



UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE ENGENHARIA BIOMÉDICA

**BR BRAILLE: PROGRAMA TRADUTOR DE TEXTOS BRAILLE
DIGITALIZADOS PARA CARACTERES ALFANUMÉRICOS EM
PORTUGUÊS**

Membros da Banca Examinadora:

Profa. Dra. Vera Lúcia da Silveira Nantes Button
DEB, FEEC, UNICAMP

Prof. Dr. José Wilson Magalhães Bassani
DEB, FEEC, UNICAMP

Prof. Dr. Antônio Augusto Fasolo Quevedo
DEB, FEEC, UNICAMP

Prof. Dr. José Oscar Fontanini de Carvalho
CCEAT, PUC-CAMPINAS

Autora: Cláudia Maria Caixeta Bezerra

Orientadora: Profa. Dra. Vera Lúcia da Silveira Nantes Button

Dissertação apresentada como parte dos requisitos exigidos para obtenção do
Título de MESTRE EM ENGENHARIA ELÉTRICA.

Campinas – SP - Brasil
Janeiro de 2003

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

B469b	Bezerra, Cláudia Maria Caixeta BR Braille: programa tradutor de textos Braille digitalizados para caracteres alfanuméricos em português / Cláudia Maria Caixeta Bezerra.--Campinas, SP: [s.n.], 2003. Orientador: Vera Lúcia da Silveira Nantes Button. Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação. 1. Processamento de imagens. 2. Cegos – Sistemas de impressão e escrita. 3. Deficientes visuais. 4. Morfologia matemática. I. Button, Vera Lúcia da Silveira Nantes. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.
-------	--

Resumo

Nesta dissertação de mestrado foi elaborado um programa tradutor de texto em código Braille para caracteres alfanuméricos, denominado BR Braille. O programa tem por finalidade facilitar a comunicação, através da escrita, entre pessoas com deficiência visual alfabetizadas em Braille e pessoas não-deficientes que não conheçam o sistema Braille, como nas salas de aula regulares em que os alunos cegos são inseridos, e nas quais, freqüentemente, os educadores não são alfabetizados em Braille. As celas Braille são compostas de seis pontos perfurados em alto relevo no papel, que quando são digitalizadas em níveis de cinza geram uma imagem com fundo igual 255 (papel branco), e valores menores de níveis de cinza para os pontos (mais escuros). Após a aquisição da imagem, foram realizados os seguintes passos: pré-processamento, segmentação, extração de características, e o reconhecimento e interpretação. No pré-processamento, faz-se a filtragem da imagem e também a rotação. Na segmentação tem-se a separação dos pontos, que são os objetos de interesse. Posteriormente através da extração de características, verifica-se a que celas estes pontos pertencem. Com as celas separadas é possível realizar a comparação de cada cela da imagem processada às celas-padrão, identificando os caracteres alfanuméricos correspondentes. O texto final não tem correção ortográfica, pois uma de suas possíveis aplicações é auxiliar na correção de trabalhos escolares. Os valores dos níveis de cinza correspondentes aos pontos vão depender muito do quanto o alto relevo do texto original está preservado. Além disso, a folha não pode ter rotação maior que 2°. Utilizando-se um Pentium III 750MHz com 256Mb de RAM, o tempo de processamento de uma folha com 654 caracteres, digitalizada em 100 dpi, é de aproximadamente 60 segundos, com erro estimado de 0,5%. Neste trabalho foram avaliadas todas as restrições à realização das etapas da tradução, levantando-se as possíveis falhas e alternativas para correção.

Abstract

In this master's degree dissertation a program was built (BR Braille) to translate Braille's code to alphanumeric characters. The purpose of this program was to facilitate the writing communication between Braille alphabetized blind people and people with normal vision that don't know the system Braille, as in regular classrooms where blind students are assisted, frequently, by Braille non-alphabetized teachers. The Braille cells have six dots punched in high relief in paper; when a Braille text is digitized in grey tone, the resulting image has background represented by the value 255 (white paper), and smaller values of grey tone represent the dots (that are darker than the background). After the image acquisition, the following steps were accomplished: preprocessing, segmentation, extraction of characteristics, recognition and interpretation. In preprocessing, the image was filtered and rotated; in the segmentation step, the dots, that are the objects of interest, were detached; the extraction of characteristics allowed verifying to which cell a determined point belongs. With the cells separated it was possible to compare each cell of the image processed to the pattern cells to identify the corresponding alphanumeric character. The final text does not have orthographic correction, since one of its possible applications is to help teachers correcting school labors. The values of the gray levels of the dots will depend on the high relief preservation of the original text, and on the text page rotation (no more than 2 degrees). When a Pentium III 750MHz with 256Mb of RAM was used, the time to process a text page with 654 characters, and digitized in 100 dpi, was approximately 60 seconds, with 0,5% error. The text restrictions to the BR Braille utilization and some solutions for correction of the possible flaws were explained.

A Deus, ao meu marido e a minha mãe

Agradecimentos:

Louvado seja Deus que me permitiu conquistar este importante objetivo. "O cavalo prepara-se para o dia da batalha, mas a vitória vem do Senhor." Provérbios 21:31.

A Capes pelo apoio financeiro.

A minha orientadora professora Dra. Vera Lúcia da Silveira Nantes Button pela confiança depositada na minha capacidade de trabalho, e amabilidade com que sempre me tratou.

Ao professor Dr. José Oscar Fontanini de Carvalho por sua sugestão do tema do trabalho.

Ao professor Dr. Alexandre Xavier Falcão por suas informações e sugestões essenciais a realização do trabalho.

A todos os professores do Departamento de Engenharia Biomédica.

A funcionária da biblioteca do Centro de Engenharia Biomédica, Valdinéia e, suas estagiárias Sílvia, Íris e Mirian.

A Dagmar e a Vanessa do Centro Cultural Louis Braille, pelas folhas em Braille para a realização dos testes preliminares.

Ao estagiário e amigo José Paulo Eboli por sua competência e presteza com que atendia as minhas solicitações.

Aos alunos de iniciação científica (alunos do 2º ano da Engenharia de Computação da UNICAMP) Adriana Keiko e Rodrigo Barros, que têm uma capacidade de trabalho e competência admiráveis. Além de seres humanos fantásticos com certeza serão profissionais super conceituados.

Às amigas Daniela, Gláucia e Suzy que tornaram o dia-a-dia muito menos desgastante, e muito mais agradável.

A todos os amigos do departamento que ajudaram direta ou indiretamente na realização do trabalho, Gentil, Maurício, Valéria, Lúcio, Pedro, Jorge, Eduardo Jorge e Ana Cristina.

Ao Pastor Marcílio Teixeira da Igreja Batista Central de Campinas, por levar tão bem a palavra de Deus, palavras de paz, amor e compromisso com Jesus Cristo.

Aos meus irmãos Carlos, Célio, Carmen e Celma por seu amor, torcida e palavras de incentivo.

Aos meus sobrinhos amados Alexandre Filho (também meu afilhado querido), Fernanda e Eduardo, Marina, Amanda e Lorena, Davi e Andressa, por seu carinho, e doçura que confortam e renovam minhas forças sempre que os vejo.

A minha mãe que foi sempre uma super mãe. Carinho, amor e dedicação são sinônimos desta mulher adorável e admirável. Maria de Lourdes Caixeta obrigada por ser minha mãe. Esta conquista também é sua.

Um agradecimento super especial a este homem, que mais que meu marido, é meu amigo, meu companheiro, meu conselheiro, meu amado. Roberto de Araújo Bezerra obrigada por tudo o que você fez. Todo o trabalho realizado teve uma participação direta dele, por isto, minha homenagem.

Sumário

<i>Capítulo 1 – Introdução</i>	1
1.1 Objetivos do Programa.....	1
1.2 Aquisição das Imagens.....	2
1.3 Proposta do Trabalho.....	3
1.4 Estrutura do Trabalho.....	3
 <i>Capítulo 2 – O Sistema Braille.....</i>	4
2.1 Uma Breve História do Braille.....	4
2.2 O Sistema Braille.....	5
2.3 Instrumentos para Escrita Braille.....	11
2.3.1 Reglete e punção.....	11
2.3.2 Máquina de Escrever Braille.....	12
2.3.3 Impressora Braille.....	13
2.4 Leitura Braille.....	13
2.5 Alguns <i>Softwares</i> de Auxílio ao Cego.....	14
2.5.1 <i>Softwares</i> Leitores de Tela de Computador.....	14
2.5.2 Reconhecimento de Voz.....	14
2.5.3 Tradução de Caracteres Alfanuméricos para Braille.....	15
2.5.4 Optical Braille Recognition.....	15
2.6 A Deficiência Visual no Brasil.....	15
 <i>Capítulo 3 – Formato de Imagens.....</i>	19
3.1 Descrição de Scanners.....	19
3.1.1 Tipos de Scanners.....	20
3.2 Especificação.....	21
3.2.1 Sensores Utilizados nos <i>Scanners</i>	21
3.2.2 Resolução.....	22
3.2.3 Profundidade de Bit.....	23
3.3.4 Medida Dinâmica.....	23
3.3 <i>Softwares</i>	24
3.4 Formatos de Imagem.....	24
3.4.1 Apresentação dos Formatos.....	28
3.4.2 Exemplo das Diferenças dos Formatos de Imagem.....	33
3.5 Formato Utilizado - Justificativa.....	36
 <i>Capítulo 4 – Tópicos em Processamento Digital de Imagens.....</i>	37
4.1 Representação de Imagens.....	37
4.2 Passos Fundamentais em Processamento de Imagem.....	39

4.3	Conceitos Básicos.....	40
4.3.1	Histograma.....	40
4.3.2	Noção de Conexidade.....	42
4.4	Limiarização.....	43
4.4.1	Limiarização de Otsu.....	44
4.5	Morfologia Matemática.....	48
4.5.1	Erosão e Dilatação Binárias.....	50
4.5.2	Erosão e Dilatação em Níveis de Cinza.....	52
4.5.3	Transformação H-Mínima ou H-Basin.....	54
4.5.4	Rotulação.....	54
4.6	Rotação.....	55
<i>Capítulo 5 –Processamento das Imagens Braille.....</i>		56
5.1	Requisitos do Sistema.....	56
5.2	Aquisição das Imagens.....	57
5.3	Introdução ao Pré-Processamento.....	58
5.4	A Escolha das Etapas do Pré-Processamento.....	59
5.5	A Sequência do Pré-Processamento.....	60
5.6	Detalhes da Segmentação.....	69
5.7	Extração de Características.....	73
5.8	Reconhecimento e Interpretação.....	74
5.9	Fim do Processamento das Imagens Braille.....	74
<i>Capítulo 6 –Teste e Resultados.....</i>		77
6.1	Uso de Diferentes <i>Scanners</i> com Folhas Escritas com Reglete.....	77
6.2	Programa Com e Sem o Código da Rotação.....	83
6.3	Resoluções Utilizadas para Digitalização do Programa.....	87
<i>Conclusão.....</i>		92
<i>Bibliografia.....</i>		96
<i>Anexos.....</i>		100
Anexo – I.....		102
Anexo – II.....		103
Anexo – III.....		104
Anexo – IV.....		105

Lista de Figuras

Figura 2.1. Representação da forma de leitura da Cella Braille.....	5
Figura 2.2. Representação das primeiras letras do alfabeto Braille. Observação: os pontos escuros representam o que estará em relevo para percepção tátil.	6
Figura 2.3. Exemplo de duas letras do alfabeto Braille.....	6
Figura 2.4. Em (1) exemplo das letras “a” e “u”. Em (2) o exemplo de “f” e “ç”.....	7
Figura 2.5. Representação Braille dos números 1, 5 e 0.....	7
Figura 2.6. Exemplo da formação do número 150 em Braille.....	7
Figura 2.7. Sinais de pontuação em Braille.....	8
Figura 2.8. Representação de mais alguns símbolos do alfabeto Braille.....	8
Figura 2.9. Alfabeto Braille (FUNDAÇÃO HILTON ROCHA, 1987).....	9
Figura 2.10. Símbolos Matemáticos (Extraído de Nascimento, Bruno, Rímoli e Cordeiro, 1987).....	10
Figura 2.11: Representação da escrita de uma Cella Braille.....	11
Figura 2.12. Figura ilustrativa do reglete de bolso e do punção.....	12
Figura 2.13: Máquina Braille (Extraído de FUNDAÇÃO HILTON ROCHA, 1987)...	13
Figura 3.1. <i>Scanner</i> de mesa.....	20
Figura 3.2. Representação do mapa de <i>pixel</i> , tendo na segunda grade valores do nível de cinza em cada coordenada.....	26
Figura 3.3. Representação da <i>Look-up Table</i> com 8 bits.....	27
Figura 3.4. (a) Fragmento da representação da imagem retirada da Figura 3.2 – sem compressão. (b) Fragmento da representação da imagem retirada da Figura 3.2 – com compressão RLE (redução de 22.23%, 2 em 9 <i>pixels</i>). Onde:	
▪ -1: código de controle do RLE	
▪ 4: número de repetições do valor 255.....	28
Figura 3.5. Tela de abertura da parte inicial do arquivo frag3p.pnm (imagem digitalizada da letra r em Braille), salvo em ASCII, aberta no bloco de notas do Windows 2000®	30
Figura 3.6. Imagem digitalizada (59x60), arquivo de extensão bmp com valor de profundidade de bits (Bit Depth), 8.....	34
Figura 4.1. Reticulado uniforme da representação matricial da imagem.....	38
Figura 4.2. Diagrama em blocos mostrando os passos do processamento de imagens digitais.....	39
Figura 4.3. Imagem digital de um texto Braille.....	41
Figura 4.4. Histograma correspondente à imagem digitalizada da Figura 4.3.....	41
Figura 4.5. Exemplos de (a) vizinhança-4 e (b) vizinhança-8.....	42
Figura 4.6. Representação de três componentes 4-conexas ou uma componente 8-conexa.....	43
Figura 4.7. Representação da divisão das classes C_0 e C_1 pela limiarização de Otsu, em uma imagem de L níveis de cinza. (a) Imagem de entrada; (b) imagem limiarizada.....	48
Figura 4.8. Representação dos elementos estruturantes mais utilizados.....	49

Figura 4.9. (a) Imagem digitalizada binária da letra “r”; (b) imagem da letra “r” em (a) erodida; (c) negativo da imagem digitalizada binária da letra “r”; e (d) imagem da letra “r” em (c)erodida.....	51
Figura 4.10. (a) Imagem digitalizada binária da letra “r”; (b) imagem da letra “r” em (a) dilatada; (c) negativo da imagem digitalizada binária da letra “r”; e (d) imagem da letra “r” em (c) dilatada.....	52
Figura 5.1. Desenho esquemático dos equipamentos necessários para utilização do programa tradutor.	57
Figura 5.2. Imagem de entrada com os valores dos <i>pixels</i> de fundo menores que 250..	61
Figura 5.3. Imagem de entrada com o valor dos <i>pixels</i> de fundo igual a 255.....	62
Figura 5.4. Histograma da imagem com fundo predominantemente escuro mostrada na Figura 5.2 (MATLAB®).....	62
Figura 5.5. Histograma da imagem com fundo predominantemente branco apresentada na Figura 5.3 (MATLAB®).....	63
Figura 5.6. Imagem da Figura 5.2 limiarizada.....	64
Figura 5.7. Resultado da rotulação feita a partir da imagem da Figura 5.6.....	65
Figura 5.8. Figura demonstrativa da composição de uma cela Braille.....	69
Figura 5.9. Centros das componentes conexas de duas linhas de uma imagem Braille digitalizada, onde estão representados os maiores e menores valores das coordenadas (x, y).	70
Figura 5.10. Divisão da primeira coluna (divisão parcial das celas) pertencente ao texto representado na Figura 5.9.....	71
Figura 5.11. (a) Segmentação da primeira cela Braille contida na figura 5.10; (b) Figura obtida a partir da segmentação da primeira cela da segunda linha do texto encontrada na Figura 5.10 (divisão total das celas).....	71
Figura 5.12. Árvore binária de busca da extração de características dos caracteres Braille.....	75
Figura 5.13. Texto traduzido a partir da imagem contida na Figura 5.2. Existe a presença de dois erros (e _) de tradução, e um (mãoè) contido no próprio texto em Braille (folha original).....	76
Figura 6.1. Texto traduzido pelo BR Braille da folha (Mateus 5:1-8) digitalizada no <i>scanner</i> da Tabela 6.3. As marcas em vermelho indicam as letras traduzidas com erro, sendo que a letra marcada em amarelo corresponde a erro na folha Braille original.....	82
Figura 6.2. Texto “Mateus 7” (folha Mateus 7: 7-12) processado com o programa BR Braille com o código de rotação desabilitado. As marcas verdes indicam onde estão as letras corretamente traduzidas.....	85
Figura 6.3 . Texto “Mateus 7” (folha Mateus 7: 7-12) processado com o programa BR Braille com o código de rotação habilitado. As marcas vermelhas indicam onde estão as letras não foram corretamente traduzidas.....	86
Figura 6.4 . Texto traduzido pelo BR Braille (digitalização 1- Tabela 6.8), com 0 % de erro, sendo que a letra marcada corresponde a erro na folha Braille original..	89
Figura 6.5 . Texto traduzido pelo BR Braille (digitalização 6- Tabela 6.8), com 0,28 % de erro, sendo que a letra marcada em amarelo corresponde a erro na folha Braille original e as letras em vermelho, os erros causados pelo programa tradutor.....	90

Lista de Tabelas

Tabela 2.1. Dados do Censo 1999 sobre a população com necessidades especiais por grupo de idade (Secretaria de Educação Especial, 2002).....	16
Tabela 2.2. Dados do censo de 1999, com o número de matrículas por tipo de necessidade especial em atividade escolar (Secretaria de Educação Especial, 2002).....	17
Tabela 3.1. Exemplos de identificadores do tipo do formato.....	29
Tabela 3.2. Dados das características das imagens, representadas pela figura 3.6, em diversos formatos. Dados obtidos através do comando “imfinfo” do MATLAB®	35
Tabela 6.1. Dados dos processamentos do programa BR Braille utilizando o <i>scanner</i> da marca HP Scanjet 4400c de folhas escritas com reglete e punção.....	78
Tabela 6.2. Dados dos processamentos do programa BR Braille utilizando o <i>scanner</i> da marca Genius ColorPage-Life Pro de folhas escritas com reglete e punção.....	79
Tabela 6.3. Dados dos processamentos do programa BR Braille utilizando o <i>scanner</i> da marca Scanner Boeder Sm@rtScan Slim Edition de folhas escritas com reglete e punção.....	80
Tabela 6.4. Dados dos processamentos do programa BR Braille completo utilizando o <i>scanner</i> da marca Scanner HP Scanjet 4400c para digitalizar folhas escritas na impressora Braille	83
Tabela 6.5. Dados dos processamentos do programa BR Braille com rotação desabilitada o utilizando o <i>scanner</i> da marca Scanner HP Scanjet 4400c para digitalizar folhas escritas na impressora Braille.....	84
Tabela 6.6. Dados do processamento do programa BR Braille utilizando para digitalização de folhas escritas com reglete e punção, o <i>scanner</i> da marca Scanner HP Scanjet 4400c com resolução igual a 100 dpi.....	87
Tabela 6.7. Dados do processamento do programa BR Braille utilizando para digitalização de folhas escritas com reglete e punção, o <i>scanner</i> da marca Scanner HP Scanjet 4400c com resolução igual a 200 dpi.....	88
Tabela 6.8. Erros de tradução do programa BR Braille utilizando o <i>scanner</i> da marca Scanner HP Scanjet 4400c com resolução igual a 100 dpi para 10 digitalizações de uma mesma folha escrita com reglete e punção.....	91

Lista de Algoritmos

Algoritmo 5.1. Algoritmo útil a várias etapas do processamento de imagem.....	66
Algoritmo 5.2. Algoritmo de limiarização de Otsu e do processo de limiarização.....	67
Algoritmo 5.3. Algoritmo da Transformação H-Mínima.....	68
Algoritmo 5.4. Algoritmo de Rotulação.....	68
Algoritmo 5.5. Algoritmo de Rotação.....	69
Algoritmo 5.6. Algoritmo da Segmentação das Celas Braille.....	73

1.1. Objetivo do Programa

Deficientes visuais totalmente cegos, em geral, realizam leitura através do alfabeto Braille, após um período de aprendizagem, normalmente realizada em centros especializados.

