-1f, -2f, -1f, 0f, 0f, 0f, 1f, 2f, 1f));	-2f, -1f, 0f, -1f, 0f, 1f, 0f, 1f, 2f));	-1f, 0f, 1f, -2f, 0f, 2f, -1f, 0f, 1f));	0f, 1f, 2f, -1f, 0f, 1f, -2f, -1f, 0f));

Figura 22b. Filtragem Robinson com cinco níveis
Fonte: EFFORD, N. (2000)

5.2.3.2 Transformações Geométricas - Rotação

Por meio da rotação, pode-se alinhar a imagem digitalizada. Na Figura 23 apresenta-se o pseudocódigo da implementação em Java para a rotação da imagem.

```
BufferedImage imagem = ImageIO.read(new File("C:\\Users\\vando\\Desktop\\teste.jpg"));
BufferedImage imagemAlterada = new BufferedImage(imagem.getHeight(), imagem.getWidth(), imagem.getType());
Graphics2D surface = imagemAlterada.createGraphics();
AffineTransform at = new AffineTransform();
at.scale(0.5, 0.5);
at.rotate(1.0 * Math.PI / 180.0, imagemAlterada.getWidth() / 2.0, imagemAlterada.getHeight() / 2.0);
surface.transform(at);
surface.drawImage(imagem, 0, 0, null);
surface.dispose();
ImageIO.write(imagemAlterada, "jpg", new File("C:\\Users\\vando\\Desktop\\teste.jpg"));
```

Figura 23. Rotação de imagem
Fonte: EFFORD, N. (2000)

O grande problema para a rotação de uma imagem de texto digitalizada é definir o ângulo correto, para assim obter o melhor desempenho, Na Figura 23 vemos que a função *rotate* da imagem esta definida em 1.0 porem ela deverá ser alterada pelo usuário.

5.2.3.3 Algoritmo de limiarização

Utilizado para determinar o nível de cinza da imagem, tornando-a assim mais nítida para a leitura OCR. Na Figura 24 temos o algoritmo em Java da limiarização para ser aplicada em imagens de texto digitalizadas.

```
BufferedImage imagem = ImageIO.read(new File(nomeArquivoOriginal));
int width = imagem.getWidth();
int height = imagem.getHeight();
for (int i = 0; i < width; i++) {
    for (int j = 0; j < height; j++) {
        int rgb = imagem.getRGB(i, j);
        int r = (int) ((rgb & 0x00FF0000) >>> 16);
        int g = (int) ((rgb & 0x0000FF00) >>> 8);
        int b = (int) (rgb & 0x000000FF);
        int media = (r + g + b) / 3;
        Color branco = new Color(255, 255, 255);
        Color preto = new Color(0, 0, 0);
        if (media > limiar) {
            imagem.setRGB(i, j, branco.getRGB());
        } else {
            imagem.setRGB(i, j, preto.getRGB());
        }
    }
}
ImageIO.write(imagem, "jpg", new File("C:\\Users\\vando\\Desktop\\teste.jpg"));
```

Figura 24. Limiarização aplicada a Java
Fonte: EFFORD, N. (2000)

A limiarização é definida a partir de um ponto chamado de limiar, onde, os valores RGB dos pixels são comparados ao valor limiar e se caso o valor limiar for maior que o valor RGB do pixel o algoritmo insere um pixel preto, caso contrário ele insere um pixel branco.

5.2.4 Resultados Obtidos

Para demonstrar a eficiência dos resultados a serem obtidos através dos algoritmos de pré-processamento de imagens, iremos utilizar a métrica conhecida como distância de Levenshtein que é responsável por medir o índice de acerto e a quantidade de caracteres que devem ser inseridos e ou apagados para se obter a palavra correta. Na Figura 25 temos a digitalização de uma pagina de um livro a partir da utilização de um scanner que será tratada com o nome de referencia 1.

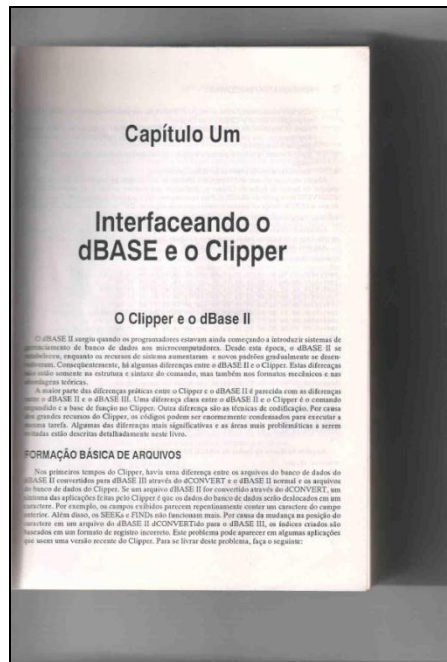


Figura 25. Imagem referencia 1
Fonte: STEPHEN, J. S. (1992)

Na Figura 25 podemos identificar claramente os problemas encontrados, como os efeitos de sombra causados pelo estado físico em que se encontra o livro. Na Figura 26 temos o resultado da imagem referencia 1 após ser aplicado o algoritmo de passa-alta.