O acesso aos microcomputadores pessoais também é possível através de adaptação destes, como por exemplo, com reconhecedores de voz que permitem a substituição do teclado de um microcomputador. A leitura também pode ser feita através da sonorização vocal de palavras ou impressão em impressora Braille.

Por outro lado, o acesso aos textos produzidos em Braille para as pessoas que convivem com os deficientes visuais, mas que não são alfabetizadas neste sistema é difícil. Por exemplo, acompanhamento dos professores ao aluno cego ou mesmo a correspondência pessoal, feita com código Braille em papel, só pode ser lida pelo não-deficiente que seja alfabetizado em Braille, que não é o caso da maioria dos familiares, amigos e professores.

O programa BR Braille tradutor de textos Braille digitalizados para textos em caracteres alfanuméricos em português foi criado para atender a necessidade de ampliar a autonomia de pessoas cegas alfabetizadas em Braille.

O programa tem por finalidade facilitar a comunicação, entre pessoas com deficiência visual alfabetizadas em Braille e pessoas não deficientes que não conheçam o sistema Braille. Uma aplicação desse programa seria nas salas de aula que trabalham com a inclusão de crianças com deficiência visual, mas que são atendidas, na maior parte do tempo, por educadores não-alfabetizados em Braille. Estas escolas, geralmente, recebem os chamados professores itinerantes que são professores especializados, ou seja, conhecedores do sistema Braille. Essas visitas são feitas periodicamente gerando ansiedade na espera da correção dos trabalhos escolares, principalmente, nas crianças.

Uma outra utilização seria em cursos à distância não mediados por computador, favorecendo, principalmente, aquelas pessoas que têm menor condição de acesso a informática, nos pequenos municípios do Brasil. Seria possível, portanto, que ao escrever através do uso de reglete ou máquina Braille, que seus textos ou trabalhos enviados aos orientadores a longa distância fossem lidos e corrigidos. Dando-lhes mais uma oportunidade de crescimento pessoal e intelectual.

A geração de autonomia leva a uma sociedade mais igualitária com a diminuição do preconceito quanto à capacidade de pessoas cegas em estarem inseridas na sociedade, principalmente se estas pessoas, por condição social, não possuem acesso a microcomputadores. Essa é uma realidade nas pequenas cidades e na periferia das grandes cidades, que têm pouca infra-estrutura, e uma população muito carente. Ampliar o leque de meios disponíveis para que estas pessoas recebam uma educação formal, e assim se tornem menos dependentes, é o principal objetivo do BR Braille.

É relevante o aspecto da simplicidade dos equipamentos necessários para a utilização do programa: um microcomputador (Pentium 100 MHz, 32 M de RAM, 4.5 GB ou superior) e um scanner-de-mesa. Equipamentos de baixo custo e que já se encontram disponíveis em várias escolas. Isso traz a possibilidade de uma maior adesão ao uso do mesmo por instituições de ensino, mesmo as localizadas em locais mais carentes, e, que possuem menor infra-estrutura.

1.2. Aquisição das Imagens

As imagens Braille são digitalizadas em um *scanner*-de-mesa qualquer, e então são armazenadas em formato BMP (*bitmap*).

É desejável que a imagem a ser submetida ao processo de digitalização tenha altura e largura adequadas, de modo que a folha com o texto em Braille esteja contida sobre o vidro do *scanner*, sem haver nem cortes e nem sobras do texto, para que não haja perda de informação no processo de digitalização.

Pode ser necessário fazer alguns ajustes na imagem original a ser digitalizada, como por exemplo, retirar possíveis marcas na folha de papel causadas pela reglete e que não sejam pontos Braille, através de um programa como o Paint Brush do Windows[®].

Um outro aspecto importante ao bom funcionamento do programa tradutor é a necessidade que o texto original seja digitalizado horizontalmente ao plano do vidro do *scanner*, sem gerar rotação na imagem digitalizada.

Os textos Braille devem ser digitalizados em uma resolução de 100 ou 200 dpi e a escolha da resolução fica a cargo do usuário. Quanto menor a resolução menor será o tempo para obtenção do texto traduzido.

1.3. Proposta do Trabalho

Neste trabalho foi desenvolvido um programa para traduzir textos Braille digitalizados, através de técnicas de processamento de imagens, que fez uso de conhecimentos sobre morfologia matemática, algoritmo de automação da binarização de imagens em nível de cinza, e algoritmo de rotulação para o pré-processamento das imagens. Além disso, também foi necessária a utilização de estruturas de dados como árvores para realizar a etapa de reconhecimento e interpretação. Para realizar a programação foi utilizada a linguagem C no ambiente C++ Builder 3.0.

1.4. Estrutura do Trabalho

No Capítulo 2, é apresentado um breve relato de como surgiu o Braille, e, também, as "regras" que constituem sua simbologia, além de apresentar os equipamentos com o quais textos Braille podem ser produzidos.

Informações sobre o *scanner*, equipamento digitalizador utilizado, são mostradas no Capítulo 3. São apresentados os tipos de imagens por eles geradas, alguns dos formatos de imagem utilizados para salvar os arquivos resultantes, as características destes arquivos, além da justificativa da utilização do formato BMP neste trabalho.

A parte teórica das técnicas de processamento de imagens utilizado, abrangendo principalmente o pré-processamento, onde essas técnicas foram mais relevantes, forma o conteúdo do Capítulo 4. No Capítulo 5, também foi abordado o processamento da imagem, mas ressaltando os aspectos construtivos do programa, e os algoritmos utilizados.

Os resultados obtidos, comparação das traduções de diversos textos, da utilização de vários *scanners*, das resoluções utilizadas ou digitalizadas com rotação indesejada, é o que traz o Capítulo 6. No Capítulo 7 tem-se a conclusão deste trabalho.

Capítulo 2 – O Sistema Braille

Este capítulo tem por objetivo estabelecer um primeiro contato com o sistema Braille, desde suas “regras” de elaboração até os instrumentos usados para sua escrita. O sistema Braille é usado por pessoas cegas para leitura e escrita. Com ele é possível representar todos os caracteres alfanuméricos utilizados na escrita corrente. A cela Braille representa um caracter, sendo que no Brasil ela é composta por seis pontos e em alguns países também se utilizam celas com oito pontos. Como o objetivo desta dissertação é traduzir textos escritos em Braille no Brasil, serão consideradas somente celas constituídas de seis pontos, distribuídos em duas colunas e três linhas.

2.1 Uma Breve História do Braille

Louis Braille nasceu no interior da França em 1809. Aos três anos feriu um olho brincando com um instrumento pontiagudo na loja de artefatos de couro do pai. Em consequência do socorro inadequado que recebeu ficou cego aos cinco anos.

Aos dez anos foi estudar em Paris, na Instituição Real para jovens cegos, onde teve inclusive aulas de música, tornando-se um excelente pianista. No sistema primitivo de leitura em que Braille aprendeu a ler, as vinte e seis letras do alfabeto eram ampliadas e tinham várias polegadas de abertura e de largura. Louis Braille, através de um artigo no jornal que um amigo lia para ele, ficou sabendo de um método alternativo de leitura para cegos. Tal método foi criado por Charles Barbier de la Serre, Capitão de Artilharia do exército de Louis XIII. Barbier elaborou um sistema de escrever que ele podia usar no escuro para fins militares, chamando-o de “escrita noturna”, o capitão usava pontos e traços em alto relevo. Barbier, então, transformou-o em um sistema de escrita para cegos que denominou *Grafica sonona*. Apesar deste sistema ser complicado, foi adotado como um método auxiliar de ensino.

Braille se entusiasmou com o sistema do Capitão Barbier e logo passou a dominá-lo. A escrita era possível com o uso de uma régua guia e de um estilete. Mas, devido às dificuldades do sistema de Barbier, Louis Braille se dedicou a inventar um sistema mais simples.

Aos quinze anos de idade inventou o alfabeto Braille, um sistema semelhante ao atual, com uma cela de seis buracos dentro de um pequeno espaço, com a qual era possível fazer sessenta e três combinações diferentes.

Já professor do Instituto Real para Jovens Cegos, Braille apresentou sua invenção para o Diretor que o autorizou a experimentá-lo no Instituto. Mais tarde, Braille aplicou seu sistema à notação musical.

No ano de 1852, Louis Braille faleceu vítima de tuberculose.

Em 1854, o alfabeto Braille foi adaptado pela escola de cegos de Paris e difundido por todo o mundo (A História de Louis Braille, 2001).

Também no ano de 1854, em 17 de setembro, foi criado no Brasil, pelo Imperador D. Pedro II, o Instituto Benjamim Constant (Rio de Janeiro – RJ) através do Decreto Imperial nº 1428, tendo sido inaugurado solenemente no dia 17 de setembro do mesmo ano, na presença do Imperador, da Imperatriz e de todo o Ministério, com o nome de Imperial Instituto dos Meninos Cegos. Este foi o primeiro passo concreto no Brasil para garantir ao cego o direito à cidadania. Foi também a primeira instituição de Educação Especial da América Latina (FUNDAÇÃO HILTON ROCHA, 1987).

2.2 O Sistema Braille

O sistema Braille é um sistema de leitura e escrita tátil que consta de seis pontos em relevo, dispostos em duas colunas de três pontos. Os seis pontos formam a “cela Braille” (O SISTEMA BRAILLE, 2002).

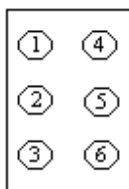


Figura 2.1. Representação da forma de leitura da Cella Braille.

A diferente disposição desses seis pontos permite a formação de sessenta e três combinações ou símbolos Braille. As dez primeiras letras do alfabeto são formadas pelas diversas combinações possíveis dos quatro pontos superiores (1, 2, 4 e 5), que compõem a primeira linha:

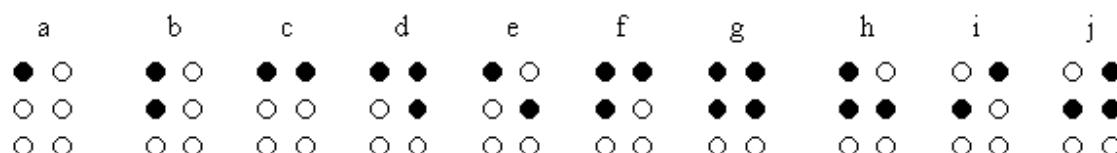


Figura 2.2. Representação das primeiras letras do alfabeto Braille. Observação: os pontos escuros representam o que estará em relevo para percepção tátil.

As dez letras seguintes, chamadas segundas linhas, ou seja, k, l, m, n, o, p, q, r, s e t, são formadas pela combinação das dez primeiras (a, b, c, d, e, f, g, h, i e j) respectivamente, com o ponto 3 da cela Braille. Por exemplo, a letra ‘o’ é representada pelo símbolo da letra ‘e’ acrescida do ponto 3.



Figura 2.3. Exemplo de duas letras do alfabeto Braille.

A terceira linha é formada pelo acréscimo dos pontos 3 e 6 às combinações da primeira linha. Mas esta linha tem algumas particularidades a serem observadas. A letra ‘w’ só foi acrescida ao alfabeto de símbolos Braille após doze anos de sua invenção, para atender às necessidades da língua inglesa. A sequência dos cinco primeiros símbolos é: u, v, x, y e z, sendo os outros cinco símbolos formados pelo ç e por letras acentuadas: ç, é, á, è e ú.



Figura 2.4. Em (1) exemplo das letras “a” e “u”. Em (2) o exemplo de “f” e “ç”.

Cada letra acentuada, â, ê, ì, ô, ù, à, ï, ü, ã, e o w, é formada pela combinação de uma das dez primeiras letras do alfabeto romano acrescida do ponto 6 da cela Braille.

Os mesmos símbolos da primeira linha do alfabeto Braille, na mesma ordem, assumem características de valores numéricos de 1 a 0, quando são precedidas pelo sinal de número, formado pelos pontos 3, 4, 5 e 6.

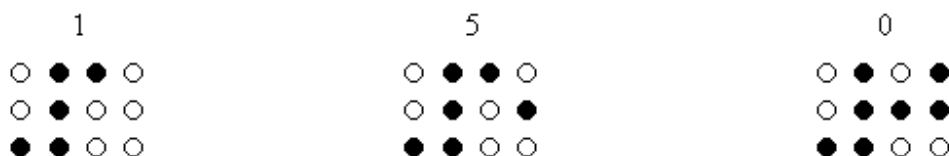


Figura 2.5. Representação Braille dos números 1, 5 e 0.

Porém, quando se escreve um número formado por dois ou mais algarismos, não é necessário repetir o sinal de número.

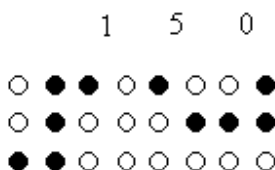


Figura 2.6. Exemplo da formação do número 150 em Braille.

Os sinais de pontuação de uso internacional são representados como as dez primeiras letras do alfabeto romano, mas os pontos em relevo não são mais os quatro pontos superiores da cela, e sim os quatro inferiores (2, 3, 5 e 6). A representação, então, fica como na Figura 2.7:

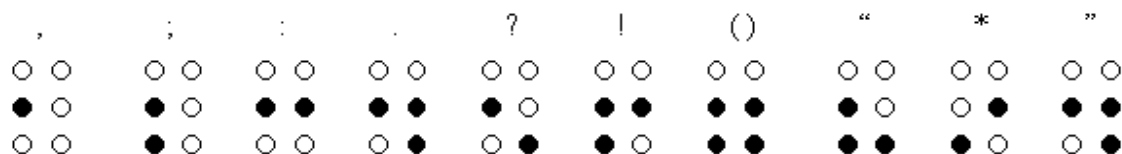


Figura 2.7. Sinais de pontuação em Braille.

Foram vistos até agora vinte e seis sinais para o alfabeto romano, dez para os sinais de pontuação de uso internacional, e dezessete sinais para algumas necessidades do idioma como o ç e o ï, num total de cinquenta e três sinais das sessenta e três combinações possíveis, além dos sinais utilizados para representação numérica. São utilizadas mais dez combinações para completar o alfabeto Braille, vistas na Figura 2.8:

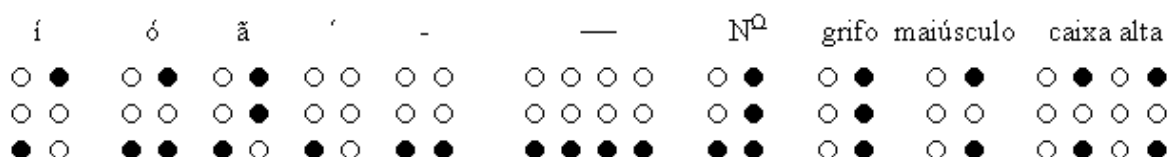


Figura 2.8. Representação de mais alguns símbolos do alfabeto Braille.

Alfabeto Braille

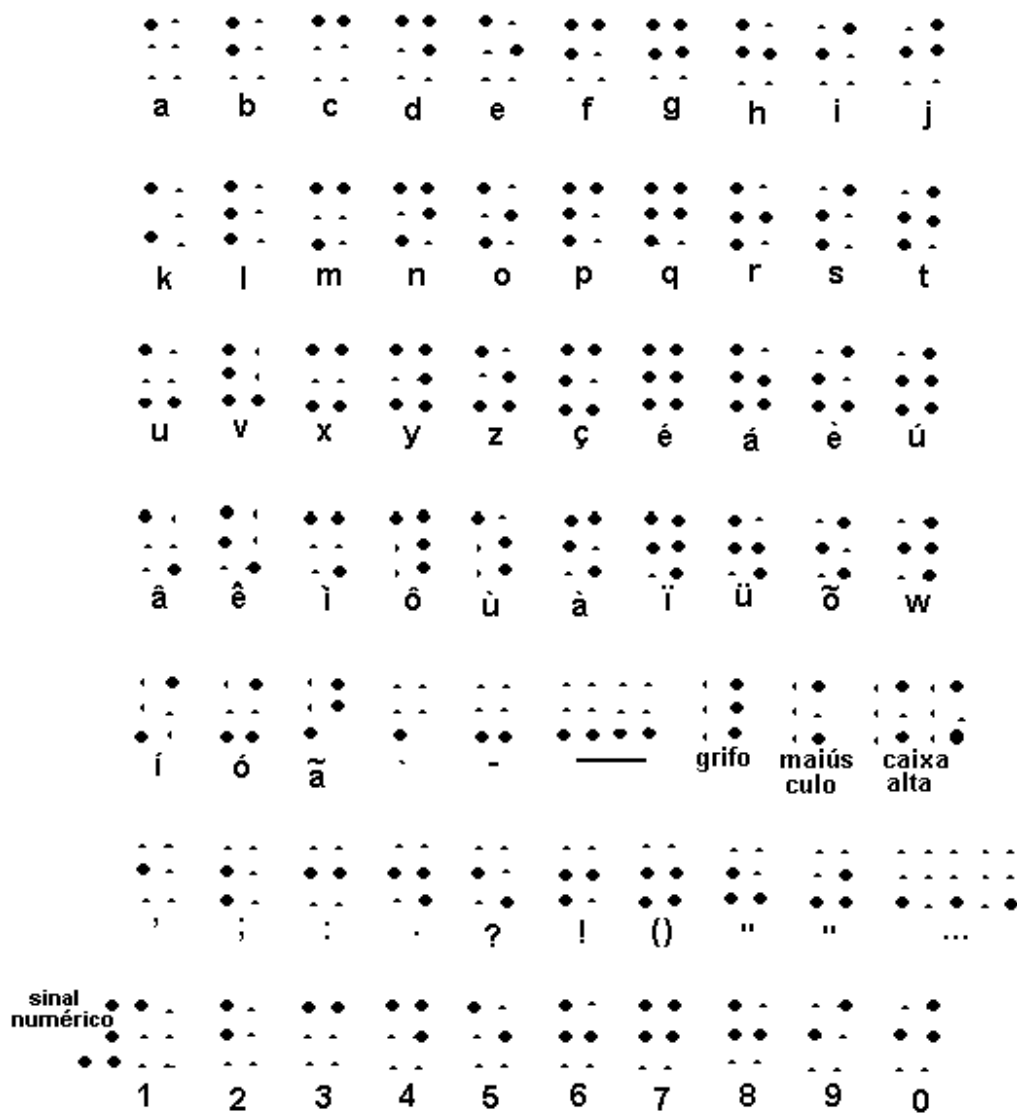


Figura 2.9. Alfabeto Braille (Extraído de FUNDAÇÃO HILTON ROCHA, 1987).

A representação de alguns símbolos matemáticos (Nascimento, Bruno, Rímoli e Cordeiro, 1987) é mostrada na Figura 2.10.

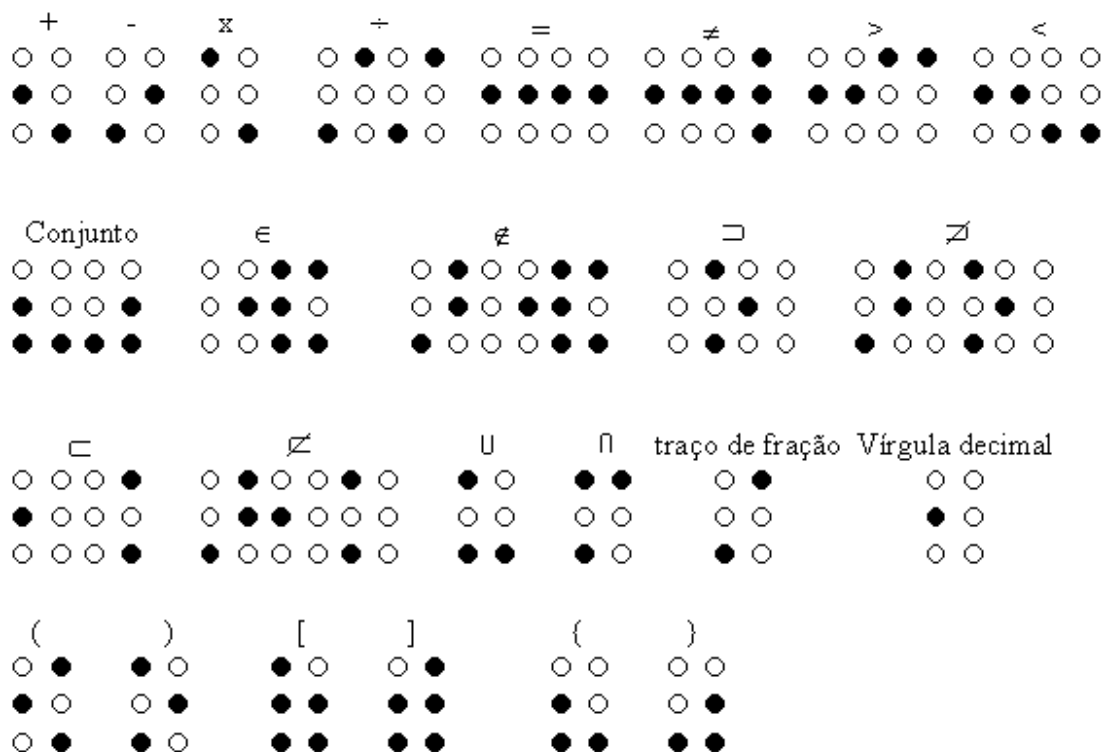


Figura 2.10. Símbolos matemáticos (Extraído de Nascimento, Bruno, Rímoli e Cordeiro, 1987).

Portanto, além do alfabeto Braille pode se utilizar as combinações para as notações matemáticas, musicais, científicas como códigos especiais. Nesta dissertação serão usados, somente, os caracteres alfanuméricos.

O código Braille possui três graus, sendo que o grau um é empregado por extenso, escrevendo-se a palavra inteira. No grau dois utiliza-se a forma abreviada, representando grupos de letras que são encontradas nas palavras de uso corrente. E o grau três também utiliza abreviações, só que mais complexas que as do grau dois. O programa de tradução vai considerar a escrita somente em grau um, ou seja, por extenso.

2.3 Instrumentos de Escrita Braille

São utilizados vários instrumentos de apoio à escrita Braille, alguns obtidos com o desenvolvimento tecnológico; diferentes aparelhos destinados a atingir o mesmo objetivo, embora por meio diferente, estão disponíveis no mercado. Alguns equipamentos serão mostrados a seguir.

2.3.1 Reglete e punção

A reglete consiste essencialmente de duas placas de metal, plástico ou madeira, fixadas em uma das extremidades através de dobradiças, para que se possa introduzir o papel entre as placas (Figura 2.12). Existe reglete de bolso e de mesa.

A placa superior da reglete funciona como uma régua possuindo janelas correspondentes às celas Braille. Diretamente sob cada janela, a placa inferior possui a configuração da cela Braille em baixo relevo (FUNDAÇÃO HILTON ROCHA, 1987).

Não existe uma padronização na confecção das regletes. Existem regletes de vários materiais e formas, mas, os pontos obtidos na folha devem possuir uma relação de distância com poucas variações através do uso deste material.

A escrita em Braille com o uso da reglete é feita da direita para a esquerda (isto é, a folha é pressionada no verso), mas a leitura é feita normalmente da esquerda para a direita. Os pontos ficam desta forma:

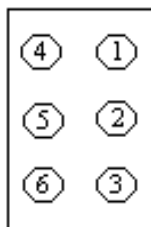


Figura 2.11: Representação da escrita de uma Cela Braille.

O punção é uma espécie de agulha grossa que é utilizado para se marcar o papel preso na reglete, ponto por ponto (FUNDAÇÃO HILTON ROCHA, 1987).

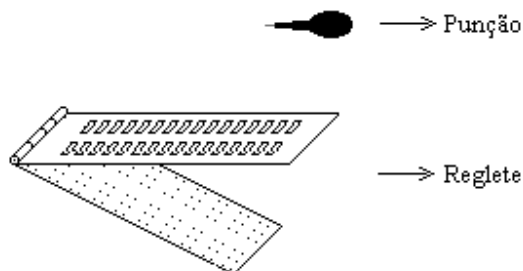


Figura 2.12. Figura ilustrativa do reglete de bolso e do punção.

2.3.2 Máquina de Escrever Braille

Além da reglete, o Braille pode ser produzido através de máquinas especiais de datilografia, de sete teclas: cada tecla corresponde a um ponto e ao espaço (6 teclas, uma para cada ponto da cela, além da tecla de espaço). O Braille, neste caso, é produzido da esquerda para a direita podendo ser lido sem a retirada do papel da máquina (FUNDAÇÃO HILTON ROCHA, 1987).

O primeiro protótipo foi desenvolvido pelo Prof. David Abraham, em 1939. Seu principal fabricante é a Perkins School for the Blind, localizada em Massachussets nos Estados Unidos.

Através de um acordo de cooperação internacional firmado com a Perkins, a LARAMARA (Associação Brasileira de Assistência ao Deficiente Visual) inaugurou em abril de 1999 a primeira fábrica de máquinas de escrever em Braille da América Latina.

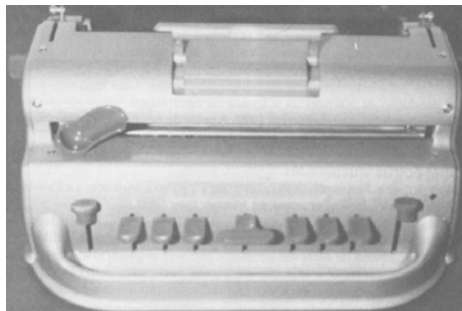


Figura 2.13: Máquina Braille (Extraído de FUNDAÇÃO HILTON ROCHA, 1987).

2.3.3 Impressora Braille

Para a produção de livros existem as impressoras Braille. O processo de escrita desta máquina permite a impressão do Braille nas duas faces do papel: os pontos são dispostos de tal forma que impressos de um lado não coincidam com pontos da outra face, reduzindo desta forma, o volume dos livros transcritos no sistema Braille.

A impressora Braille utiliza programas transcritores Braille. Estes programas executam a transcrição de textos escritos no sistema óptico (sistema de escrita comum), armazenados em computadores, para o sistema Braille, disponibilizando-os para serem impressos (Carvalho, 2001).

2.4 Leitura Braille

A leitura é feita da esquerda para a direita na maioria das vezes com o dedo indicador, mas também é usado o dedo médio ou anular.

Alguns leitores conseguem ler com as duas mãos outros só com uma. Em média, lêem cento e quatro palavras por minuto. Há os que conseguem ler cento e vinte e cinco palavras por minuto usando uma só mão, e outros até duzentas e cinquenta palavras por minuto usando as duas mãos (O SISTEMA BRAILLE, 2002).

2.5 Alguns Softwares de Auxílio ao Cego

2.5.1 Softwares Leitores de Tela de Computador

Segundo Carvalho (2001), os leitores de tela são *softwares* que acessam os textos armazenados no computador e os enviam aos sintetizadores de voz, efetuando um processo padronizado de conversão denominado TSC (Text-to-Speech Conversion).

Com o auxílio dos leitores de tela/sintetizadores de voz, estes *softwares* sonoros podem vocalizar cada letra que está sendo digitada naquele momento, através de uma configuração pessoal. Caso não queira escutar letra a letra pode-se desconfigurar tal modo, digitar tudo e ir corrigindo os erros depois, “lendo” tudo de uma vez, e parando nos erros audíveis. Tais *softwares* ainda podem ler, além da tela inteira e letra por letra, palavra por palavra ou linha por linha.

2.5.2 Reconhecedores de Voz

O *software* de reconhecimento de fala trabalha desmontando o som em unidades menores chamadas fonemas e analisando depois as palavras, contexto, frases, etc. É um *software* bastante complexo e de grande ajuda para os cegos, pois permite a substituição do teclado de um computador, para a introdução de dados por comando de voz.

Eles podem ser ajustados para reconhecerem uma grande variedade de comandos de um usuário em particular, mas são menos eficientes em receber comandos de mais de um usuário. Quando são ajustados para reconhecer múltiplos usuários, o número de comandos que passam a “entender” com segurança é menor do que aqueles disponíveis quando estavam ajustados para o reconhecimento de comandos de um usuário específico (Carvalho, 2001).

2.5.3 Tradução de Caracteres Alfanuméricos para Braille

Estes softwares propiciam a escrita Braille através da impressora. O texto em caracteres alfanuméricos pode ser armazenado na memória do computador, faz-se a tradução para os caracteres Braille e, então, o software faz a interação com a impressora Braille, podendo com isto, gerar textos Braille com muito mais rapidez, que os métodos tradicionais (reglete e punção, e máquina Braille).

2.5.4 Optical Braille Recognition (OBR)

Este *software* está disponível para compra no mercado internacional, e faz a tradução do Braille para caracteres alfanuméricos (não traduz para o português do Brasil). É capaz de traduzir folhas impressas dos dois lados, e está na versão 3.6.

É compatível somente com algumas marcas e modelos de *scanners*. Isto, porque sua interface faz comunicação direta com tais equipamentos (alguns modelos das marcas Hewlett Packard, Microtek, Mustek, Visionner e UMAX).

2.6 A Deficiência Visual no Brasil

São ditas deficientes visuais as pessoas que possuem visão subnormal ou totalmente cegas. Os dados fornecidos pelo MEC (Ministério de Educação e Cultura) e apresentados nas tabelas 2.1 e 2.2 mostram em números a situação dos deficientes físicos e mentais levantada pelo Censo de 1999 (Secretaria de Educação Especial).

A visão subnormal é definida como a capacidade de visão situada entre 20/40 e 20/200 após a correção que uma pessoa possui. Uma pessoa com visão de 20/200 é aquela que consegue ver algo a 20 pés (6,096 m) de distância da mesma maneira que uma outra pessoa normal consegue ver a 200 pés (60,96 m) de distância. Uma pessoa considerada com visão normal possui a capacidade de visão de 20/20 (Carvalho, 2001).

Uma pessoa é classificada como legalmente cega quando sua acuidade visual é 20/200 ou pior após a correção, ou quando seu campo de visão é menor que 20 graus (Carvalho, 2001).

De acordo com a Tabela 2.1, 1,06 % da população brasileira entre 0 e 19 anos possui algum tipo de deficiência, e dentre estes 2,9 % são cegos. Nesta faixa de idade, normalmente, os deficientes visuais estão freqüentando a escola (desde a creche até o ensino médio), e são alfabetizados em Braille. Para facilitar sua inclusão no ambiente escolar, é necessário o uso de meios tecnológicos disponíveis.

Tabela 2.1. Dados do Censo 1999 sobre a população com necessidades especiais por grupo de idade (Secretaria de Educação Especial, 2002).

População Residente, por tipo de Deficiência Segundo a Situação do Domicílio e os Grupos de Idade												
Grupos de Idade	População Total	Total Deficientes	Cegueira	Surdez	Total Físicos	Hemiplegia	Paraplegia	Tetraplegia	Falta de Membro	Mental	Mais de Uma	Sem Declaração
Total	146.815.795	2.199.034	45.852	173.582	602.352	208.565	201.617	46.989	145.181	658.915	87.084	531.249
0-4 anos	16.532.398	124.008	4.932	5.391	12.424	2.390	5.076	2.286	2.672	14.895	3.664	82.702
5-14 anos	34.472.777	371.635	10.098	28.820	50.333	12.684	20.653	9.672	7.324	119.650	15.706	147.028
15-19 anos	15.012.515	201.117	5.285	14.838	38.918	11.479	16.938	4.396	6.105	76.265	8.955	56.856
20-24 anos	13.558.380	187.512	6.082	12.460	38.289	10.817	16.204	3.348	7.920	75.764	7.496	47.421
25-29 anos	12.629.050	172.836	6.200	8.548	41.386	11.833	16.449	2.804	10.300	70.236	5.977	40.489
30-59 anos	43.934.563	714.303	42.752	47.773	233.577	84.486	67.723	11.033	70.335	241.457	24.675	124.069
60-69 anos	6.411.777	187.211	22.053	19.482	85.122	38.253	21.053	4.235	21.581	34.300	7.708	18.546
70-79 anos	3.156.268	146.336	26.157	19.974	65.161	26.132	20.077	5.156	13.796	18.415	6.832	9.797
80 ou mais	1.108.067	94.076	22.293	16.296	37.142	10.491	17.444	4.059	5.148	7.933	6.071	4.341

A Tabela 2.2 mostra que a maior parte das matrículas da Educação Especial de deficientes visuais se concentra no Ensino Fundamental, 11.924 matrículas, que correspondem a 64% do total de matrículas (18.629). Pode-se estimar através destes números que o aluno deficiente visual deve encontrar diversas barreiras para a continuação de seus estudos; além do mais, a preocupação com a educação e a inclusão do deficiente visual na sociedade é bastante recente.

Tabela 2.2. Dados do censo de 1999, com o número de matrículas por tipo de necessidade especial em atividade escolar (Secretaria de Educação Especial, 2002).

Matrículas da Educação Especial por Tipo de Necessidade Especial e Nível de Ensino – Censo 1999 (*)									
Nível De Ensino	Deficiências					Altas Habilidades Superdotados	Condutas Típicas	Outras Necessidades	Total
	Visual	Auditiva	Física	Mental	Múltipla				
Creche	770	1806	2346	12180	8334	13	493	3044	28986
Pré-escola	1404	6618	2917	39312	11385	102	1734	3410	66882
Fundamental	11924	31825	8151	101968	14607	863	4786	21391	195515
Médio	876	899	495	475	236	40	26	143	3190
Jovens/Adultos	751	2228	1082	6074	835	17	84	264	11335
Outros	2904	4434	2342	37987	11348	193	2100	6913	68221
Total	18629	47810	17333	197996	46745	1228	9223	9223	374129

O trabalho desenvolvido, nesta dissertação de mestrado, poderá facilitar a comunicação destes alunos alfabetizados em Braille com professores, colegas de escola, familiares e amigos, que em sua maioria desconhecem o sistema Braille de leitura e escrita.

Capítulo 3 –Formato de Imagens

Os *scanners* são equipamentos que digitalizam imagens a partir de originais em papel ou em transparências. Eles convertem luz em informação digital através de pequenos componentes eletrônicos chamados CCD (*Charged Couple Device*), PMT (*Photomultiplier Tubes*) e CIS (*Contact Image Sensors*). A informação, então, é enviada a um computador.

A imagem digitalizada é dividida em grades pelo *scanner* que usa uma coluna de componentes para captar quanta luz é refletida em cada locação. Cada célula em uma grade é chamada de *picture element*, ou *pixel*. Os *scanners* se diferem pela quantidade de *pixels* que cada um tem e pelo modo que o objeto é digitalizado.

Cada *scanner* utiliza um *driver* específico, chamado *driver* Twain, para se comunicar com o *software* de tratamento de imagem. Este *driver* é fornecido junto com o equipamento. Quando o computador tem todos os dados relativos à imagem, ele constrói um arquivo que representa a imagem na forma digital.

Existem algoritmos específicos na elaboração dos arquivos de imagem que fornecem os chamados formatos de imagem. Esses algoritmos podem ser de domínio público ou não. Equipamentos que capturam imagem podem trabalhar com a imagem vetorial, ou, com a imagem por “varredura”, conhecida como *raster*. Como nessa dissertação foram utilizados os *scanners* para digitalizar as imagens Braille, e eles fornecem imagens *raster*, só serão detalhados alguns dos diferentes formatos desse tipo de imagens.

3.1. Descrição de Scanners

Os *scanners* possuem componentes foto-sensíveis que convertem a intensidade de luz refletida em um sinal elétrico. Pode-se dizer, simplificadaamente, que a variação de tensão é transformada numa matriz de *pixels*, sendo que cada *pixel* é empregado para representar a intensidade de luz.

Pixel é o menor elemento que compõe a imagem. Ele é representado em uma matriz com o valor do nível de intensidade que possui. A forma e a cor de uma imagem podem ser representadas numa matriz bidimensional de pontos. O *scanner* pode produzir imagens em branco e preto, em escalas de cinza, ou em cores.

3.1.1. Tipos de Scanners

Existem tipos diferentes de *scanners*, sendo que os mais utilizados nas instituições de auxílio aos cegos, e para uso doméstico, são os chamados *scanners* de mesa. Portanto, eles foram utilizados nessa dissertação para digitalizar as imagens Braille.

- *Scanners* de Mesa

Quando o original for opaco, como no caso dos utilizados nessa dissertação, o mesmo é colocado em uma bandeja de vidro, enquanto o cabeçote move-se por baixo dela. Alguns *scanners* de mesa vêm com unidades de transparência (opcionais) que lhe permitem digitalizar *slides*. Trata-se de uma cobertura de *scanner* que difunde luz uniformemente pela mídia transparente que se adapta no lugar da tampa do scanner.

Os *scanners* de mesa (Figura 3.1) podem digitalizar folhas de vários tamanhos, mas, a imagem desejada deve estar contida na bandeja de vidro. Além disso, podem ser digitalizados até pequenos objetos de três dimensões; como no caso da escrita Braille, cujos pontos são em alto relevo.

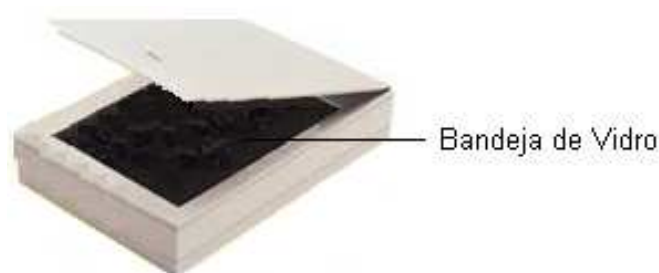


Figura 3.1. *Scanner* de mesa

3.2. Especificação

Cada fabricante de *scanner* de mesa possui diferentes modelos. Os modelos de *scanners* de um ou mais fabricantes diferem entre si quanto à técnica de dirigir a luz para as páginas, captar a luz refletida, e convertê-la de um sinal elétrico para uma forma digital. Diferentes aspectos podem ser observados:

- Sensores Utilizados nos *Scanners*
- Resolução
- Profundidade de Bit (*Bit-depth*)
- Medida Dinâmica (*Dynamic range*)

3.2.1. Sensores Utilizados nos *Scanners*

Três diferentes sensores são utilizados nos *scanners*. Na maioria deles, existe um componente que é o dispositivo de acoplamento por carga – CCD (*Charged Couple Device*). Outros utilizam os sensores de contato de imagem, ou CIS (*Contact Image Sensors*), e também há os que utilizam os tubos fotomultiplicadores, ou PMT (*Photomultiplier Tubes*).

O PMT é um sensor mais sofisticado que o CIS ou o CCD; são utilizados em *scanners* de tambor. O sensor do tipo CIS contém uma coluna simples de sensores baseado em LEDs que são colocados muito próximos ao original a ser digitalizado, e sua utilização resulta em equipamentos mais leves e finos. Aqui, detalhar-se-á, um pouco mais, o equipamento de acoplamento de carga (SCANNERS, 2002).

- CCD (Dispositivo de Acoplamento de Carga)

O CCD tem como base pastilhas semicondutoras com determinado número de recipientes capazes de armazenar carga elétrica. Portanto, o CCD equívale a uma memória analógica, cujos dados são acessados serialmente através de transferência de carga entre os recipientes. A

utilização de material sensível a luz no CCD faz com que a carga armazenada em cada recipiente seja proporcional à intensidade luminosa incidente sobre o mesmo (Marques Filho e Neto Vieira, 1999).

A matriz CCD varia conforme a resolução que o dispositivo é capaz de alcançar e o tamanho máximo da superfície que ele tem condições de digitalizar. Representada por linhas de sensores em um *chip* CCD que são chamadas linha de matriz (White, 1993).

Uma imagem digital é armazenada como uma matriz de dados de duas dimensões contendo *pixels* endereçados por coordenadas x, y (ou linha, coluna). A fonte de luz ilumina uma fina faixa horizontal da imagem, a luz refletida nesta linha é capturada pela matriz CCD e cada elemento que a compõe produz as tensões analógicas correspondentes (o preto corresponde à tensão zero e o branco, à tensão máxima). Depois, esta informação é convertida, através de um conversor analógico-digital (Netravali e Haskell, 1988).