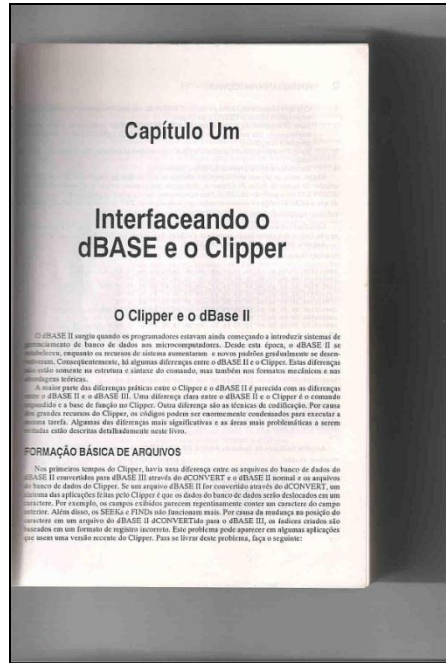


Figura 26. Imagem de referencia 1 corrigida

Para o programa de leitura OCR obter um melhor desempenho, foi aplicado a esta imagem a filtragem passa-alta, definindo melhor os seus pixels. Na Figura 27 temos a digitalização de uma pagina de outro livro a partir da utilização de um scanner que será tratada com o nome de referencia 2.

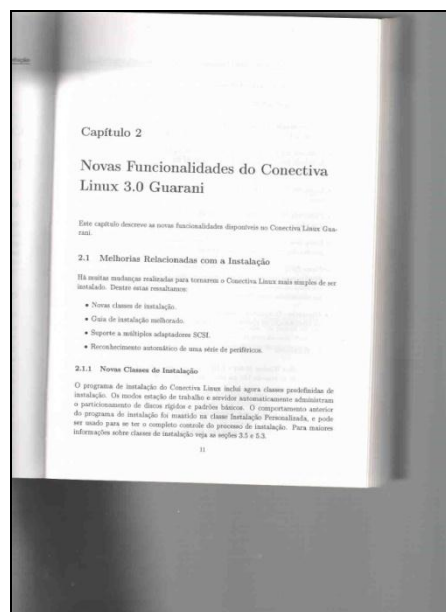


Figura 27. Imagem de referencia 2
Fonte: CONECTIVA (1999)

Assim como na Figura 25 a Figura 27 possui praticamente os mesmos problemas, como os efeitos de sombra e dobra de página. Na Figura 28 temos o resultado da imagem referencia 2 após ser aplicado o mesmo algoritmo de passa-alta aplicado a Figura 25.

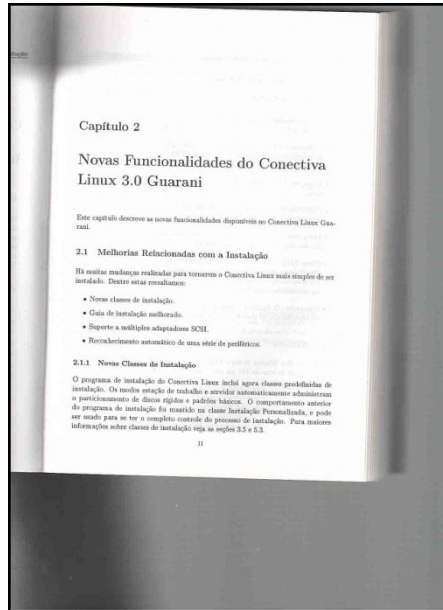


Figura 28. Imagem de referencia 2 corrigida

Com a imagem corrigida utilizando o filtro de passa-alta, os pixels tornam-se mais definidos e assim é possível obter os resultados da leitura OCR mais positivos. Na Figura 29 temos outra imagem digitalizada chamada de referencia três.

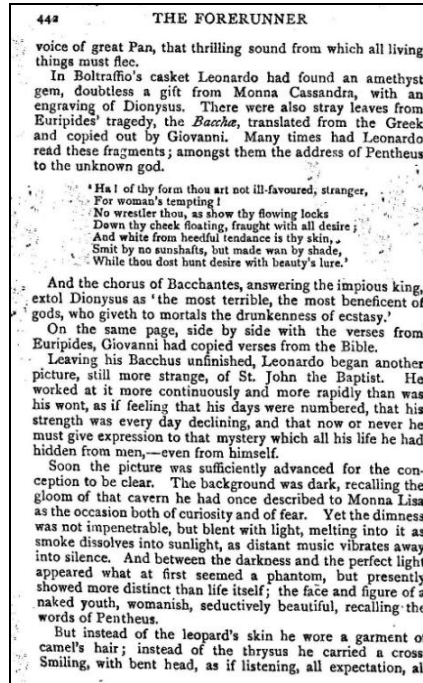


Figura 29. Imagem de referencia 3
Fonte: BLUELEAF (2011)

Na Figura 29 podemos ver que a imagem digitalizada possui diversos ruídos, dificultando a leitura OCR. Na Figura 30 temos a imagem referencia 3 após ter sido submetida aos algoritmos de pré-processamento de imagem.

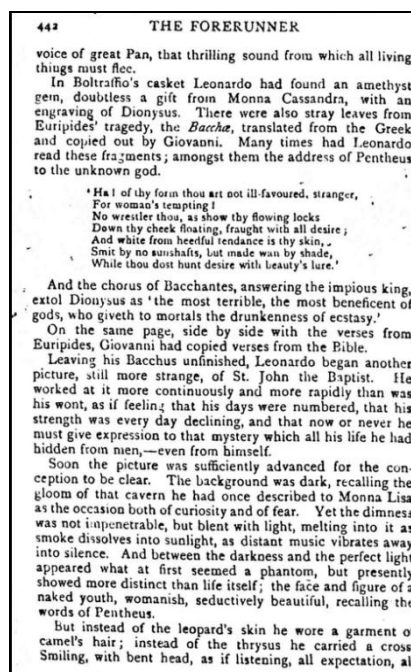


Figura 30. Imagem de referencia 3 corrigida

Na Figura 30 temos o resultado da aplicação do algoritmo do filtro da mediana, onde a mesma busca os pixels em comum e elimina os ruídos. Na Figura 31 chamada de referencia 4, temos o exemplo de diferentes ruídos em uma única imagem digitalizada.