A velocidade com que o cabeçote se desloca para capturar cada uma das linhas de varredura determina a taxa de amostragem na direção y. A largura da superfície digitalizada dividida pelo número de elementos CCD instalados na matriz determina a taxa de amostragem na direção x, que é usada como referência para medir a taxa de amostragem óptica do dispositivo, normalmente expressa por pontos por polegada (Foley, van Dam, Feiner e Hughes, 1990).

3.2.2. Resolução

A resolução não se refere apenas ao grau com que o *scanner* é capaz de capturar os detalhes, ou seja, a faixa de frequências na luz refletida sobre o objeto. Além de se atentar para a matriz de elementos CCD e o conversor A/D, na observação da taxa de amostragem óptica, outros fatores como a qualidade das lentes e dos filtros empregados, o movimento dos elementos mecânicos que suportam todo o sistema óptico, as vibrações produzidas, e até as condições de umidade e temperatura, devem ser observados. (Foley, van Dam, Feiner e Hughes, 1990).

Um *scanner* com uma taxa de amostragem de 600 dpi (pontos por polegada), mas com um sistema óptico de baixa qualidade, pode ser inferior a outro que tenha uma taxa de amostragem

de 400 dpi. Uma resolução ótima é obtida tendo a uma taxa de amostragem elevada somada a uma alta qualidade no sistema óptico.

Há duas maneiras de se medir a resolução: a óptica e a interpolada. A resolução óptica é determinada pelo número de *pixels* que se tem no *scanner*. A interpolação é um processo aplicado a sinais, de forma geral, e a imagens, de forma particular, depois de se obter a amostra a uma taxa determinada e é realizada através de algoritmos.

3.2.3. Profundidade de Bit

Profundidade de *bit* ou *bit depth* é o número de bits usados para representar cada *pixel* em uma imagem, determinando sua cor ou tonalidade. Ou seja, é o número de bits por *pixel* alocado para o armazenamento de informações sobre as cores indexadas em um arquivo gráfico; quanto mais bits uma imagem tiver, mais informações ela pode conter.

Uma imagem binária é representada por *pixels* de 1 bit cada, que podem representar 2 tons (tipicamente preto e branco). Usualmente, determina-se 0 para preto e 1 para branco. E uma imagem em nível de cinza utiliza 2 a 8 bits para representar suas tonalidades. Com 2 bits têm-se 4 combinações possíveis, já com 8 bits a possibilidade de se ter 256 (2^8) níveis de cinza diferentes para cada *pixel* (MOVING THEORY INTO PRATICE – DIGITAL IMAGE TUTORIAL, 2002)

3.2.4. Medida Dinâmica

É a diferença entre os valores mínimo e máximo das tonalidades que um *scanner* pode registrar. Esta medida é expressa em uma escala de branco a aproximadamente preto, este número diz quanto o *scanner* é capaz de distinguir nos seus tons. (MOVING THEORY INTO PRATICE – DIGITAL IMAGE TUTORIAL, 2002).

3.3. *Softwares*

Um pacote de *softwares* vem junto aos *scanners*. Os *drivers* e os *softwares* de editoração de imagens são muito importantes no processo de digitalização da imagem.

- *Driver Twain*

Twain é uma padronização que permite que se utilize o *scanner* a partir de qualquer aplicativo gráfico. Em outras palavras, as imagens podem ser capturadas dentro do aplicativo gráfico preferido.

Quando o aplicativo gráfico solicita algum comando ao *scanner*, o *driver Twain* entra em ação. Na verdade, o driver é um subprograma responsável pela captura da imagem. Não importa de qual programa se utiliza o *scanner*, o *driver Twain* estará sempre atuando. O *scanner* será utilizado sempre da mesma maneira, independente do aplicativo gráfico utilizado para realizar a captura das imagens (Torres, 2001).

- *Softwares* de Editoração de Imagens

Estes softwares de tratamento de imagem fazem parte do pacote que vem com o *scanner*, ou mesmo podem ser obtidos separadamente. É possível, com estes softwares, a escolha do algoritmo a ser utilizado, ou seja, escolher qual a extensão com que o arquivo da imagem será salvo. Seja esta imagem de qualidade fotográfica, ou um documento com texto alfanumérico ou em Braille.

3.4. **Formatos de Imagem**

Quando imagens são capturadas eletronicamente por *scanners* ficam armazenadas em arquivos. Diferentes fabricantes de equipamentos digitais e de programas de computador

desenvolveram uma grande quantidade de formatos de arquivos. Os formatos de arquivos descrevem como as imagens são organizadas dentro do disco ou da memória do computador.

Os problemas decorrentes de se armazenar e transmitir imagens em vários formatos aparecem quando é necessário manipular as mesmas. Existem dois tipos de imagens: imagens vetoriais e “varredura” (*Raster Image Format*). Será detalhado, somente, este último formato, por ser o utilizado pelos *scanners* de mesa, como visto anteriormente.

Os *pixels* descrevem imagens, ponto a ponto, com cor ou nível de cinza em cada ponto, ou seja, a informação gráfica é descrita como um conjunto de *pixels*, normalmente organizados da esquerda para a direita e de cima para baixo. Na cor de cada *pixel* está toda a informação. Nas imagens reais, cada *pixel*, portanto, tem uma localização específica e uma cor atribuída a ele.

Não há um padrão ISO (*International Organization for Standardization*) para transferência e armazenamento de informação descritiva de imagem *raster*, mas há numerosos formatos desenvolvidos por fabricantes e laboratórios de pesquisa. Apesar de não serem padronizados, eles têm a mesma forma.

O formato de imagem *raster* é feito de duas partes (Post e Hin, 1992):

- a cabeça que descreve todos os parâmetros de imagem tais como: um rótulo de identificação (GIF87a para GIF, P6 para PPM), o tamanho, a posição, o número de canais, o número de cores por canal, a presença do canal *alpha*¹, um mapa de cor global, a cor de fundo e comentários;
- e o corpo que é feito de dados de imagem linear. Cada *pixel* é definido para um endereço dentro de uma tabela LUT (*look-up table*) ou diretamente dos valores das três cores (isto é, componentes RGB). Então, estes dados podem ser comprimidos usando algoritmos diferentes, esta imagem é dita indexada.

¹É um canal adicional, invisível na imagem final, que usualmente é empregado para armazenar o recorte de transparência da imagem ou máscaras de seleções.

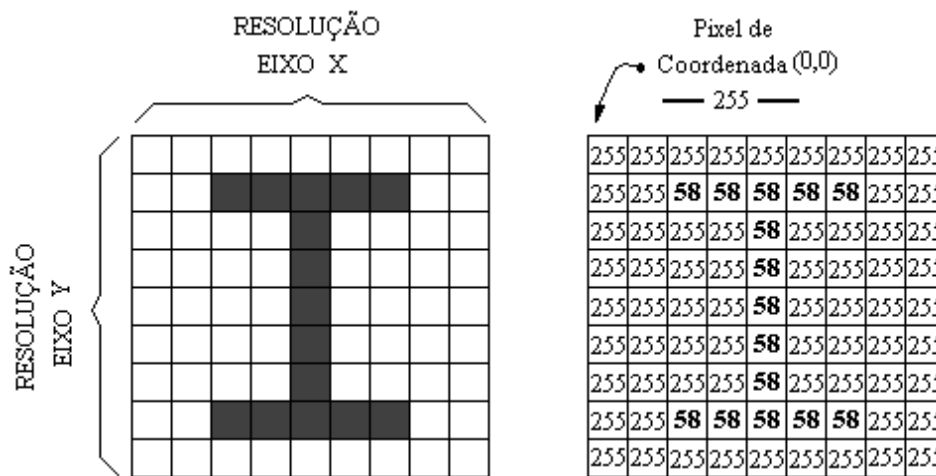


Figura 3.2. Representação do mapa de *pixel*, tendo na segunda grade valores do nível de cinza em cada coordenada.

Considere um mapa de cor $l: I \rightarrow E$. Ou seja, l é uma curva definida em um intervalo I e tomando valores em um espaço de cor E . Se o dispositivo utiliza N bits para implementar o sólido de cor, então, tem-se 2^N valores distintos nesse sólido. Ao invés de endereçar diretamente pontos no sólido de cor, utilizam-se esses valores para endereçar diretamente pontos no domínio do mapa de cor l . A implementação do mapa de cor, no dispositivo gráfico, é chamado de *look-up table* ou tabela LUT (Gomes e Velho, 1994).

Considerando uma imagem colorida, isto é, com componentes RGB, sendo cada componente de cor do sistema representado por N bits, ter-se-á uma imagem com 2^{3N} valores de *pixel* possíveis (16.777.216, no caso da Figura 3.3), guardados na memória. Observando a Figura 3.3, é possível notar que utilizando a LUT, cada *pixel* será endereçado à posição da tabela onde se encontra a cor do *pixel*, o que faz com que se tenha 2^N cores no mapa (256 cores no mapa).

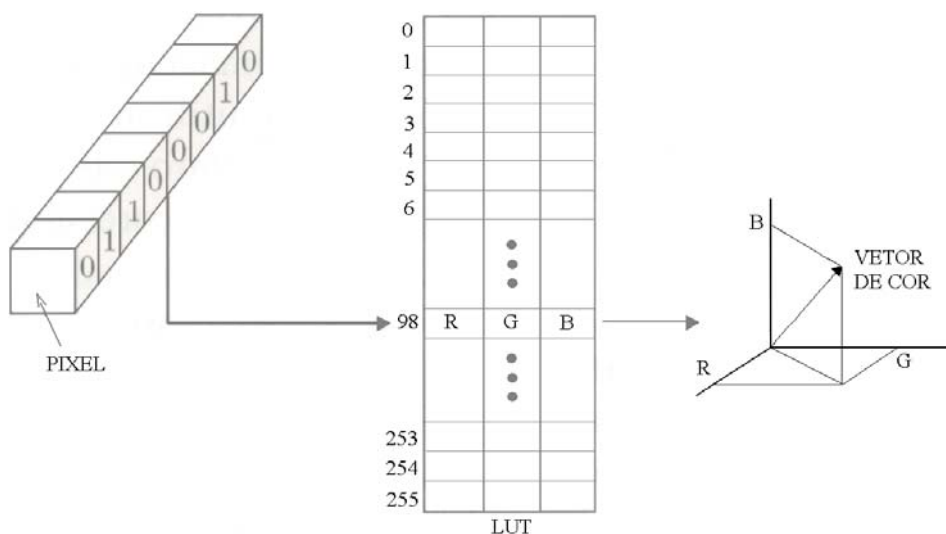


Figura 3.3. Representação da *Look-up Table* com 8 bits

Mas, é importante observar que, se o valor de N for grande, ou seja, um valor de 24 bits ou mais para cada componente de cor, a utilização da LUT não é indicada, pois a tabela seria muito grande, o que traria um gasto de memória não compensatório. Quando se endereça diretamente o espaço de cor, diz-se ter uma cor real, ou *truecolor*.

As imagens *raster* trazem um gasto de memória elevado. Existem muitos algoritmos de compressão de informação, mas nem todos são adequados à compressão de imagem. Em se tratando de algoritmos que são elaborados com este fim, é importante ressaltar que não se consegue uma uniformidade de resultados entre eles, isto é, a compressão obtida dependerá da natureza da imagem que se está trabalhando.

Quando se tem uma imagem com pouca variação de cores, como é o caso estudado nesta dissertação, tem-se, portanto, grupos de *pixels* consecutivos de mesma cor. Neste tipo de imagem pode-se utilizar algoritmos de compressão do tipo RLE (*Run Length Encoded*) que substitui estes grupos por duas unidades de informação: o número de *pixels* do grupo e o índice de cor dos *pixels*. Ou pode-se utilizar a família de algoritmos LZ, que detectam todos os *pixels* de um grupo

que não precisam ter a mesma cor, mas sim possuir uma seqüência constante de cores (1 FORMATOS DE IMAGEM, 2002).



Figura 3.4. (a) Fragmento da representação da imagem retirada da Figura 3.2 – sem compressão. (b) Fragmento da representação da imagem retirada da Figura 3.2 – com compressão RLE (redução de 22.23%, 2 em 9 *pixels*). Onde:

- -1: código de controle do RLE
- 4: número de repetições do valor 255.

Imagens digitais de qualidade fotográfica não têm uma boa compressão com os algoritmos acima citados, possuem *pixels* consecutivos que sofrem alteração de cores de forma continuada e suave. Elas necessitam de algoritmos específicos, como os baseados numa transformada que deriva da Transformada de Fourier que é a Transformada Discreta do Co-seno. Os formatos que utilizam este tipo de compressão (exemplo do formato que utiliza a norma internacional JPEG) não serão detalhados, neste trabalho, pois não se trabalhará com imagens de qualidade fotográfica.

A seguir, serão descritos alguns formatos de imagem utilizados nos *softwares* de editoração de imagem. O primeiro formato a ser descrito, que pertence a família PBM (*Portable Bitmap Utilities*), pois, este se trata de um formato simples em que se tem a possibilidade de visualização de sua estrutura.

3.4.1. Apresentação dos Formatos

- PBM (*Portable Bitmap Utilities*): Família de Utilitários de *BitMaps* Portáteis

Engloba, na verdade, mais três formatos além do PBM (imagens em preto e branco); o *Portable Greymap Utilities* (PGM), imagens em nível de cinza, o *Portable Pixmap Utilities* (PPM) para imagens coloridas, e o *Portable Anymap Utilities* (PNM) que opera sobre todos os *bitmaps* produzidos pelos outros programas.

O PBM é o formato de arquivo mais simples, que embora pertença e seja patenteado por Jef Poskanzer, o fonte e o executável estão, livremente, disponíveis na Internet. O uso deste formato como um intermediário para propósito de conversão é aconselhado por Post e Hin (1992).

Quando convertemos um arquivo gráfico de um formato para outro, tem-se o arquivo fonte (o formato corrente) e o arquivo destino (formato desejado). Trabalha-se, então, pegando um arquivo fonte e convertendo em um formato intermediário. Aquele formato intermediário é então convertido para o formato destino.

O formato PBM faz parte de um pacote de domínio público muito popular para os sistemas Unix[®], chamado PBMPLUS. Vem com uma biblioteca que inclui conversores para outros formatos e algumas ferramentas – exemplo de quantificação, filtragem e ampliação. Uma outra característica é que ele não possui algoritmo de compressão.

Por ter um elevado gasto de memória, é justificável sua utilização como um formato intermediário. As imagens podem ser salvas em outros formatos e, então, convertidas ao PBM. Faz-se o processamento dessas imagens, e de volta, se necessário, as imagens processadas retornam ao formato destino, que pode ser um formato novo, ou o mesmo formato de origem (Post e Hin, 1992).

Tabela 3.1. Exemplos de identificadores do tipo do formato

Tipo	ASCII	Binário
PBM	P1	P4
PPM	P3	P6

A definição da imagem é a seguinte: o identificador do formato (Tabela 3.1), um possível comentário (utilizando o símbolo #), a largura e tamanho da imagem, a cor máxima, o valor do componente e então, os dados da imagem feitos dos componentes RGB (caso imagem colorida) de cada ponto. Exemplo na Figura 3.5, a seguir:

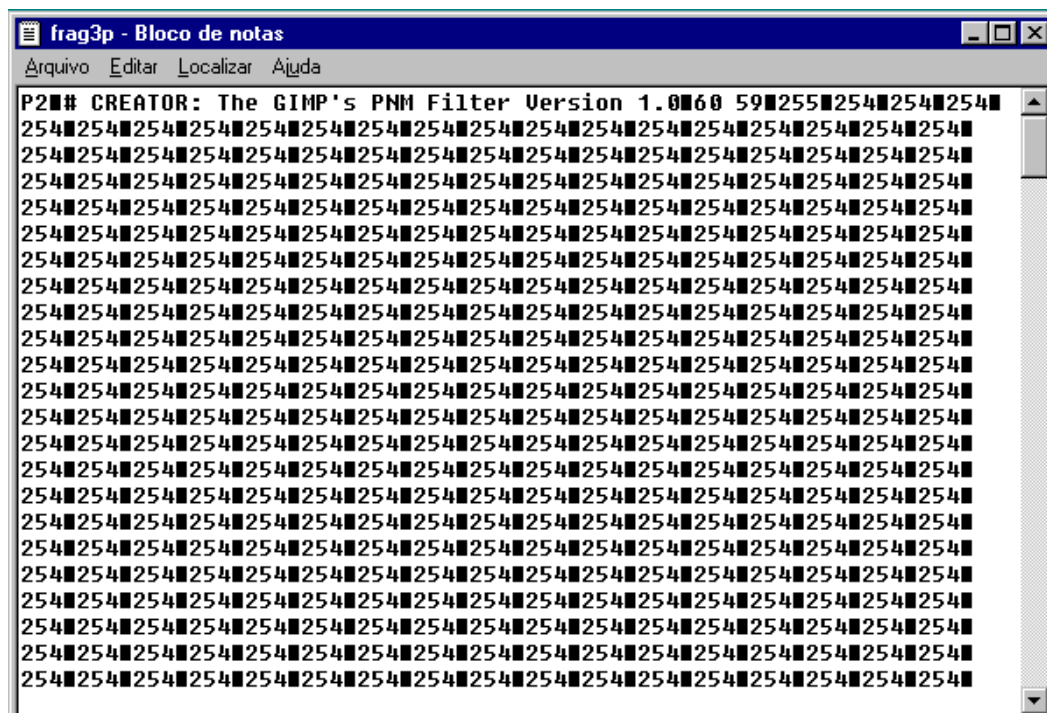


Figura 3.5. Tela de abertura da parte inicial do arquivo frag3p.pnm (imagem digitalizada da letra r em Braille), salvo em ASCII, aberta no bloco de notas do Windows 2000®.

Pode-se ver, na Figura 3.5:

- Identificador – P2
- Comentário (#) – CREATOR: THE GIMP'S PNM FILTER VERSION 1.0
- 60 59 – correspondente ao tamanho da imagem
- 255 – o valor do componente

- 254 – dados da imagem
- ■ – tecla ENTER.

Quando a representação for em ASCII, os valores correspondentes aos *pixels* serão apresentados em notação decimal e separados por espaço em branco, ou equivalentes; no caso da representação binária, armazenam os valores correspondentes ao *pixels* das imagens em caracteres (bytes) contíguos, sem separador (1 FORMATOS DE IMAGEM, 2002).

- GIF (*Graphics Interchange Format*): Formato Gráfico de Permutação

Esse formato permite armazenar e transmitir imagens com um máximo de 256 cores, definida sempre por meio de mapas de cor. Cada *pixel* de uma imagem no GIF contém um índice correspondente ao número de ordem da sua cor no mapa de cores. O conjunto dos índices que compõe cada imagem está comprimido pelo algoritmo Lempel-Zio e Welch (LZW), que é um algoritmo de compressão sem perda (1 FORMATOS DE IMAGEM, 2002).

A informação transmitida por um canal de dados GIF está organizada em blocos, denominados blocos lógicos que são: blocos de controle (por poder trabalhar com mais de uma imagem), de processamento de imagem (*image rendering*), e os blocos especiais; cujos itens vão desde o cabeçalho ao bloco descritor da imagem que, como próprio nome diz, possui os parâmetros que caracterizam a imagem, além do mapa local de cores, caso este exista.

Não tem, como o PBM, ferramentas anexas (exemplificadas quando era descrito o PBM). Também, não possui canal *alpha*, utilizado para imagens em transparências. A descompressão de dados GIF é mais lenta do que a compressão RLE, mas o espaço ocupado é muito menor.

- TIFF (*Tags Image File Format*): Formato de Arquivo de Rótulos de Imagem

Este é um dos formatos mais usados na computação gráfica. Esta família de formatos é projetada para promover trocas de dados de uma imagem digital, este formato foi projetado para diminuir os problemas encontrados nos arquivos de formato fixo.

O formato TIFF armazena, então, uma grande variedade de dados de imagem em campos de *tag* (marcas) que tem várias vantagens (Lindley, 1991):

- a) Programas aplicativos podem ignorar algumas *tags* que não compreenderem.
- b) Novas *tags* podem ser adicionadas sem prejuízo às antigas. Isto permite que este formato não se torne obsoleto.
- c) *Tags* privadas podem ser definidas para conter informações proprietárias dentro do arquivo TIFF sem afetar outros aplicativos.

O canal *alpha* (para uso em transparência) está presente no mesmo; este é um padrão do formato.

Existem muitos algoritmos de compressão, incluindo o LZW; que pode, portanto, ser utilizado opcionalmente, diminuindo muito o tamanho do arquivo sem causar perdas na qualidade da imagem.

A versatilidade do TIFF traz, também, o problema na particularidade de algumas versões, que, por isso, necessitam de *softwares* de tratamento de imagem específicos para sua utilização. Ver-se-á um exemplo deste fato, no item 3.5.2, a seguir.

- DIB (*Device Independent Bitmap*): Equipamento Independente de Mapa de Bit

Também designado por BMP – Microsoft[®]. É independente do processo empregado por cada placa gráfica e do monitor para apresentar a cor dos *pixels* das imagens. No formato DIB, as imagens são descritas por varredura ascendente das linhas, ou seja, de baixo para cima. Os *pixels* de cada linha são varridos da esquerda para a direita

Descreve imagens de 2, 16, 256 ou 2^{24} cores, respectivamente, utilizando, portanto, 1, 4, 8 ou 24 bits por *pixel*, com a utilização de mapa de cores em imagens de até 256 cores, permitindo, também a compressão de imagens com 16 ou 256 com algoritmo RLE (1 FORMATOS DE IMAGEM, 2002).

- PCX (*PC Paintbrush*)

É um dos formatos *raster* mais antigos para permitir armazenamento e padronização de imagens gráficas. Por ele ter sido criado há muitos anos, esse formato é, provavelmente, o que é aceito por mais aplicativos gráficos existentes (Lindley,1991).

Foi criado pela *ZSoft*. Comprime os dados com compressão tipo *RLE*; as imagens, neste formato, ocupam, portanto, menos espaço de armazenamento que imagens padrão BMP. A versão atual suporta até cores em 24 bits (true color).

- PNG (*Portable Network Graphics*): Redes de Trabalho Gráfico Portáteis

O formato PNG surgiu como resposta às limitações técnicas e às restrições legais derivadas dos direitos de propriedade do formato GIF, pelos direitos de propriedade do formato e do algoritmo de compressão empregado, o LZW (1 FORMATOS DE IMAGEM, 2002).

Também foram feitas outras alterações. O algoritmo de compressão, que no GIF era o LZW (não é mais de domínio público) e no PNG, passou a ser Adam7. E, nas imagens a cores passaram a ser permitido 24 ou 48 bits por *pixel*. Diferente do GIF, esse formato tem a presença do canal *alpha*, definindo máscaras de transparência globais. Além de ter como opcional, a informação sobre a correção gama (correção adequada às cores nas unidades de saída gráfica) aplicada à imagem.

3.4.2. Exemplo das Diferenças dos Formatos de Imagem

No *MGI PhotoSuite SE*, *software* que vem com o *scanner* de mesa TCÊ S440. A imagem da letra r em Braille (Figura 3.6) foi capturada e armazenada nas extensões:

- bmp - 8 e 24 bits por *pixel*
- gif - 8 bits por *pixel*
- tif - 8 e 24 bits por *pixel*
- pcx - 8 e 24 bits por *pixel*



Figura 3.6. Imagem digitalizada (59x60), arquivo de extensão bmp com valor de profundidade de bits (*Bit Depth*), 8.

Em se tratando do objetivo deste trabalho, a imagem (Figura 3.6) capturada e armazenada, nos diversos formatos acima citados, não teve alteração de qualidade. Houve, sim, quando da utilização da *toolbox* de processamento de imagem do MATLAB[®], algumas observações:

- As imagens com extensão tif (formato TIFF) não foram abertas, pois esta ferramenta não suporta o algoritmo de compressão LZW nas imagens com este formato. Como explicado, quando se descreu este formato, é um problema da particularidade de suas versões, ou seja, o TIFF, como um padrão de formato com muitas variações, pode encontrar incompatibilidade entre versões usadas em diferentes *softwares* de processamento de imagem encontrados no mercado.
- Houve também, um alerta (*warning*), quando da utilização da imagem no formato pcx com 24 bits por *pixel*. O mapa de cores não foi encontrado, mas isto, não ocasionou maiores problemas na imagem utilizada.

A seguir, encontra-se uma tabela (Tabela 3.2) que fornece algumas características dos arquivos das imagens em diferentes formatos. É possível, através de seus dados, observar como se comporta cada formato com seus diferentes valores de profundidade de (*bit depth*), e, também, as diferenças de uma extensão em relação a outra, em se tratando do tamanho do arquivo.

Tabela 3.2. Dados das características das imagens, representados pela Figura 3.6, em diversos formatos. Dados obtidos através do comando “imfinfo” do MATLAB[®].

MATLAB [®]					
FORMATO	BIT DEPTH	TIPO DE CORES	MAPA DE CORES	TIPO DE COMPRESSÃO	TAMANHO DO ARQUIVO (BYTES)
BMP	8	indexada	[256x3 double]	Ausente	4618
BMP	24	truecolor	[]	Ausente	10674
GIF	8	indexada	[256x3 double]	LZW	1685
TIFF	8	grayscale	[]	LZW	568
TIFF	24	truecolor	[]	LZW	916
PCX	8	indexada	[]	RLE	1536
PCX	24	truecolor	[]	RLE	1935

Pela Tabela 3.2, é possível ver que para uma mesma imagem de mesmo tamanho, mas com formatos, e, número de bits por *pixel* diferentes, tem-se um gasto de memória variável. Comparando o formato GIF (8 bits por *pixel*) e o BMP(8 bits por *pixel*), observa-se que o gasto do primeiro (1685 bytes) é quase três vezes menor que deste último (4618 bytes); e observando a diferença do BMP de 8 e de 24 bits por *pixel*, o de 24 bits por *pixel* gasta mais do dobro da memória (10674 bytes) do que utiliza 8 bits por *pixel*.

Na primeira comparação, feita no parágrafo anterior, está se comparando um formato que utilizou um algoritmo de compressão e um outro que não usou deste artifício. Uma imagem BMP que tem três vezes o número de bits/*pixel* que a outra, também BMP, gastará, aproximadamente, três vezes mais memória. Ou seja, uma imagem 59x60 (3540 bits), 8 bits/*pixel* implica em 3540 bytes; e 24 bits/*pixel*, 10620 bytes, tirando-se o gasto com os próprios dados de digitalização do formato.

3.5. Formato Utilizado – Justificativa

Neste trabalho, foi utilizado o formato BMP (ou DIB) de 8 *bits* por *pixel*, com a utilização do *software* aplicativo do ambiente *Windows* (Microsoft[®]), *Paint Brush*, como uma forma de padronizar as imagens, detalhe que se mostrou importante para a execução desse trabalho. Sendo que esta escolha foi feita levando-se em conta que as instituições de auxílio ao deficiente visual possuem, em sua maioria, computadores que trabalham com *Windows* (Microsoft[®]) e possuem o *Paint Brush*, tirando-se com isso, a dependência do *software* agregado ao *scanner*.

Como foi visto no item anterior, este tipo de formato ocupa mais memória que o TIFF, por exemplo, mas isto não trará maiores problemas ao usuário já que após completar o processamento do BR Braille e com isso a tradução, a imagem em formato BMP não necessita mais ser armazenada, podendo ser apagada, e a memória liberada.

Capítulo 4 – Tópicos em Processamento Digital de Imagens

Neste capítulo, são abordados os aspectos teóricos relevantes para a realização dessa dissertação. Não se pretende realizar um estudo aprofundado sobre cada item, e sim dar uma visão geral, começando por entender como se constitui a imagem digital.

A imagem é um sinal bidimensional. Deve-se, portanto, estabelecer um universo matemático no qual seja possível definir diversos modelos abstratos de uma imagem. Em seguida deve-se criar um universo de representação discreta desses modelos, com o objetivo de obter uma codificação da imagem de computador (Gomes e Velho, 1994).

Os passos fundamentais para se representar o processamento digital de imagens são: a aquisição da imagem, o pré-processamento, a segmentação, a extração de características, e o reconhecimento e interpretação.

Abordar-se-á algumas operações utilizadas neste trabalho, tais como limiarização, transformação h-mínima, rotulação e também serão destacadas algumas ferramentas necessárias ao entendimento destas operações.

Uma área do processamento de imagens abordada é a morfologia matemática. Segundo Haralick e Nadadur (2000), a morfologia matemática é baseada na teoria de conjuntos. É um sistema algébrico que provém de dois operadores básicos - dilatação e erosão.

4.1 Representação de Imagens

Para ser adequada ao processamento computacional, uma função $f(x,y)$ precisa ser digitalizada tanto espacialmente quanto em amplitude. A digitalização das coordenadas espaciais (x,y) é denominada amostragem da imagem e a digitalização da amplitude é chamada quantização em níveis de cinza (Gonzalez e Woods, 2000).

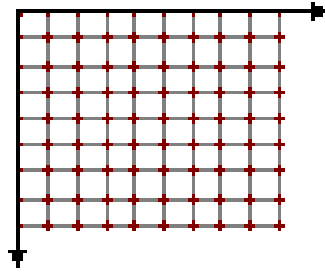


Figura 4.1. Reticulado uniforme da representação matricial da imagem.

A imagem digitalizada pode ser representada por uma matriz $M \times N$, sendo que x está no intervalo de 0 a $M-1$ e y , no intervalo de 0 a $N-1$:

$$f(x, y) = f(x \cdot \Delta x, y \cdot \Delta y). \quad (1)$$

$$f(x, y) = \begin{bmatrix} f(0, 0) & \Lambda & f(0, 1) & \Lambda & f(0, N-1) \\ f(1, 0) & \Lambda & f(1, 1) & \Lambda & f(1, N-1) \\ \vdots & & \vdots & & \vdots \\ f(M-1, 0) & \Lambda & f(M-1, 1) & \Lambda & f(M-1, N-1) \end{bmatrix} \quad (2)$$

Os índices de linhas identificam um ponto na imagem, e o correspondente valor do elemento da matriz identifica o nível de cinza naquele ponto; portanto, $f(x, y)$ representa o valor do nível de cinza da imagem.

4.2 Passos Fundamentais em Processamento de Imagem

No diagrama em blocos da Figura 4.2 estão representados os principais passos do processamento de imagem. O primeiro passo neste processamento é a aquisição da imagem, isto é, a digitalização da imagem através de um equipamento digitalizador.

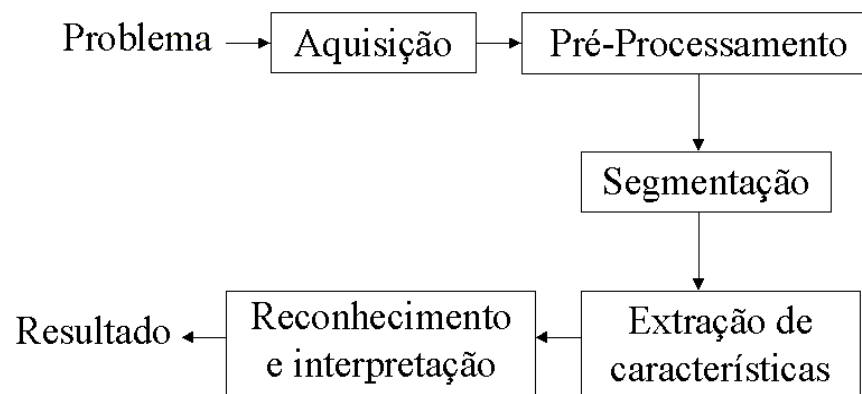


Figura 4.2. Diagrama em blocos mostrando os passos do processamento de imagens digitais.

Na próxima etapa, faz-se o pré-processamento da imagem que consiste em trabalhar a imagem de forma a colocá-la numa representação adequada ao interesse do estudo. Por exemplo, rotacionar a imagem, para que se possa extrair características referentes à orientação do *pixel* na etapa seguinte, ou fazer a filtragem da imagem, para retirar *pixels* indesejáveis. Segundo Gonzalez e Woods (2000), o pré-processamento tem a função de melhorar a imagem de forma a aumentar as chances de sucesso das etapas seguintes.

Com a imagem já preparada é possível realizar a segmentação, que divide uma imagem de entrada em partes ou objetos constituintes. Já a extração de características vai utilizar a imagem segmentada para classificar as características significantes ao estudo. Estas características são dependentes dos objetivos que se tem na imagem, ou seja, se com uma mesma imagem se busca informações diferentes, os algoritmos utilizados nesta etapa, também, serão diferentes.

E por último, temos o reconhecimento e interpretação. Nesta etapa, a imagem estudada será confrontada com a informação pretendida, buscando obter na interpretação a atribuição de significado ao conjunto de objetos.

4.3 Conceitos Básicos

4.3.1 Histograma

O histograma de uma imagem é simplesmente um conjunto de números indicando o percentual de *pixels* naquela imagem, que apresentam um determinado nível de cinza. Estes valores são geralmente representados por um gráfico de barras que fornece para cada nível de cinza o número (ou percentual) de *pixels* correspondentes na imagem (Marques Filho, e Neto Vieira, 1999).

Uma imagem digital com níveis de cinza no intervalo $[0, L-1]$ tem um histograma de função discreta $p(r_k) = n_k/n$, em que r é o k -ésimo nível de cinza, n_k é o número de *pixels* na imagem com esse nível de cinza, n é o número total de *pixels* na imagem e $k = 0, 1, 2, \dots, L-1$. Simplificadamente, pode-se dizer que $p(r_k)$ dá uma estimativa da probabilidade de ocorrência do nível de cinza r_k (Gonzalez e Woods, 2000).

A Figura 4.3 mostra uma imagem digitalizada e a Figura 4.4 mostra o histograma correspondente, obtido com a *toolbox* de processamento de imagem do MATLAB®.

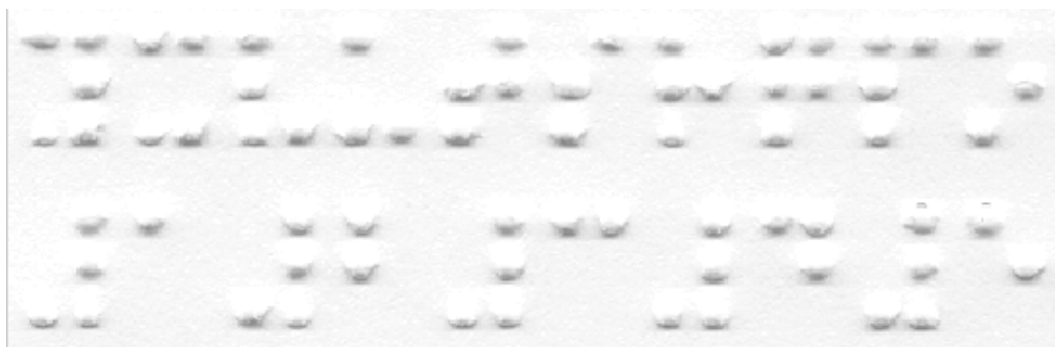


Figura 4.3. Imagem digital de um texto Braille.

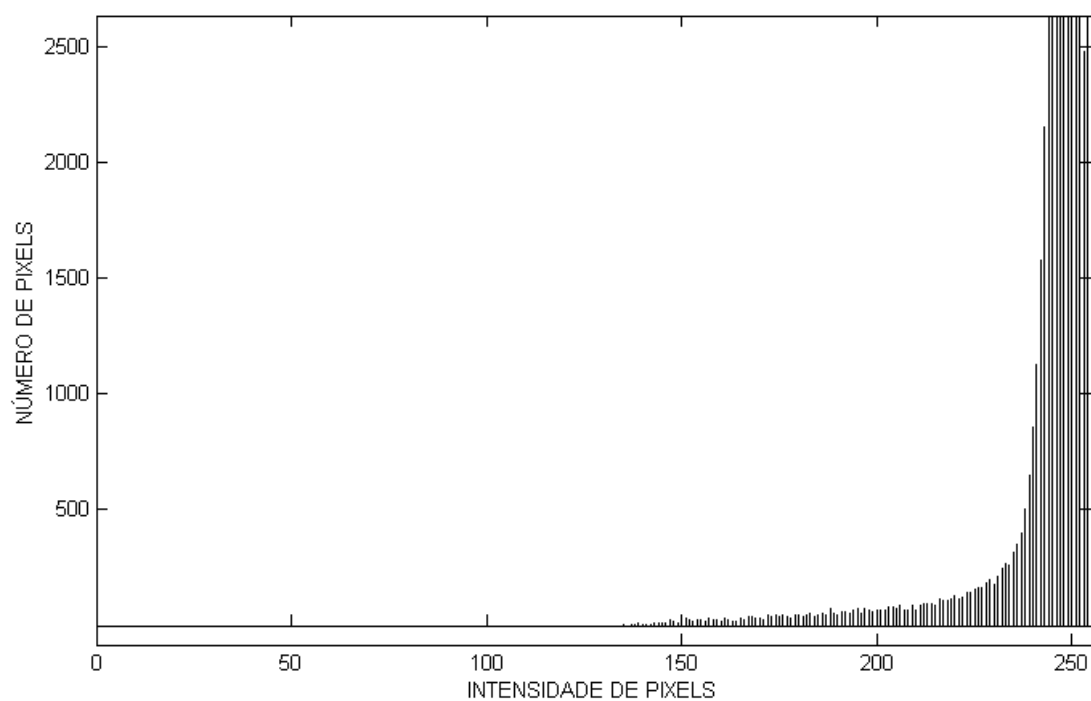
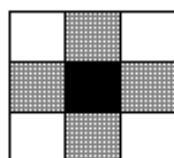


Figura 4.4. Histograma correspondente à imagem digitalizada da Figura 4.3.

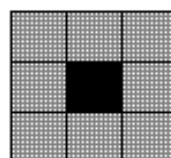
4.3.2 Noção de Conexidade

A adjacência entre pontos pode ser definida a partir da noção de vizinhança. A vizinhança tem duas formas mais comuns (Figura 4.5) de ser definida (Jesus, 1997):

- vizinhança 4: os 4 pontos imediatamente ao norte, sul, leste e oeste do *pixel* em estudo formam a vizinhança;
- vizinhança 8: os 8 pontos imediatamente ao nordeste, leste, sudeste, sul, sudoeste, oeste, noroeste e norte do *pixel* em estudo formam a vizinhança.



(a)



(b)

Figura 4.5. Exemplos de (a) vizinhança-4 e (b) vizinhança-8.

Uma primeira definição de um conjunto conexo é que dois pontos quaisquer desse conjunto podem ser ligados por uma curva completamente contida nesse conjunto. Uma outra definição mais complexa é que dois pixels são ditos conexos se existe no mínimo uma seqüência de pixels que os liga, tal que dois pixels consecutivos dessa seqüência satisfaçam a condição de conectividade imediata (Facon, 1993). A Figura 4.6 exemplifica a diferença em se considerar uma componente 4-conexa e 8-conexa. Se a figura for observada com conexidade-8 tem-se um único componente; considerando a conexidade-4, ela possui três componentes.

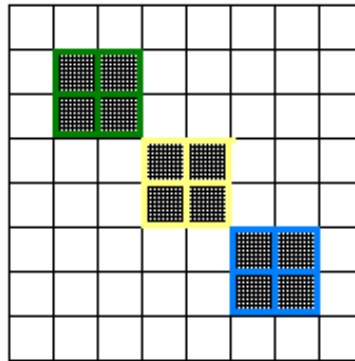


Figura 4.6. Representação de três componentes 4-conexas ou uma componente 8-conexa.

4.4 Limiarização

A maioria dos métodos de limiarização encontrados na literatura é de natureza global e isto pode ser explicado por sua menor complexidade e conseqüentemente menor tempo de execução.(Queiros, 1997). Existem outros métodos, como o dependente do ponto, o dependente por região e o local; pelas vantagens citadas foi escolhido um método de limiarização global para ser utilizado nesse trabalho.