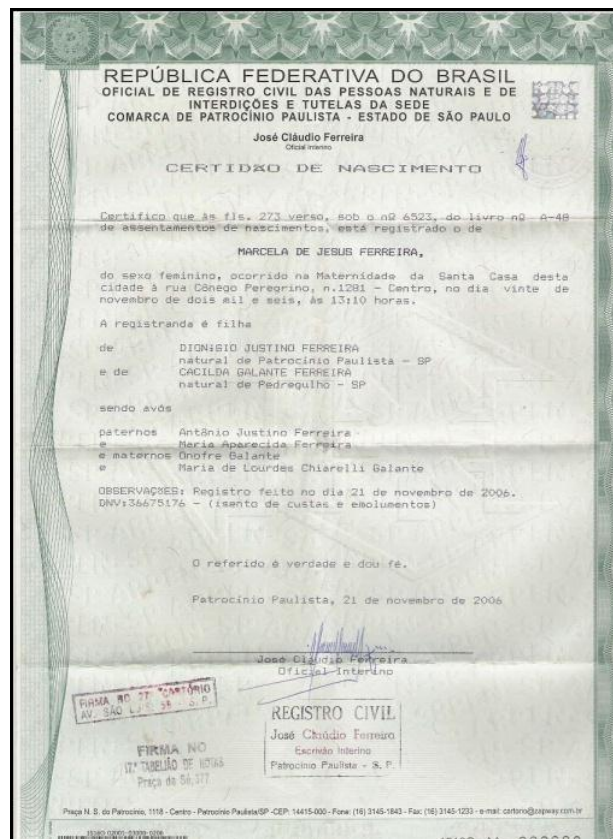


Figura 31. Imagem de referencia 4

Para a correção de alguns problemas encontrados na digitalização de imagens, devemos aplicar vários algoritmos para obter-se o melhor resultado. Na Figura 32 temos o resultado da aplicação de vários algoritmos de pré-processamento de imagem.

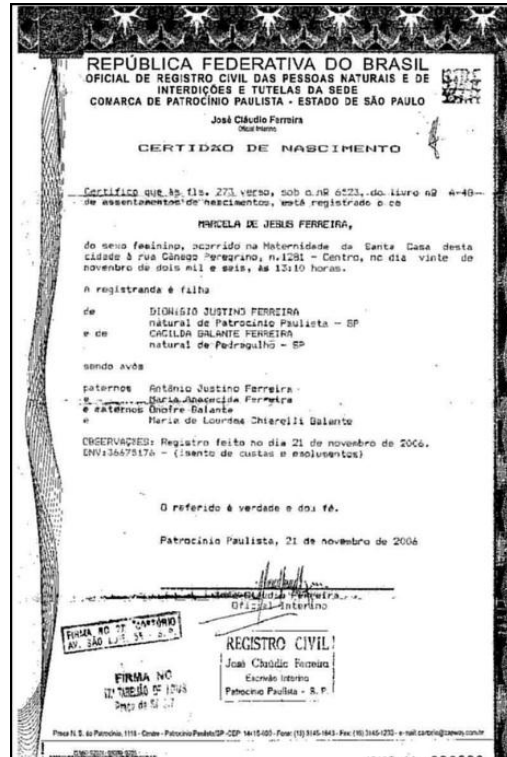


Figura 32. Imagem de referencia quatro corrigida

Para a correção dos diversos tipos de ruídos encontrados na Figura 32, foi utilizando primeiramente o algoritmo de limiarização com tamanho do limiar de 180, para nivelar melhor os tons de cinza, depois o algoritmo da filtragem passa-baixa para suavizar a imagem e por ultimo foi aplicado por duas vezes seguidas o algoritmo da filtragem passa-alta para definir melhor as bordas dos caracteres presentes na imagem.

5.2.4.1 Resultados Obtidos das Métricas da Distância de Levenshtein

Aplicando o algoritmo da distância de Levenshtein para medir o tamanho da diferença entre o texto original e os textos obtidos com a utilização de um programa de leitura OCR sem a aplicação de nenhum pré-processamento de imagem, obtemos os seguintes resultados:

- a) imagem de referencia um: 340 exclusões e ou inclusões e um índice de acerto de 82,49%;

- b) imagem de referencia dois: 395 exclusões e ou inclusões e um índice de acerto de 82,05%;
- c) imagem de referencia três: 170 exclusões e ou inclusões e um índice de acerto de 82,33%;
- d) imagem de referencia numero quatro: 914 exclusões e ou inclusões e um índice de acerto de 27,52%;

O próximo passo é aplicar a métrica da distância de Levenshtein para comparar o texto original e o texto obtido do programa de leitura OCR obtidos com a ajuda do pré-processamento de imagens:

- a) imagem de referencia um: 252 exclusões e ou inclusões e um índice de acerto de 87,02%;
- b) imagem de referencia dois: 199 exclusões e ou inclusões e um índice de acerto de 90,96%;
- c) imagem de referencia três: 127 exclusões e ou inclusões e um índice de acerto de 86,80%;
- d) imagem de referencia quatro: 752 exclusões e ou inclusões e um índice de acerto de 40,36%;

Com os resultados obtidos acima, montamos a Tabela 1 e assim podemos avaliar os resultados obtidos e determinar se houve ou não benefícios com a utilização do pré-processamento de imagens, lembrando que foram avaliadas as quatro imagens de referência sem o pré-processamento de imagem e as mesmas imagens submetidas aos algoritmos de pré-processamento de imagem, onde cada imagem foi submetida a métrica da distância de Levenshtein.