No método de limiarização global, se for determinado um nível de limiar, o método é denominado binível (binarização da imagem), e, se mais de um limiar for determinado ele é chamado multinível (imagem composta de mais de dois níveis de cinza). Nessa dissertação, será utilizada a binarização da imagem, ou seja, a imagem resultante dessa operação será uma imagem em preto e branco.

É importante selecionar um limiar adequado de nível de cinza para extração dos objetos de seus fundos de imagem (Otsu, 1979). Várias técnicas foram propostas para automatizar a determinação do limiar.

Soares (2001) avaliou três métodos de binarização: limiarização de Otsu, classificação por *Watershed* e detecção de passagem por zero, sendo que a limiarização de Otsu se mostrou mais robusta nos aspectos de tempo de processamento e qualidade de imagem de saída.

4.4.1 Limiarização de Otsu

Sejam os *pixels* dados numa imagem representados por L níveis de cinza $[1, 2, \dots, L]$. O número de *pixels* no nível i é denotado por n_i e o número total de *pixels* por $N = n_1 + n_2 + \dots + n_L$. Normaliza-se o histograma, acumulando-o como uma distribuição de probabilidade:

$$p_i = n_i / N, \quad p_i \geq 0, \quad \sum_{i=1}^L p_i = 1. \quad (3)$$

Onde:

p_i = probabilidade de ocorrência da intensidade i .

Em seguida, dividimos os *pixels* em duas classes: C_0 e C_1 (plano de fundo e objetos), por meio de um valor de limiar k ; C_0 denota *pixels* com níveis $[1, \dots, k]$; e C_1 denota *pixels* com níveis $[k + 1, \dots, L]$. Então, as probabilidades de ocorrência das classes C_0 e C_1 , $\Pr(C_0)$ e $\Pr(C_1)$, e suas médias μ_0 e μ_1 são dadas por:

$$\omega_0 = \Pr(C_0) = \sum_{i=1}^k p_i = \omega(k) \quad (4)$$

$$\omega_1 = \Pr(C_1) = \sum_{i=k+1}^L p_i = 1 - \omega(k) \quad (5)$$

e

$$\mu_0 = \sum_{i=1}^k i \Pr(i|C_0) = \sum_{i=1}^k i p_i / \omega_0 = \mu(k) / \omega(k) \quad (6)$$

$$\mu_1 = \sum_{i=k+1}^L i \Pr(i|C_1) = \sum_{i=k+1}^L i p_i / \omega_1 = \frac{\mu_T - \mu(k)}{1 - \omega(k)} . \quad (7)$$

Onde os momentos cumulativos de ordem “zero” e “um” do histograma (para níveis de cinza até k) são dados respectivamente por:

$$\omega(k) = \sum_{i=1}^k p_i \quad e \quad \mu(k) = \sum_{i=1}^k i p_i \quad (8)$$

E a média total dos níveis na imagem é dada por:

$$\mu_T = \mu(L) = \sum_{i=1}^L i p_i \quad (9)$$

As relações a seguir se verificam para qualquer escolha de k :

$$\omega_0 \mu_0 + \omega_1 \mu_1 = \mu_T , \quad \omega_0 + \omega_1 = 1 \quad (10)$$

As variâncias das classes são dadas por:

$$\sigma_0^2 = \sum_{i=1}^k (i - \mu_0)^2 \Pr(i|C_0) = \sum_{i=1}^k (i - \mu_0)^2 p_i / \omega_0 \quad (11)$$

$$\sigma_1^2 = \sum_{i=k+1}^L (i - \mu_1)^2 \Pr(i|C_1) = \sum_{i=k+1}^L (i - \mu_1)^2 p_i / \omega_1 \quad (12)$$

Para conseguir uma melhor limiarização (de nível k), são estabelecidos critérios de medida (ou medida de separabilidade):

$$\lambda = \sigma_B^2 / \sigma_w^2, \quad k = \sigma_T^2 / \sigma_w^2, \quad \eta = \sigma_B^2 / \sigma_T^2 \quad (13)$$

Onde:

$$\begin{aligned} \sigma_w^2 &= \omega_0 \sigma_0^2 + \omega_1 \sigma_1^2 \\ \sigma_B^2 &= \omega_0 (\mu_0 - \mu_T)^2 + \omega_1 (\mu_1 - \mu_T)^2 \\ &= \omega_0 \omega_1 (\mu_1 - \mu_0)^2 \end{aligned} \quad (14)$$

E devido à Equação (10),

$$\sigma_T^2 = \sum_{i=1}^L (i - \mu_T)^2 p_i \quad (15)$$

σ_W^2 , σ_B^2 e σ_T^2 são: variância das classes, a variância entre classes, e a variância total dos níveis, respectivamente. Então, o problema está em otimizar a escolha de um valor limiar k que maximize uma das funções objeto dadas na Equação (13).

Segundo Otsu (1979), o critério de medida mais simples de k é o η , e portanto, foi o critério adotado.

Para conseguir um k^* ótimo, é necessário maximizar η , e para isto é preciso achar o valor máximo de σ_B^2 .

Portanto,

$$\eta(k) = \sigma_B^2(k) / \sigma_T^2 \quad (16)$$

$$\sigma_B^2(k) = \frac{[\mu_T \omega(k)]^2}{\omega(k)[1-\omega(k)]} \quad (17)$$

e o valor ótimo do k^* é:

$$\sigma_B^2(k^*) = \max_{1 \leq k < L} \sigma_B^2(k). \quad (18)$$

Como consequência, ao encontrar o valor de $\sigma_B^2(k^*)$ (Equação (18)), encontra-se o k^* , ou seja, o valor ótimo de k , dividindo a imagem em duas classes, C_0 e C_1 , como está representado na Figura 4.7.

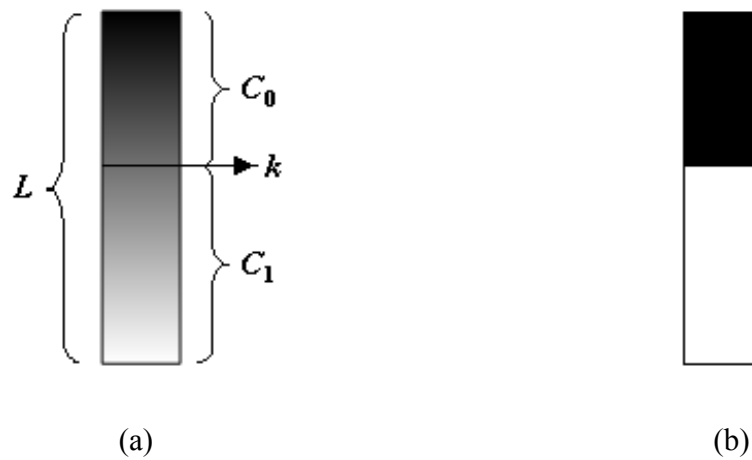


Figura 4.7. Representação da divisão das classes C_0 e C_1 pela limiarização de Otsu, em uma imagem de L níveis de cinza. (a) Imagem de Entrada; (b) Imagem limiarizada.

4.5 Morfologia Matemática

Morfologia matemática pode ser definida como uma teoria para análise de estruturas espaciais.

- Morfologia - porque objetiva analisar o contorno (molde) e a forma de objetos.
- Matemática - a análise do contorno e/ou forma é baseada em um conjunto de teorias, geometria integral e álgebra combinacional.

Os problemas de análise de imagem, na verdade, são muito complexos e podem ser resolvidos somente se foi utilizada uma combinação de várias transformações elementares (Soille, 1999).

O princípio básico da morfologia matemática consiste em extrair uma informação relativa à geometria e a topologia de um conjunto desconhecido de uma imagem pela transformação a partir de um outro conjunto completamente definido chamado elemento estruturante (Facon, 1993).

Morfolologicamente, mensurar as características de um objeto nada mais é do que procurar ver neste objeto aquelas características conhecidas que se deseja encontrar. As operações que são realizadas sobre este objeto de estudo são denominadas "Operações Morfológicas". As características conhecidas que são utilizadas num processo de comparação são os chamados "Elementos Estruturantes" (Araújo Júnior, 1996).

Os elementos estruturantes (S.E.), como foi dito anteriormente, são objetos conhecidos. Eles são escolhidos conforme o que se deseja obter na imagem, ou seja, ele é dependente da forma do objeto estudado. Para facilidade da maioria dos trabalhos requeridos sobre imagens, usa-se S.E planos, ou seja formados por um conjunto de *pixels* binário.

As formas dos S.E. mais utilizada são: o disco Euclidiano, o segmento de linha horizontal, a cruz e a caixa (*box*), que são mostrados na Figura 4.8.



Figura 4.8. Representação dos elementos estruturantes mais utilizados.

Em uma imagem Braille digitalizada, como a apresentada na Figura 4.3, pode-se ver que os pontos constituintes da cela Braille se aproximam da forma de um disco, e, portanto, para caracterizá-los de forma adequada o elemento estruturante poderia ter esta forma. Embora esta escolha fosse natural, se fosse escolhida a cruz ou a caixa também não se estaria afastando muito da representação do ponto da cela Braille. Isto porque numa imagem digital um disco na verdade seria dado por uma aproximação e para se conseguir um melhor resultado teria que se usar um número grande de *pixels* vizinhos, o que também pode ser inconveniente.

Na morfologia matemática dá-se um tratamento diferente a imagens binárias e imagens em nível de cinza, ou seja, tem-se a morfologia binária e a morfologia cinza. Em Facon (1993) encontra-se a seguinte descrição delas:

- Morfologia Binária - na vizinhança de cada *pixel* da imagem original, é procurada uma configuração de pontos pretos e brancos. Quando a configuração é encontrada, ao *pixel*

correspondente ao da imagem resultante é dado o rótulo "verdadeiro"; senão, o *pixel* resultante recebe o rótulo de "falso".

- Morfologia Cinza - na vizinhança de cada pixel ou numa parte da vizinhança da imagem original, é necessário conhecer o valor do pixel mais escuro "min", o valor do pixel mais claro "max". O valor de pixel resultante corresponde a uma combinação particular de "max" e "min". O tamanho e a forma da vizinhança, as regiões de pesquisa de "min" e "max" e o algoritmo determinam completamente uma operação de morfologia cinza.

As operações fundamentais em morfologia matemática são a erosão e a dilatação (operação dual da erosão). Sendo que, como visto anteriormente, a erosão e dilatação binária diferem da erosão e dilatação em imagens em níveis de cinza.

4.5.1 Erosão e Dilatação Binárias

Antes de equacionar a erosão, é preciso citar a translação. A translação de um conjunto A por um ponto x é denotado por $A + x$ e é definido por:

$$A + x = \{a + x : a \in A\} \quad (19)$$

Onde: a = coordenada de um *pixel* pertencente a A .

A erosão de A por B , denotada por $\varepsilon_B(A)$, pode ser interpretada como o lugar geométrico de todos os centros, tal que a translação de B por x esteja totalmente contida em A (Shih e Mitchell, 1992). A equação que expressa a erosão de uma imagem A por um elemento estruturante B é dada por:

$$\varepsilon_B(A) = \{x \in B / (B + x) \subseteq A\} \quad (20)$$

Na Figura 4.9, será representada a letra “r” digitalizada Braille (binária), com sua imagem negativa e suas respectivas erosões. Nota-se que o efeito da erosão num ponto Braille preto é de ampliação do mesmo, já num ponto Braille branco é de redução do mesmo.

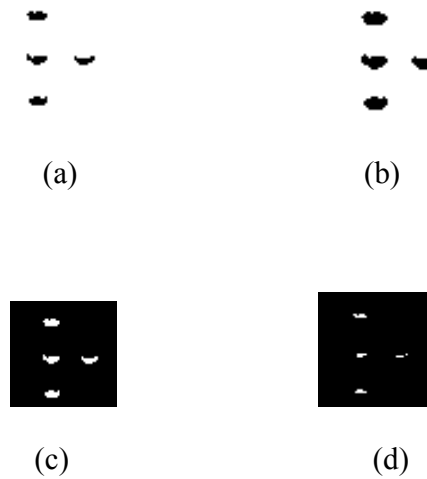


Figura 4.9. (a) Imagem digitalizada binária da letra “r”; (b) imagem da letra “r” em (a) erodida; (c) negativo da imagem digitalizada binária da letra “r”; e (d) imagem da letra “r” em (c) erodida.

A Equação (20) diz que ao transladar o elemento estruturante sobre a imagem, os pontos x (origem do E.S.) são marcados quando o E.S. em translação estiver totalmente contido na imagem A .

Como foi dito anteriormente, a dilatação é uma operação dual da erosão, mas antes de representar a dilatação é preciso caracterizar a reflexão. A reflexão do elemento estruturante B é dada por:

$$\hat{B} = \{-b : b \in B\} \quad (21)$$

A formulação da dilatação que envolve a translação do elemento estruturante rotacionado que intersecta a imagem de entrada é dada por:

$$\delta_B(A) = \{x: (\hat{B} + x) \cap A \neq \emptyset\} \quad (22)$$

Agora na Figura 4.10, será mostrada a letra “r” digitalizada Braille (binária), com sua imagem negativa e suas respectivas dilatações. Nota-se a dualidade entre a erosão e a dilatação, comparando as Figura 4.9 e 4.10.

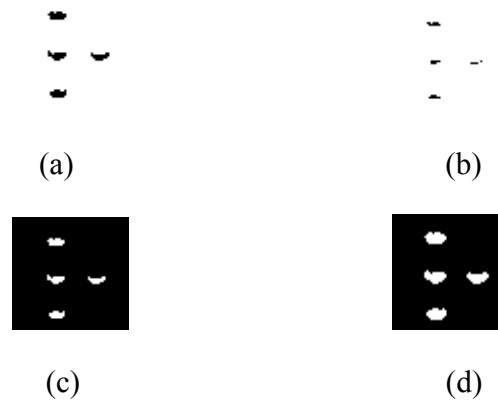


Figura 4.10. (a) Imagem digitalizada binária da letra “r”; (b) imagem da letra “r” em (a) dilatada; (c) negativo da imagem digitalizada binária da letra “r”; e (d) imagem da letra “r” em (c) dilatada.

4.5.2 Erosão e Dilatação em Níveis de Cinza

Shih e Mitchell (1989) citam várias proposições de erosão e dilatação nível de cinza, dentre elas a proposição de um método computacional, que eles chamam alternativo, destes operadores.

Por ser facilmente implementável este método computacional foi escolhido para ser utilizado nesta dissertação. Sua formulação é:

$$f:F \rightarrow E \text{ e } k:K \rightarrow E.$$

Então $f \oplus_g k:F \oplus_g K \rightarrow E$ pode ser calculado por:

$$(f \theta_g k)(x) = \min \{f(x+z) - k(z)\} \quad (23)$$

para todo $z \in K$ e $x+z \in F$; e

$$(f \oplus_g k)(x) = \max \{f(x-z) + k(z)\} \quad (24)$$

para todo $z \in K$ e $x-z \in F$.

É mais simples e mais usual que se trabalhe com E.S. planar, e neste caso $k(z)$ será igual a zero. Então, nas equações anteriores é eliminado o $k(z)$.

$$(f \theta_g k)(x) = \min \{f(x+z)\} \quad (25)$$

para todo $z \in K$ e $x+z \in F$; e

$$(f \oplus_g k)(x) = \max \{f(x-z)\} \quad (26)$$

para todo $z \in K$ e $x-z \in F$.

4.5.3 Transformação H-Mínima ou H-Basin

A transformação h-mínima é dada por:

$$HMIN_{h,D}(f) = f \perp_D (f + h) \quad (27)$$

Ou seja, fazer a transformação h-mínima é igual a fazer a reconstrução superior da imagem de entrada f , na imagem marcadora $(f + h)$, sendo que a reconstrução superior é obtida, fazendo a erosão condicional da imagem marcadora $(f + h)$ pelo elemento estruturante D , onde a condição de parada é que a imagem erodida esteja contida dentro da imagem de entrada.

$$f \perp_D (f + h) = ((f + h) \ominus_f D)^\infty \quad (28)$$

4.5.4 Rotulação

A rotulação de componentes conexas de uma imagem binária é uma transformação diretamente associada com a noção de conexidade. Consiste em juntar cada *pixel* pertencente a uma componente conexa da imagem binária de entrada para um valor específico de nível de cinza, sendo considerados diferentes valores para cada componente conexa (costuma-se desconsiderar o fundo de imagem, mantendo-o no valor original). A imagem resultante da rotulação é conhecida como imagem rotulada (Soille, 1999).

4.6 Rotação

A rotação das imagens é feita utilizando-se das seguintes equações (Sharnet, 1999):

$$x' = x \cdot \cos(\alpha) + y \cdot \sin(\alpha) \quad (29)$$

$$y' = y \cdot \cos(\alpha) - x \cdot \sin(\alpha) \quad (30)$$

Onde x e y são as coordenadas do *pixel* antes da rotação e x' e y' as coordenadas após a rotação.

O cálculo do ângulo α é feito através do método dos mínimos quadrados:

$$\alpha = \frac{(c - d \cdot e)}{\left(f - \frac{e^2}{n}\right)} \quad (31)$$

Onde:

$$c = \sum x \cdot y$$

$$d = \sum \frac{y}{n}$$

$$e = \sum x$$

$$f = \sum x^2$$

e x e y são as coordenadas dos *pixels* correspondentes aos pontos 1 e 4 de cada cela da linha que se quer calcular o ângulo de rotação e n é o número total de pontos considerados.

Neste capítulo, foi vista uma descrição teórica dos tópicos. No Capítulo 5 esta teoria será aplicada para a realização do programa tradutor.

Capítulo 5 –Processamento das Imagens Braille

O programa tradutor BR Braille tem por finalidade auxiliar pessoas com deficiência visual alfabetizadas em Braille a se comunicarem, através da escrita, com pessoas que não conhecem o sistema Braille.

Todo o processamento realizado é dependente da imagem digitalizada através do *scanner*, ou seja, escolheu-se os algoritmos a serem utilizados estudando as características da maioria das imagens obtidas no processo de digitalização.

Como foi visto no capítulo anterior, foram realizados os seguintes passos após a aquisição da imagem: pré-processamento, segmentação, extração de características, reconhecimento e interpretação. Este capítulo mostra como foi realizada cada etapa do processamento digital de imagens Braille, incluindo, também, figuras de resultados obtidos em cada uma delas.

5.1. Requisitos do Sistema

Segundo Carvalho (2001) que sugeriu a realização deste trabalho em sua tese de doutorado, o programa tem que atender aos seguintes requisitos:

- O processo deverá ser de fácil operação.
- O processo deverá ser de baixo custo.
- O sistema deverá, na sua composição, fazer uso, o máximo possível, dos componentes de hardware e software já existentes no mercado, mais comuns no dia-a-dia da maioria dos usuários de informática.
- No caso em que haja necessidade de desenvolvimento de novos componentes, estes deverão ser compatíveis com as plataformas mais comuns do mercado.

Para atender às sugestões acima, foram realizadas as etapas especificadas nos próximos itens, sendo que não foi necessário acrescentar nenhum componente extra, além do próprio programa BR Braille.

5.2. Aquisição das Imagens



Figura 5.1. Desenho esquemático dos equipamentos necessários para utilização do programa tradutor.

Os equipamentos necessários (Figura 1) são de baixo custo e normalmente já estão disponíveis em muitos dos locais a que esta interface se destina. Para digitalizar a imagem utiliza-se um *scanner* de mesa, e um microcomputador que tenha um processador Pentium, 100 MHz, com 32 Mb de RAM, ou superior com Windows 98[®]. A saída do texto traduzido pode ser armazenada na memória do computador e/ou enviada para a impressora.

Como foi visto no Capítulo 2, o Braille é um sistema de leitura e escrita tátil formado, no Brasil, por seis pontos em alto-relevo, dispostos em duas colunas de três pontos. Quando digitalizados em nível de cinza geram, em sua maioria, imagens com nível de fundo próximo ou igual a 255 (papel branco) e valores de *pixel* menores que 240 para os pontos (mais escuros).

O reconhecimento dos caracteres encontra as seguintes dificuldades:

- Rotação da imagem na hora de digitalizar;
- Alteração do relevo dos pontos Braille escritos no papel;
- Alguma rotação na linha dos caracteres Braille que pode ser causada ou pela utilização de uma máquina ou impressora Braille com mecanismo de inserção do papel desregulado ou a má colocação do papel nos equipamentos, ou ainda quando no uso da reglete, na retirada do mesmo para colocá-lo na posição abaixo para continuar escrevendo o texto.

Deve-se, portanto, verificar se não ocorreu nenhum desses problemas para que se obtenha um melhor desempenho do programa tradutor.

É necessário que se tenha uma folha digitalizada com um máximo de rotação igual a 2°, senão a percentagem de erro cresce drasticamente. Além disso, os valores dos níveis de cinza dos pontos vão depender muito do quanto o alto relevo do texto original está preservado, e isso vai influenciar no resultado final da tradução, assim, quanto mais preservados os pontos, menos erros obtém-se na tradução.

5.3. Introdução ao Pré-Processamento

A imagem Braille digitalizada apresenta ruídos além das informações desejadas. Estes ruídos são retirados através da etapa conhecida como pré-processamento. Portanto, é neste momento que se separa o que é desejado do que não é desejado na imagem. As operações escolhidas para se fazer o pré-processamento tiveram como prioridade a não exclusão da informação desejada, ou seja, os pontos pertencentes à cela Braille. Esse aspecto é necessário, pois ruído e pontos Braille possuem características topológicas muito semelhantes. Se um ruído estiver presente após a realização do pré-processamento é possível fazer sua retirada na próxima etapa, mas se um ponto da cela Braille for retirado, esta informação perdida não tem como ser recuperada.

5.4. A Escolha das Etapas do Pré-Processamento

O ruído aparece nas imagens devido ao processo de digitalização, como foi visto no Capítulo 2. Além disso, o vidro do *scanner* e a própria folha podem conter sujeiras, que ao serem digitalizadas correspondem a ruídos semelhantes aos anteriores e também totalmente aleatórios.

Como os pontos Braille e os ruídos encontrados nas imagens digitalizadas possuem características topológicas semelhantes, utilizou-se um operador da morfologia matemática (explicado no capítulo anterior) como um primeiro passo para se realizar a remoção de pontos indesejados.

A morfologia matemática é baseada na análise das formas da imagem, portanto, adequada a esse estudo, pois através dos detalhes que separam os ruídos dos pontos desejados é que teremos um tratamento adequado na imagem.

É necessário conhecer os valores de *pixel* existentes no fundo da imagem, nos pontos da cela Braille e nos ruídos. O estudo dos valores do *pixel* foi realizado usando uma *toolbox* de processamento de imagem do MATLAB[®].

Fazendo uma análise desses valores, na maioria das imagens digitalizadas, notou-se que os pontos da cela Braille se encontram abaixo de 85% do valor do fundo; isto não quer dizer que todos os *pixels* correspondentes a um ponto terão valores abaixo de 85%, mas sim que a maioria deles está nesta faixa de valores, o que implica na preservação do ponto como um todo. Já os *pixels* pertencentes ao ruído terão, em sua maioria, um valor maior que 85% do valor de fundo. O estudo destes detalhes característicos das imagens é importante para a escolha do operador morfológico inicial a se utilizar para o pré-processamento.

Nesse momento, o que diferencia um ponto Braille de um ruído é a diferença no número de *pixels* que tem valores mais próximos dos valores de fundo da imagem. Portanto, deve-se utilizar um operador morfológico que elimine estes *pixels* fazendo-os iguais ao fundo, restando com isto poucos *pixels* que sejam pertencentes ao ruído.

Ao se conseguir esse resultado, a diferença entre ruído e ponto Braille não será mais os valores dos *pixels* e sim a quantidade deles, permitindo que se trabalhe com a imagem

binária, não necessitando que ela esteja em nível de cinza. A partir daí rotula-se os pontos de tal forma que os *pixels* que formam cada ponto sejam reconhecidos como pertencentes a um determinado ponto, ou seja, cada ponto passa a ter uma “identidade” própria.

A rotação é uma outra etapa importante do pré-processamento, pois como visto acima é possível que haja folhas cujas linhas apresentem uma inclinação. Contudo o algoritmo utilizado para calcular o ângulo de rotação de cada linha atende a pequenas rotações, como especificado acima.

5.5. A Seqüência do Pré-Processamento

É possível fazer uma escolha adequada, com essa análise de como se comportam as imagens Braille digitalizadas, ou seja, que o pré-processamento elimine uma grande parte dos ruídos da maioria das imagens, conseguindo obter um resultado satisfatório. Essa é uma etapa de fundamental importância para se obter sucesso na tradução do texto Braille para o texto em caracteres alfanuméricos.

Inicialmente, utilizou-se a transformação h-mínima como base para o pré-processamento. Ela produz uma elevação do nível de cinza na imagem como um todo (dos pontos desejados, do ruído e do fundo da imagem).

Os *pixels* pertencentes ao ruído são mais próximos aos do fundo. Esses irão alcançar o valor de 255, em sua maioria, assim como uma pequena parte de *pixels* pertencentes aos pontos Braille. Esses dois resultados levam a uma remoção significativa de ruídos e a uma diminuição do tamanho dos pontos Braille; além disso, o fundo da imagem vai ficar totalmente branco, isto é, com todos os *pixels* com valores iguais a 255.

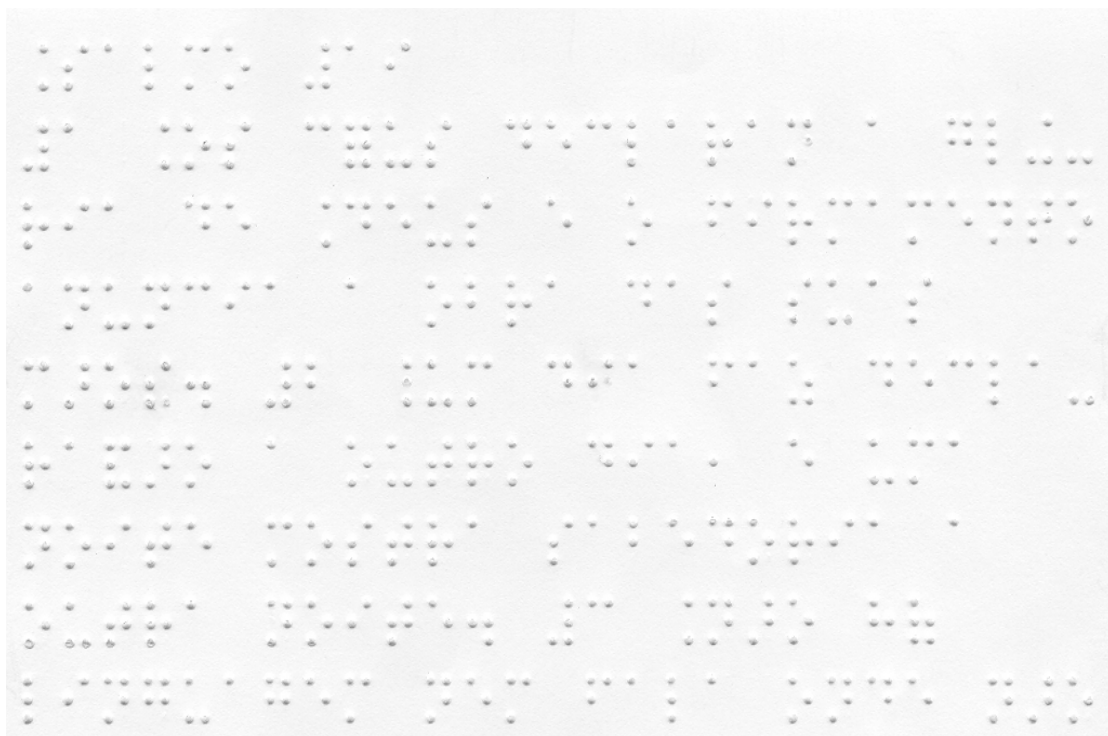


Figura 5.2. Imagem de entrada com os valores dos *pixels* de fundo menores que 250.

A utilização desse operador foi adequada a algumas imagens digitalizadas (como por exemplo, a da Figura 5.2), mas não para todas (como a apresentada na figura 5.3). Imagens com fundo predominantemente branco, ou seja, de valor maior que 250, têm seus pontos Braille, em sua maioria, também com níveis de cinza elevados, como pode ser visto através do histograma na Figura 5.5. Portanto, não é satisfatória a utilização da transformação h-mínima nesse tipo de imagem. Por outro lado, tal transformação é adequada a imagens com fundo predominantemente escuro (com valores de níveis de cinza menores) (Figura 5.2), como indicado no histograma da Figura 5.4.



Figura 5.3. Imagem de entrada com o valor dos *pixels* de fundo igual a 255.

Observando-se o histograma (Figura 5.4) da imagem com fundo claro, verifica-se que a maior quantidade de *pixels* que a imagem Braille possui pertence ao fundo. Dessa forma define-se o valor de *pixel* do fundo da imagem, e, se haverá ou não a necessidade de utilizar a transformada h-mínima.

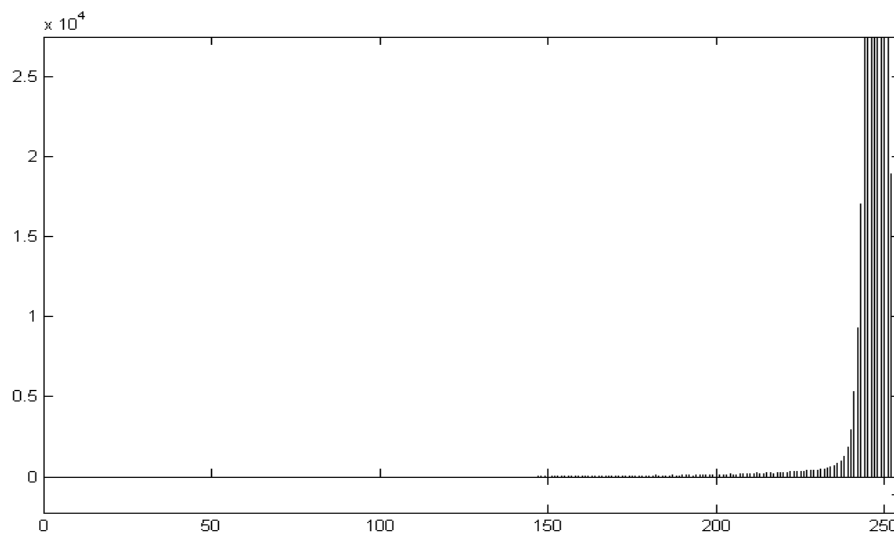


Figura 5.4. Histograma da imagem com fundo predominantemente escuro mostrada na Figura 5.2 (MATLAB®).

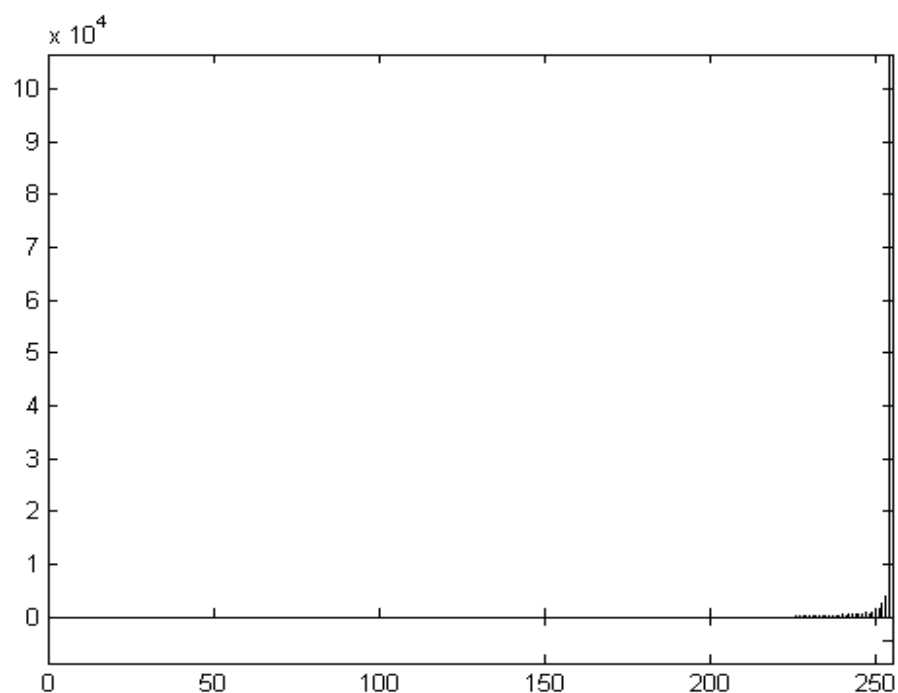


Figura 5.5. Histograma da imagem com fundo predominantemente branco apresentada na Figura 5.3 (MATLAB®).

A partir da análise sobre a realização da transformada h-mínima, os valores de nível de cinza não determinam mais a diferença entre pontos Braille e ruído, podendo ser feita a limiarização, isto é, transformar a imagem em níveis de cinza em imagem binária.

O valor de limiar deve ser feito automaticamente, isto é, sem a intervenção do usuário, pois se pressupõe que ele não tem um conhecimento sobre os valores de *pixels* que compõem imagens digitalizadas. O valor do limiar vai ser dependente da imagem que está sendo processada no momento. Para isto usou-se a limiarização de Otsu, que como visto no capítulo anterior, utiliza-se do histograma da imagem para achar o valor limiar ótimo.

Após encontrar o limiar, “varre-se” a imagem e se faz a alteração: todo *pixel* que tiver um valor acima do limiar, será modificado para o valor 1, ou seja, o *pixel* será branco; e o que estiver abaixo torna-se 0 (Figura 5.6).

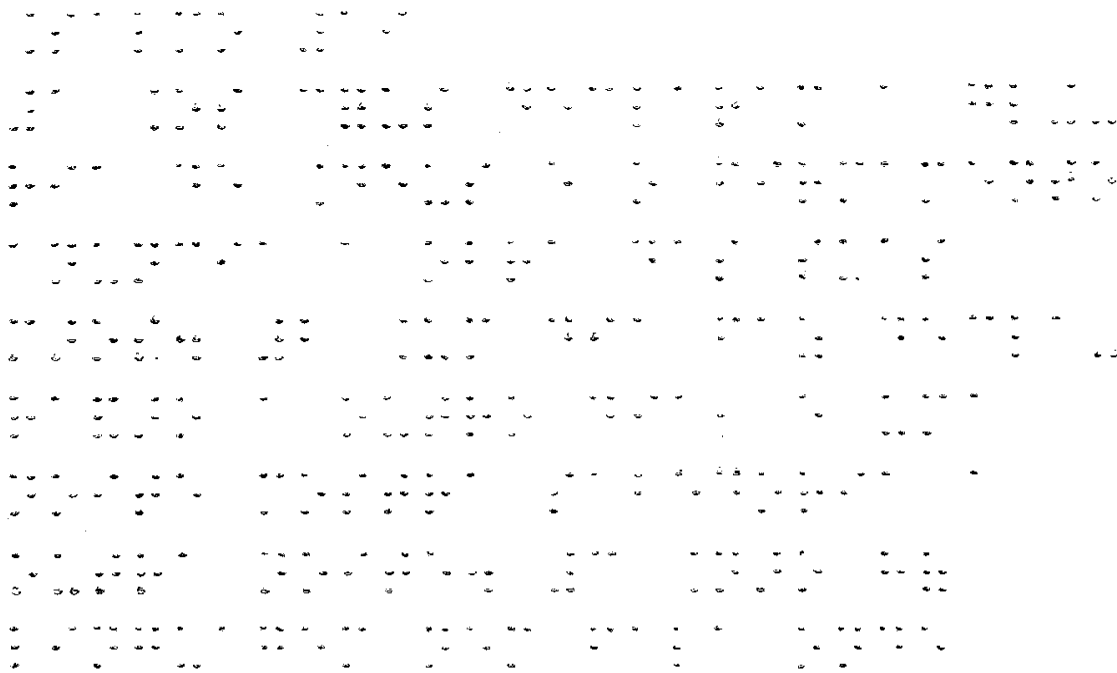


Figura 5.6. Imagem da Figura 5.2 limiarizada.

Nota-se que os parâmetros são relacionados, ou seja, dependendo da imagem resultante obtida após a transformação h-mínima, tem-se um valor de limiar da imagem binarizada.

O próximo passo é a rotulação.

A rotulação identifica as componentes conexas e quantifica o número de *pixels* em cada componente. Portanto, dependendo do número de ruídos (*pixels* indesejados) restantes após a limiarização, excluem-se as componentes conexas que tenham um valor menor que um valor mínimo e maior que o valor máximo. Estes cálculos devem ser feitos através de um percentual da resolução (dpi) com que a imagem foi digitalizada.

Na Figura 5.7, os pontos em vermelho são componentes conexas cujos números de *pixels* eram menores que o valor calculado para essa imagem, ou seja, um percentual da resolução dessa imagem.

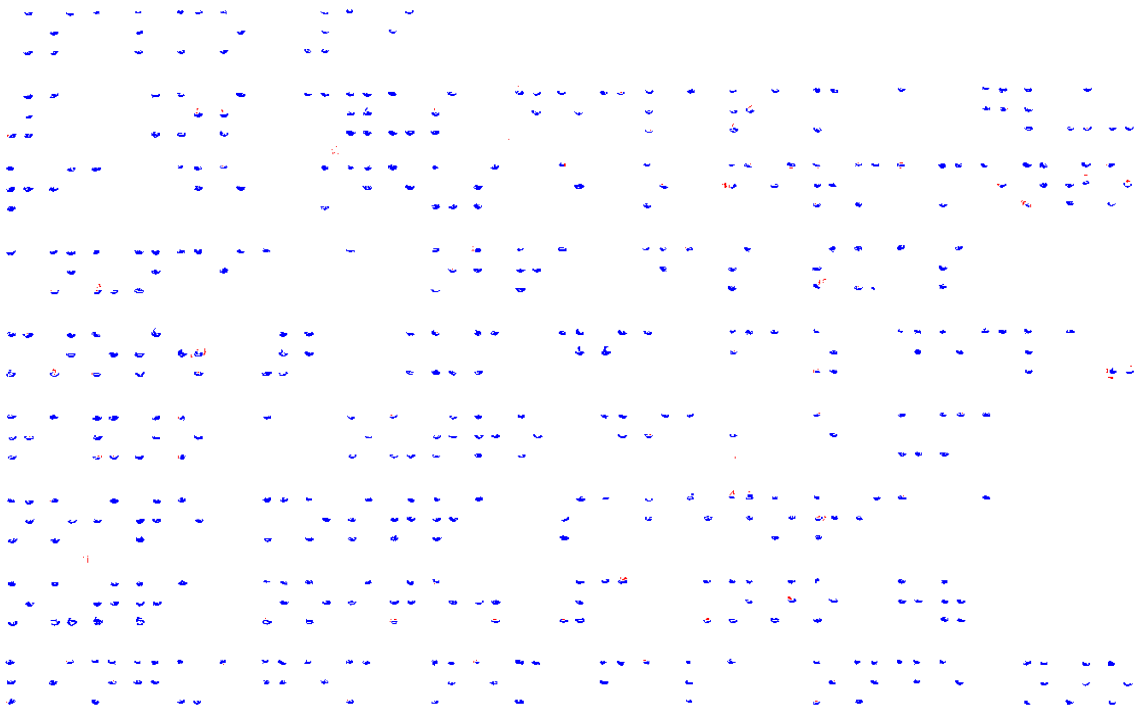


Figura 5.7. Resultado da rotulação feita a partir da imagem da Figura 5.6.

Realizando essas operações para o pré-processamento, utiliza-se um número de redundâncias satisfatórias para exclusão dos *pixels* indesejados. Por exemplo, ao se fazer a limiarização muitos ruídos são excluídos, mas em compensação vários *pixels* são desconectados uns dos outros, gerando novos ruídos. A diferença é que esses novos ruídos têm um número de *pixels* previsível, o que facilita sua exclusão através da rotulação. Portanto, na verdade, essa redundância é algo que obrigatoriamente tem que acontecer.