Tabela 1 - Aplicação da métrica da Distância de Levenshtein

Imagem	Sem Processamento	Com Processamento	% Eficiência
imagem de referencia um	340 – 82,49%	252 – 87,02%	+ 4,53 %
imagem de referencia dois	395 – 82,05%	199 – 90,96%	+ 8,91%
imagem de referencia três	170 – 82,33%	127 – 86,80%	+ 4,47%
imagem de referencia quatro	914 – 27,52%	752 – 40,36%	+ 12,84%

Na Tabela 1 temos os resultados, onde, comparando os resultados da utilização de imagens sem o pré-processamento com as imagens com o pré-processamento foi possível determinar que houve um ganho na porcentagem de eficiência gerada pela distância de Levenshtein na aplicação dos algoritmos de pré-processamento de imagem.

CONCLUSÃO

Este trabalho de conclusão de curso foi realizado com o propósito de desenvolver-se um protótipo de um sistema em Java de pré-processamento de imagens.

Para melhorar a eficiência da leitura ótica de caracteres, no aplicativo desenvolvido foi realizado a implementação dos algoritmos de pré-processamento de imagem. Com a utilização de técnicas de correção como filtragem, transformações geométricas, limiarização.

Com a utilização dos filtros foi possível realizar diversos testes com as imagens de referencia e assim obter a melhor seqüência de algoritmos a ser aplicados.

Com o pré-processamento das imagens, as mesmas foram submetidas a um leitor OCR, gerando um arquivo de texto e foram feitas comparações as imagens digitalizadas sem a aplicação dos algoritmos de pré-processamento.

Para que fosse avaliado o desempenho do sistemas de pré-processamento das imagens geradas em comparação as imagens digitalizadas, obteve-se um arquivo com o texto referencia a ser avaliado. Aplicaram-se as regras estabelecidas pela métrica da distância de Levenshtein, onde a mesma avalia o número de inclusões e exclusões em relação ao texto de referência.

Com os resultados obtidos pela métrica da distância de Levenshtein, foi possível determinar um ganho na eficiência da leitura OCR. A taxa de acerto do sistema resultou entre quatro a doze por cento.

Os resultados foram obtidos utilizando algumas imagens digitalizadas de referência, compararmos a um volume maior, o sistema de pré-processamento atendeu o propósito inicial, que seria o de tornar mais eficiente o reconhecimento de caracteres.

Para futuros estudos utilizando o pré-processamento das imagens para o reconhecimento ótico de caracteres, fica registrado como sugestão, o aprimoramento da taxa da eficiência dos algoritmos de pré-processamento de imagem, com a aplicação de novos métodos e métricas de avaliação.

Outro foco para futuros estudos será a integração do sistema de pré-processamento de imagem integrando-o a um digitalizador e a um sistema de reconhecimento.

REFERÊNCIAS

ALVES, N. F. Estratégias para melhoria do desempenho de ferramentas comerciais de reconhecimento óptico de caracteres. 2003. 119 f. Dissertação (Mestrado em Engenharia elétrica) Universidade Federal de Pernambuco, Pernambuco.

BLUELEAF. Escaneamento de livros para uso pessoal e pequenas empresas. Disponível em: <www.blueleaf-book-scanning.com/ocr_comparison_acrobat.html>. Acesso em 30 de maio de 2011.

CONNECTIVA. Manual do usuário LINUX. Rio de Janeiro: Conectiva, 1999.

DEMETRIO, T. Protótipo para Visualização, Processamento E Análise de Imagens Médicas. 2003. 68 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) Universidade do Extremo Sul Catarinense, Criciúma.

EFFORD, N. Digital image processing: a practical introduction using Java. New York: Addison-Wesley, 2000. 340p.

FILHO, O.; VIEIRA NETO, H. Processamento Digital de Imagens. Rio de Janeiro: Brasport, 1999.

FREEONLINEPHOTOEDITOR. Editor de imagens on-line. Disponível em: <www.freeonlinephotoeditor.com>. Acesso em 31 de agosto de 2009.

GONZALEZ, R. C.; WOODS, R. E. Processamento de imagens digitais. São Paulo: Edgard Blücher, 2000. 509 p.

HENG, C. Free OCR Software. Disponível: <<http://www.thefreecountry.com/utilities/ocr.shtml>>. Acesso em 15 setembro 2009.

HEWLETT-PACKARD. Suporte Técnico. Disponível: <<http://www.hp.com>>. Acesso em 5 de outubro de 2009.

JOCR. Aplicação OCR livre. Disponível em: <jocr.sourceforge.net/>. Acesso em 22 de setembro de 2011.

JUNKER, M.; HOCH, R.; DENGEL, A. On the Evaluation of Document Analysis Components by Recall, Precision, and Accuracy. Proceedings of 5th International Conference on Document Analysis and Recognition. India, pp. 713-716, 1999.

KUBIÇA, S. Gerenciamento eletrônico de documentos. Bate Byte, Curitiba, n. 88, jul. 1999.

LOPES, J. M. B. Formatos de imagem. 2008. 42 f. Trabalho de Conclusão de Curso (Licenciatura em Engenharia Informática e de Computadores) Instituto Superior Técnico, Universidade de Lisboa, Lisboa.

MACHADO, André Wilson; SOUKI, Bernardo Quiroga; MAZZIEIRO, Ênio Tonani Avaliação de quatro métodos de visualização de imagens digitais em odontologia. Revista Odonto Ciência, Porto Alegre, p. 132-138, 2006.

MAINTZ, J. B. A.; VIERGEVER, M. A. A survey of medical image registration. Medical Image Analysis, 1998.

MARQUES FILHO, O. VIEIRA NETO, H. Processamento Digital de Imagens . Rio de Janeiro: Brasport, 1999.

MELLO, C. A. B. Computacionais do Processamento de Imagens de Documentos Históricos. Recife: UFPE, 2003. Tese (Doutorado em Ciência da Computação), Centro de Informática, Universidade Federal de Pernambuco.

OCRAD. Aplicação OCR livre. Disponível em: <www.gnu.org/software/ocrad/>. Acesso em 15 de setembro de 2011.

OCROPUS. Aplicação OCR livre. Disponível em: <code.google.com/p/ocropus/>. Acesso em 30 de setembro de 2011.

OLIVEIRA, F. Processamento de imagens em escritórios. Disponível: <<http://biblioteca.senac.br/fulltext/pdf/2102039045.pdf>>. Acesso em: 1 junho 2009.

SEBASTIANI, R. L. Ferramenta de apoio a diagnósticos médicos baseados no processamento de imagens de ressonância magnética. Instituto de Ciências Exatas e Tecnológicas. Centro Universitário Feevale Novo Hamburgo, RS, p. 1-6, 2009.

STEPHEN, J. S. Programando em CLIPPER summer 87. Rio de Janeiro: Berkeley, 1992.

TESSERACT. Aplicação OCR comercial. Disponível em: < code.google.com/p/tesseract-ocr/>. Acesso em 20 de setembro de 2011.

TOPOCR. Aplicação OCR livre. Disponível em: < www.topocr.com/index.html>. Acesso em 25 de setembro de 2011.

APÊNDICE

Pré-Processamento de Imagens para o Reconhecimento Ótico de Caracteres

Vanderval Borges de Souza¹

¹Universidade do Extremo Sul Catarinense (UNESC)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

vandervalbs@gmail.com

Abstract. *This study demonstrates the development of a Java's application and has the purpose to apply the images preprocessing algorithms to read optic of characters. The techniques of improvement are applied in scanned images described in this theory fundamentation, even it's a known subject its application tends to contribute scientifically to the development of new techniques of algorithms implementation. The developed application read an user defined image and load it in a visible area, after this the application provides several of filters, to users can run them and obtain the waited results, and that the user can define the best filters to be applied, after this the wrong image is saved in a user's defined area. From this moment the created image is done to be submitted to the optic reader and it will create a text file, case be necessary, the text file created can be submitted to Levenshtein distance so the user can analyze the efficiency ratio of performed preprocessing.*

Resumo. *Este artigo descreve o desenvolvimento de uma aplicação em Java com o intuito de aplicar os algoritmos de pré-processamento de imagem para a leitura ótica de caracteres. São aplicadas em imagens digitalizadas as técnicas de aprimoramento descritas na fundamentação teórica, mesmo que seja uma abordagem já conhecida a sua aplicação tende a contribuir cientificamente para o desenvolvimento de novas técnicas de implementação dos algoritmos. A aplicação desenvolvida lê uma imagem definida pelo usuário e a carrega em uma área visível, após isso a aplicação disponibiliza uma série de filtros, para que o usuário possa executá-los e obter os resultados esperados, sendo que o usuário pode definir a melhor sequência de filtros a serem aplicados, depois disto a imagem errada é gravada na área pré-definida pelo usuário. A partir deste momento a imagem gerada esta pronta para ser submetidas ao leitor ótico de caracteres, que irá gerar um arquivo de texto, caso seja necessário o arquivo de texto gerado poderá ser submetido à métrica da distância de Levenshtein para que o usuário possa analisar os índices de eficiência dos pré-processamentos realizados*

1. Introdução

O processo de OCR consiste em realizar o reconhecimento de caracteres presentes em um arquivo de imagem digital, distinguindo-os de figuras, ruídos ou outros de dados que possam estar presentes e transpô-los para caracteres em padrão American Standard Code for Information Interchange (ASCII). Os sistemas de OCR classificam-se em on-line e off-line.

O OCR envolve as seguintes etapas, em sequência: aquisição de imagens; pré-processamento; segmentação; reconhecimento e pós-processamento (KUBIÇA, 1999).

A aquisição consiste em capturar os documentos e convertê-los em imagens digitais. A etapa de pré-processamento possibilita corrigir e preparar as imagens para os processos subsequentes. A segmentação identifica, dentro de uma imagem, blocos de texto, figuras, gráficos, dados, entre outros, para a etapa de reconhecimento.

A fase de reconhecimento visa classificar objetos (caracteres) nos blocos segmentados e selecionados em padrões por meio das características envolvidas. Esta etapa é altamente sensível a efeitos de deformações geométricas, tonalidades, texturas, sombras, ruídos, dentre outros. O pós-processamento realiza a análise do contexto visando eliminar ambigüidades no processo de reconhecimento, por exemplo, diferenciando o dígito zero "0" da letra O.

Neste Trabalho de Conclusão de Curso será abordada a etapa de Pré-Processamento que visa corrigir efeitos de degradação oriundos da aquisição e realçar dados relevantes (bem como eliminar dados que podem dificultar o processo) de forma a prover melhorias das respostas de OCR dos documentos. Exemplos de efeitos de degradação envolvem inclinação, ruído, falhas, dentre outros.

Podem ser usadas técnicas baseadas em equalização de histograma, filtragem, limiarização, transformações geométricas, eliminação (redução) de ruídos, afinamento ou esqueletização, operações morfológicas, dentre outras, sendo que estas abordagens podem ser híbridas, ou seja, elas podem estar combinadas, com distintos parâmetros de utilização bem como sintonias associadas, conduzindo a distintas imagens pré-processadas e, conseqüentemente, quando submetidas a um sistema de OCR, produzindo diferentes arquivos de texto (MELLO, 2003).

Este projeto irá investigar e comparar as principais técnicas de processamento de imagens e computação gráfica aplicada ao pré-processamento no OCR, por meio de um estudo teórico-experimental sobre cada uma delas.

O desempenho de sintonia e parâmetros utilizados será analisado por meio de métricas objetivas e ferramentas automáticas para avaliar as técnicas isoladamente e em conjunto em termos de acertos, erros e rejeições na transcrição por meio do processo de OCR sobre os textos contidos nas imagens originais.

A partir das análises efetuadas, será apresentada uma proposta do sistema de pré-processamento com os parâmetros avaliados, bem como a implementação de um protótipo de sistema com desempenho compatível ou superior aos sistemas propostos atualmente na literatura.

2. Imagens Digitais

Imagens são produzidas por meio de uma variedade de dispositivos físicos, como câmeras, equipamentos de radiografia, de ultrassom, radares entre outros, as utilizações destas imagens possuem diversos objetivos que vão do puro entretenimento até as aplicações sócias econômicas (MELLO, 2003).

Uma imagem digital é basicamente uma matriz onde são definidos todos os pontos ou pixels, onde cada ponto possui uma escala de cor. Através da função matemática $f(x, y)$ é possível encontrar a sua localização e o valor que representa a sua cor e assim manipulá-la, sendo, que sempre

partiremos de um ponto de origem considerando ele como o número zero até a sua extremidade, onde pode haver inúmeras posições.

3. Reconhecimento Ótico de Caracteres

O reconhecimento óptico de caracteres é a simples conversão de uma imagem em uma representação numérica adequada para o processamento digital subsequente. A aquisição destas imagens compreende dois elementos principais, o primeiro é um dispositivo físico sensível a uma faixa de energia no espectro eletromagnético o segundo converter o sinal elétrico analógico em informação digital, podendo ser representada através dos bits 0 e 1 (FILHO; VIEIRA NETO, 1999).

O processo chamado de Reconhecimento Óptico de Caracteres (Optical Character Recognition - OCR) é um sistema automático que deve diferenciar os caracteres no formato ASCII das imagens e de outros dados para que com isso se obtenha apenas a parte textual da imagem.

4. Pré-Processamento de imagens para o Reconhecimento Ótico de Caracteres

Nesta fase serão apresentados os recursos utilizados, o desenvolvimento, avaliação e os resultados obtidos através dos algoritmos, métricas e métodos utilizados para o pré-processamento de imagens para a leitura OCR.

4.1 Algoritmos para o Pré-Processamento de Imagens

Para obtermos um melhor resultado do programa OCR ao ser aplicarmos as imagens de referência encontradas, serão utilizados os algoritmos de pré-processamento, onde cada um vai executar uma função específica para minimizar os problemas encontrados.

Algoritmos para a Filtragem como os filtros passa-alta, passa-baixa, Gausiano, Prewitt, Kirsch, Robinson e medianos sendo estes filtros considerados muito importantes para o pré-processamento de imagens (GONZALES; WOODS, 2000). A filtragem passa-alta tende a acentuar melhor os níveis de cinza de alta frequência, ignorando os níveis de cinza de baixa frequência, enquanto a tendência da filtragem passa-baixa é a de minimizar os tons de cinza de baixa frequência, dando assim mais enfoque aos tons de cinza de alta frequência. Com a utilização da filtragem mediana é possível reduzir a quantidade de ruídos, pois o algoritmo busca os valores de RGB pixel a pixel, soma e divide o resultado da soma por três e assim obtêm o valor da media referente ao pixel. O filtro Gausiano trata-se de um filtro que trabalha com baixas frequências e assim denominada como sendo do tipo de filtro passa-baixa, onde a aplicação do algoritmo é feita sucessivas vezes seguindo sempre o comprimento da matriz informada. O algoritmo do filtro Prewitt é composta pelos comprimentos das oito matrizes que compõe o filtro, sendo que cada uma delas trata de uma posição da imagem definidas como direcional região norte, noroeste, oeste, sudoeste, sul, sudeste, leste e nordeste a imagem tende ao realce de altas frequências (DEMETRIO 2003), ocorrendo o mesmo ocorre com o filtro Kirsh.

Com a utilização das transformações geométricas do tipo rotação, pode-se alinhar a imagem digitalizada. O grande problema para a rotação de uma imagem de texto digitalizada é definir o ângulo correto, para assim obter o melhor desempenho.

O algoritmo de limiarização é utilizado para determinar o nível de cinza da imagem, tornando-a assim mais nítida para a leitura OCR. Definida a partir de um ponto chamado de limiar,

onde, os valores RGB dos pixels são comparados ao valor limiar e se caso o valor limiar for maior que o valor RGB do pixel o algoritmo insere um pixel preto, caso contrário ele insere um pixel branco.

4.2 Imagens de Referência

Para a aplicação dos algoritmos utilizados para a execução das correções e para obtermos os resultados da métrica de avaliação, adquirimos algumas imagens por meio da digitalização de alguns documentos utilizando um scanner e utilizando imagens de texto conhecidas.

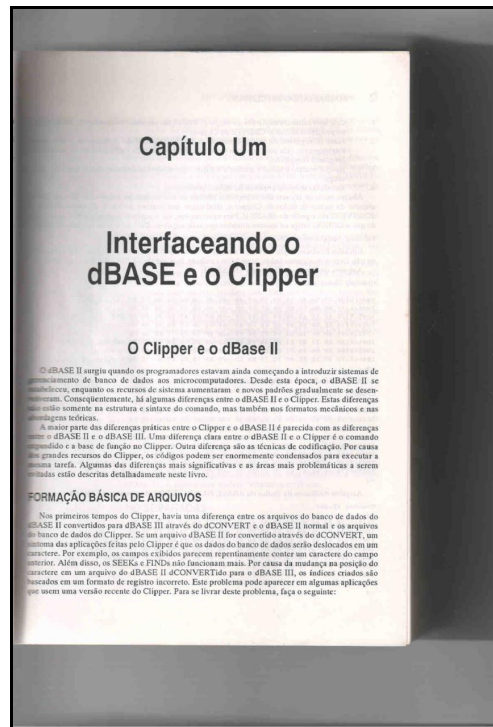


Figura 1. Imagem de Referência 1

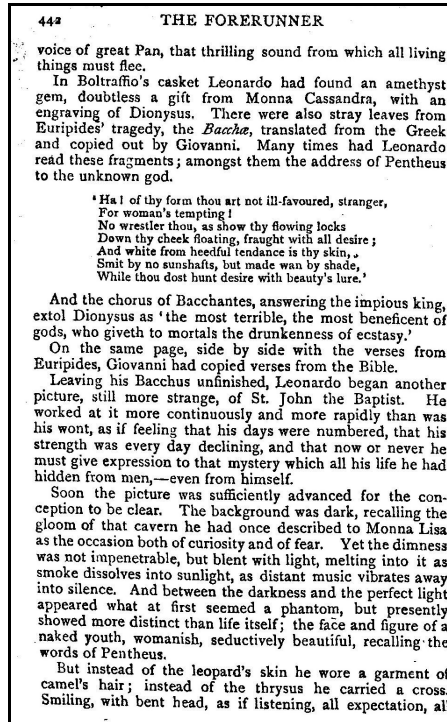


Figura 2. Imagem de Referência 2

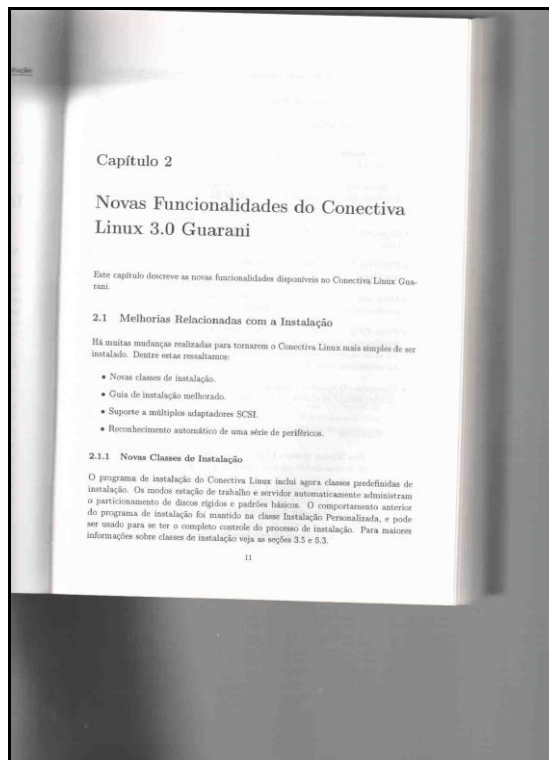


Figura 3. Imagem de Referência 3

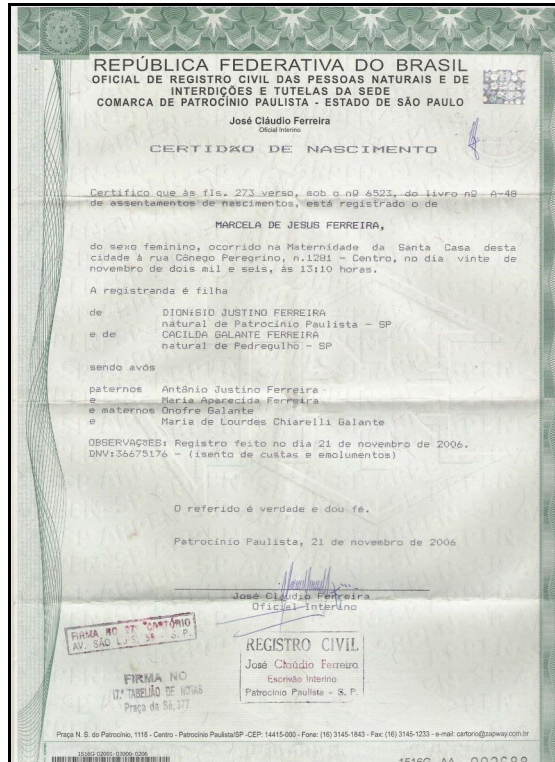


Figura 4. Imagem de Referência 4

5. Resultados Obtidos

Para demonstrar a eficiência dos resultados a serem obtidos através dos algoritmos de pré-processamento de imagens, iremos utilizar a métrica conhecida como distância de Levenshtein que é responsável por medir o índice de acerto e a quantidade de caracteres que devem ser inseridos e ou apagados para se obter a palavra correta.

Aplicando o algoritmo da distância de Levenshtein para medir o tamanho da diferença entre o texto original e os textos obtidos com a utilização de um programa de leitura OCR sem a aplicação de nenhum pré-processamento de imagem, obtemos os seguintes resultados:

- a) imagem de referencia um: 340 exclusões e ou inclusões e um índice de acerto de 82,49%;
- b) imagem de referencia dois: 395 exclusões e ou inclusões e um índice de acerto de 82,05%;
- c) imagem de referencia três: 170 exclusões e ou inclusões e um índice de acerto de 82,33%;
- d) imagem de referencia numero quatro: 914 exclusões e ou inclusões e um índice de acerto de 27,52%;

O próximo passo é aplicar a métrica da distância de Levenshtein para comparar o texto original e o texto obtido do programa de leitura OCR obtidos com a ajuda do pré-processamento de imagens:

- a) imagem de referencia um: 252 exclusões e ou inclusões e um índice de acerto de 87,02%;
- b) imagem de referencia dois: 199 exclusões e ou inclusões e um índice de acerto de 90,96%;
- c) imagem de referencia três: 127 exclusões e ou inclusões e um índice de acerto de 86,80%;
- d) imagem de referencia quatro: 752 exclusões e ou inclusões e um índice de acerto de 40,36%;

Com os resultados obtidos acima, foi montada a Tabela 1 e assim podendo assim avaliar os resultados obtidos e determinar se houve ou não benefícios com a utilização do pré-processamento de imagens, lembrando que foram avaliadas as quatro imagens de referência sem o pré-processamento de imagem e as mesmas imagens submetidas aos algoritmos de pré-processamento de imagem, onde cada imagem foi submetida a métrica da distância de Levenshtein.

Tabela 1 - Aplicação da métrica da Distância de Levenshtein

Imagem	Sem Processamento	Com Processamento	% Eficiência
imagem de referencia 1	340 – 82,49%	252 – 87,02%	+ 4,53 %
imagem de referencia 2	395 – 82,05%	199 – 90,96%	+ 8,91%
imagem de referencia 3	170 – 82,33%	127 – 86,80%	+ 4,47%
imagem de referencia 4	914 – 27,52%	752 – 40,36%	+ 12,84%

Referências

ALVES, N. F. Estratégias para melhoria do desempenho de ferramentas comerciais de reconhecimento óptico de caracteres. 2003. 119 f. Dissertação (Mestrado em Engenharia elétrica) Universidade Federal de Pernambuco, Pernambuco.

BLUELEAF. Escaneamento de livros para uso pessoal e pequenas empresas. Disponível em: <www.blueleaf-book-scanning.com/ocr_comparison_acrobat.html>. Acesso em 30 de maio de 2011.

CONNECTIVA. Manual do usuário LINUX. Rio de Janeiro: Conectiva, 1999.

DEMETRIO, T. Protótipo para Visualização, Processamento E Análise de Imagens Médicas. 2003. 68 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) Universidade do Extremo Sul Catarinense, Criciúma.

EFFORD, N. Digital image processing: a practical introduction using Java. New York: Addison-Wesley, 2000. 340p.

FILHO, O.; VIEIRA NETO, H. Processamento Digital de Imagens. Rio de Janeiro: Brasport, 1999.

FREEONLINEPHOTOEDITOR. Editor de imagens on-line. Disponível em: <www.freeonlinephotoeditor.com>. Acesso em 31 de agosto de 2009.

GONZALEZ, R. C.; WOODS, R. E. Processamento de imagens digitais. São Paulo: Edgard Blücher, 2000. 509p.

HENG, C. Free OCR Software. Disponível: <<http://www.thefreecountry.com/utilities/ocr.shtml>>. Acesso em 15 setembro 2009.

HEWLETT-PACKARD. Suporte Técnico. Disponível: <<http://www.hp.com>>. Acesso em 5 de outubro de 2009.

JOCR. Aplicação OCR livre. Disponível em: <jocr.sourceforge.net/>. Acesso em 22 de setembro de 2011.

JUNKER, M.; HOCH, R.; DENGEL, A. On the Evaluation of Document Analysis Components by Recall, Precision, and Accuracy. Proceedings of 5th International Conference on Document Analysis and Recognition. India, pp. 713-716, 1999.

KUBIÇA, S. Gerenciamento eletrônico de documentos. Bate Byte, Curitiba, n. 88, jul. 1999.

LOPES, J. M. B. Formatos de imagem. 2008. 42 f. Trabalho de Conclusão de Curso (Licenciatura em Engenharia Informática e de Computadores) Instituto Superior Técnico, Universidade de Lisboa, Lisboa.

MACHADO, André Wilson; SOUKI, Bernardo Quiroga; MAZZIEIRO, Ênio Tonani Avaliação de quatro métodos de visualização de imagens digitais em odontologia. Revista Odonto Ciência, Porto Alegre, p. 132-138, 2006.

MAINTZ, J. B. A.; VIERGEVER, M. A. A survey of medical image registration. Medical Image Analysis, 1998.

MARQUES FILHO, O. VIEIRA NETO, H. Processamento Digital de Imagens . Rio de Janeiro: Brasport, 1999.

MELLO, C. A. B. Computacionais do Processamento de Imagens de Documentos Históricos. Recife: UFPE, 2003. Tese (Doutorado em Ciência da Computação), Centro de Informática, Universidade Federal de Pernambuco.

OCRAD. Aplicação OCR livre. Disponível em: <www.gnu.org/software/ocrad/>. Acesso em 15 de setembro de 2011.

OCROPUS. Aplicação OCR livre. Disponível em: <code.google.com/p/ocropus/>. Acesso em 30 de setembro de 2011.

OLIVEIRA, F. Processamento de imagens em escritórios. Disponível: <<http://biblioteca.senac.br/fulltext/pdf/2102039045.pdf>>. Acesso em: 1 junho 2009.

SEBASTIANI, R. L. Ferramenta de apoio a diagnósticos médicos baseados no processamento de imagens de ressonância magnética. Instituto de Ciências Exatas e Tecnológicas. Centro Universitário Feevale Novo Hamburgo, RS, p. 1-6, 2009.

STEPHEN, J. S. Programando em CLIPPER summer 87. Rio de Janeiro: Berkeley, 1992.

TESSERACT. Aplicação OCR. Disponível em: <code.google.com/p/tesseract-ocr/>. Acesso em 20 de setembro de 2011.

TOPOCR. Aplicação OCR livre. Disponível em: <www.topocr.com/index.html>. Acesso em 25 de setembro de 2011.