Em seguida, tem-se a execução da rotação da imagem para corrigir pequenos desvios como já citado acima. Foi considerada a correção de cada linha por vez e para isso é feita a aproximação do ângulo, chamado de α através do método dos mínimos quadrados (apresentado no capítulo 4), utilizando-se os pontos 1 e 4 das celas pertencentes à linha. Para se corrigir um eventual deslocamento horizontal das linhas uma em relação à outra, foi realizado o cálculo de um novo ângulo utilizando-se os pontos 1, 2 e 3 das celas pertencentes à primeira coluna de celas, ou seja, a coluna mais à esquerda na página.

Assim, considerando a posição da linha na página original, é possível realizar a correção desse deslocamento através de um simples cálculo trigonométrico.

Os algoritmos apresentados, neste capítulo, serão descritos na notação de pseudocódigo.

Algoritmos Usados no Pré-Processamento

Algoritmo: Histograma

Entrada: Imagem $Cela_i$

Saída: $Hist_{ij}$

```
Para Todo  $Cela_i$  faça
    Para Todo  $j$  entre 1 e 6 faça
        Para Todo  $p \in \text{quadrante}_j$  faça
            Se  $\exists p = 1$  então
                 $Hist_{ij} = 1$ ;
            Fim Se
        Senão
             $Hist_{ij} = 0$ ;
        Fim Senão
    Fim Para Todo
Fim Para Todo

FIM DO ALGORITMO
```

Algoritmo 5.1. Algoritmo útil a várias etapas do processamento de imagem.

Algoritmo: Limiarização

Para x de 0 a 255 **faça**

$\text{probabilidade}[x] = \text{histograma}[x] / n$

/* n é o número total de pixels da imagem*/

Fim Para

$\mu_t[1] = \text{probabilidade}[1]$;

$\omega_0[1] = \text{probabilidade}[1]$;

Se $\omega_0[1] \neq 0$ **então**

$\mu_0[1] = \mu_t[1] / \omega_0[1]$;

senão

$\mu_0[1] = 0$;

Fim Se

```

Para x de 2 a 255 faça
     $\omega_0[x] = \omega_0[x-1] + \text{probabilidade}[x];$ 
     $\mu_t[x] = \mu_t[x-1] + x * \text{probabilidade}[x];$ 
    Se  $\omega_0[x] \neq 0$  então
         $\mu_0[x] = \mu_t[x] / \omega_0[x];$ 
    senão
         $\mu_0[x] = 0;$ 
    Fim Se
Fim Para

Para x de 1 a 255 faça
    Se  $(1 - \omega_0[x]) \neq 0$  então
         $\mu_1[x] = (\mu_t[255] - \mu_t[x]) / (1 - \omega_0[x]);$ 
    senão
         $\mu_1[x] = 0;$ 
    Fim Se
     $\sigma_b[x] = \omega_0[x] * (1 - \omega_0[x]) * (\mu_1[x] - \mu_0[x])^2;$ 
Fim Para

Para x de 1 a 255 faça
     $\sigma_t = \sum \{ (x - \mu_t[255])^2 * \text{probabilidade}[x] \};$ 
Fim Para

Para x de 1 a 255 faça
    Razão[x] =  $\sigma_b[x] / \sigma_t;$ 
Fim Para
Limiar=1;
Maxi=Razão[1];
Para x de 2 a 255 faça
    Se Razão[x] > Maxi então
        Maxi=Razão[x];
    Limiar=x;
Fim Se
Fim Para
/*p é o pixel da imagem I em nível de cinza*/
Para Todo p ∈ I faça
    Se p < limiar então
        p=1;
    senão
        p=0;
    Fim Se
Fim Para Todo
FIM DO ALGORITMO

```

Algoritmo 5.2. Algoritmo de limiarização de Otsu e do processo de limiarização.

Algoritmo: Transformação H Min

Entrada: Imagem ImgIn, inteiro h, Elemento Estruturante E

Saída: Imagem ImgOut

Para Todo $p \in \text{ImgIn}$ e $b \in \text{ImgTmp}$ **faça**

$b \leftarrow p + h$;

Fim Para Todo

Para Todo $s \in \text{ImgOut}$ **faça**

$s \leftarrow \text{SupRec}(\text{ImgTmp}, \text{ImgIn}, E)$;

Fim Para Todo

FIM DO ALGORITMO

Algoritmo 5.3. Algoritmo da Transformação H-Mínima.

Algoritmo: Rotulação

Entrada: Imagem ImgIn

Saída: Imagem ImgOut

Rotulo $\leftarrow 0$;

Para Todo $p \in \text{ImgIn}$ **faça**

Se $p = 1$ e $\notin \text{Rotulados}$ **então**

 Insere p em Lista;

 Insere p em Rotulados;

Enquanto Lista $\neq$ vazia **faça**

 Remove b(x,y) de Lista;

$\text{ImgOut}(x,y) \leftarrow \text{Rotulo}$;

 Insere Vizinhos(b) em Lista;

 Insere Vizinhos(b) em Rotulados;

Fim Enquanto

Fim Se

Incremento Rotulo;

Fim Para Todo

FIM DO ALGORITMO

Algoritmo 5.4. Algoritmo de Rotulação.

Algoritmo: Rotação

Entrada: Imagem ImgIn

Saída: Imagem ImgOut

Para Todo $p(x,y) \in \text{ImgIn}$ **faça**

Se $q(x*\cos(\theta)+y*\sin(\theta), y*\cos(\theta)-x*\sin(\theta)) \in \text{ImgOut}$ **faça**

$q(x*\cos(\theta)+y*\sin(\theta), y*\cos(\theta)-x*\sin(\theta)) \leftarrow p(x,y);$

Fim Se

Fim Para Todo

FIM DO ALGORITMO

Algoritmo 5.5. Algoritmo de Rotação.

5.6. Detalhes da Segmentação

Nessa etapa, os pontos da cela Braille já se encontram rotulados, portanto, prontos para que se possa calcular o centro de cada componente conexa. Após a localização de cada centro da componente conexa, o ponto será representado por um único *pixel* (localizado no centro) que será utilizado como informação para separação das celas.

Como foi mostrada no Capítulo 2, a cela é composta de 3 linhas e duas colunas (Figura 5.8).

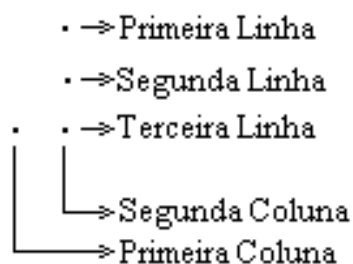


Figura 5.8. Figura demonstrativa da composição de uma cela Braille.

Encontra-se a menor e maior coordenada x da imagem Braille através da varredura do texto, para que sejam delimitadas as colunas da cela. Separando cada coluna do texto até seu ponto de coordenada x mais à direita (Figura 5.9).

Para que o programa identifique o início da cela, é necessário que a folha Braille digitalizada contenha no mínimo, um ponto pertencente à primeira coluna de alguma cela inicial em uma das linhas do texto. Isto é, primeiramente, é feita uma varredura em todo o texto para que seja encontrado o ponto que está mais à esquerda. Como a segmentação é baseada em distâncias, se este ponto não pertencer a primeira coluna de alguma cela, toda a segmentação estará comprometida.

Observando a primeira linha do texto Braille representado na Figura 5.9, é possível notar que não há nenhum ponto na primeira coluna da primeira cela Braille. A menor coordenada x dos pontos pertencentes a este exemplo seria a da primeira cela da segunda linha. Sendo a segunda linha usada como base inicial de separação de todas as celas Braille desse exemplo.

Esse critério de identificação das celas impossibilita que a tradução do texto seja feita em uma única linha. O símbolo Braille para letra maiúscula tem dois pontos na segunda coluna da cela (primeira cela da primeira linha do texto), e em geral, quando se escreve um texto, se inicia com letra maiúscula. Portanto, quanto maior o número de linhas do texto, maiores as chances da segmentação ser bem sucedida.

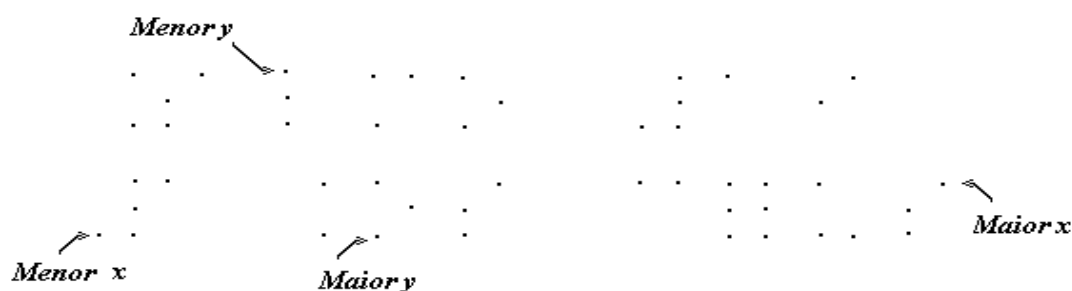


Figura 5.9. Centros das componentes conexas de duas linhas de uma imagem Braille digitalizada, onde estão representados os maiores e menores valores das coordenadas (x, y).

Após a realização da separação parcial do texto, em colunas (Figura 5.10), faz-se à separação total através do conhecimento da menor e da maior coordenada y (Figura 5.11), que também são obtidas através da varredura de todo o texto digitalizado. As distâncias entre os pontos que compõem uma cela irão determinar esta divisão.



Figura 5.10. Divisão da primeira coluna (divisão parcial das celas) pertencente ao texto representado na Figura 5.9.



Figura 5.11. (a) Segmentação da primeira cela Braille contida na figura 5.10; (b) Figura obtida a partir da segmentação da primeira cela da segunda linha do texto encontrada na Figura 5.10 (divisão total das celas).

O centro da componente conexa resulta nas coordenadas do *pixel* que a representa. As coordenadas x e y do *pixel* (x_{centro} , y_{centro}) terá o valor dado por:

$$x_{\text{centro}} = x_{\text{min}} + ((x_{\text{max}} - x_{\text{min}})/2);$$

$$y_{\text{centro}} = y_{\text{min}} + ((y_{\text{max}} - y_{\text{min}})/2).$$

Onde:

x_{centro} – é a coordenada x do *pixel* central que representa a componente conexa.

y_{centro} – é a coordenada y do *pixel* central que representa a componente conexa.

x_{min} – é a menor coordenada x do *pixel* pertencente a componente que se encontra mais a esquerda da folha digitalizada.

x_{max} – é a coordenada x do *pixel* pertencente a componente que se encontra mais a direita da folha digitalizada.

y_{min} – é a coordenada y do *pixel* pertencente a componente que se encontra mais acima da folha digitalizada.

y_{max} – é a coordenada y do *pixel* pertencente a componente que se encontra mais abaixo da folha digitalizada.

Algoritmo Usado na Segmentação

Algoritmo: Segmenta Cella

Entrada: Imagem ImgIn

Saída: Cella_i

Para Todo p, r ∈ ImgIn **faça**

Se p é início da coluna_i **faça**

 a ← coordenada horizontal de p;

Fim Se

```

    Se  $r$  é fim da coluna; faça
         $b \leftarrow$  coordenada horizontal de  $r$ ;
    Fim Se
Fim Para Todo
Para Todo  $p, r \in \text{ImgIn}$  faça
    Se  $p$  é início da linha; faça
         $c \leftarrow$  coordenada vertical de  $p$ ;
    Fim Se
    Se  $r$  é fim da linha; faça
         $d \leftarrow$  coordenada vertical de  $r$ ;
    Fim Se
     $\text{Cela}_k \leftarrow$  Imagem delimitada por  $a, b, c, d$ ;
Fim Para Todo

FIM DO ALGORITMO

```

Algoritmo 5.6. Algoritmo da Segmentação das Celas Braille.

5.7. Extração de Características

A extração de características consiste em verificar a imagem segmentada, ou seja, as celas, verificando a quantidade de *pixels* em cada cela, sendo que a área de cada uma delas é entendida como possuindo seis quadrantes. Cada cela pode ter sido segmentada sem conter nenhum *pixel*, o que é característica de espaço, ou também “pula linha”; e até seis *pixels* por cela, que corresponde à letra “é”. Se em um quadrante da cela tiver mais de um ponto, um deles será desconsiderado.

5.8. Reconhecimento e Interpretação

Essa é a etapa final do processamento de imagens. Para o reconhecimento é construída uma árvore de busca binária com 6 níveis (Figura 5.12), sendo que no último nível são anexados os caracteres alfanuméricos, os símbolos de pontuação e os números.

Cada nível da árvore representa presença ou ausência de um ponto pertencente à cela Braille. Esta estrutura de dados é carregada na memória do computador quando o programa é executado.

As matrizes obtidas após a extração de características compostas de zeros e uns são então, submetidas ao processo de busca através da árvore, da seguinte maneira:

- se zero (ou seja, se não existir o ponto Braille na cela), desce a árvore pela esquerda;
- se um (ou seja, se existir o ponto Braille na cela), desce pela direita.

Quando chega no último nível, a busca retorna os caracteres anexados à cela.

5.9. Fim do Processamento das Imagens Braille

Após a realização de todas as etapas necessárias ao processamento de imagem, conforme diagrama de blocos do capítulo anterior. Os textos, em caracteres Braille, são traduzidos para caracteres alfanuméricos (Figura 5.13). Sendo que a digitalização (rotação inferior a 2º) e o pré-processamento das imagens são etapas fundamentais para se chegar com sucesso às outras.

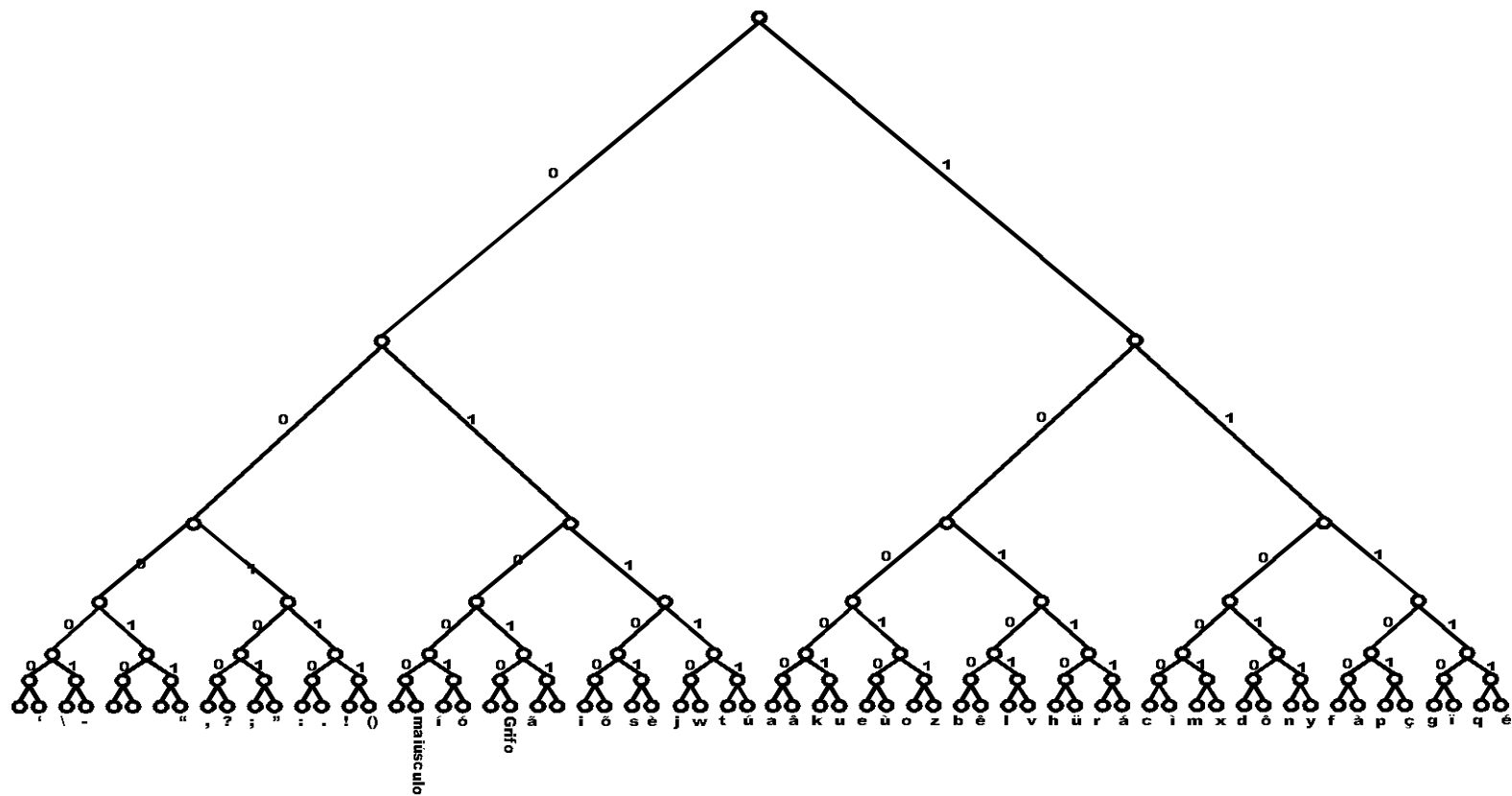


Figura 5.12. Árvore binária de busca da extração de características dos caracteres Braille

Salmo 19
1 Os céus declaram a gló-
ria de Deus e o firmamento|
anuncia a obra das suas
mãoè. 2 _um dia faz decla-
ração a outro dia, e uma
noite mostra sabedoria a
outra noite. 3 Não há
linguagem nem fala onde não

Figura 5.13. Texto traduzido a partir da imagem contida na Figura 5.2. Existe a presença de dois erros (| e _) de tradução, e um (mãoè) contido no próprio texto em Braille (folha original).

Capítulo 6 – Testes e Resultados

Neste capítulo, são apresentados alguns testes realizados com o programa tradutor BR Braille desenvolvido.

Os testes realizados foram feitos com folhas escritas com o uso de *reglete e punção* e de uma *impressora Braille* (vistos no Capítulo 2), e com três *scanners* de diferentes marcas.

Com as folhas escritas na impressora Braille, será possível se verificar a importância do algoritmo da rotação, pois é feita uma comparação da versão do BR Braille com o código da rotação desabilitado, e, com ele habilitado.

Com o uso do reglete e punção, serão realizadas as traduções mostrando o uso de três *scanners* para digitalizar a imagem em uma resolução de 200 dpi, além de realizar a digitalização em 100 dpi com o *scanner* de melhor resultado nestes testes.

De posse dos testes acima, será escolhido o *scanner* de melhor desempenho ao qual uma folha de texto em Braille será submetida a dez digitalizações diferentes a fim verificar o comportamento do BR Braille.

6.1. Uso de Diferentes *Scanners* com Folhas Escritas com Reglete

Neste teste, foram usadas quinze folhas escritas com reglete e punção e o processo de aquisição das imagens foi feito usando-se três *scanners* de marcas e modelos diferentes. Os textos em Braille foram feitos para preencher todo o espaço disponível da folha. Cada texto citado está contido em uma única folha. Foram usadas as mesmas folhas em todos os *scanners*.

A resolução utilizada na digitalização de todas as folhas nos três *scanners* foi de 200 dpi. Portanto, com mesma resolução e mesmos originais para a digitalização, tem-se que as condições de entrada são as mesmas para todos os equipamentos. Com isto, é possível verificar se há diferenças de desempenho do programa tradutor em relação aos equipamentos utilizados na aquisição das imagens Braille.

As Tabelas 6.1, 6.2 e 6.3 mostradas, a seguir, contém os dados das traduções realizadas.

Tabela 6.1. Dados dos processamentos do programa BR Braille utilizando o *scanner* da marca HP Scanjet 4400c de folhas escritas com reglete e punção.

SCANNER HP SCANJET 4400C				
Folhas	Número de Caracteres da Folha Original	Número de Caracteres do Texto Traduzido	Número de Erros da Tradução	Erros (%)
Deuteronômio 28:1-7	717	719	7	0,98
I João 1: 5-9	645	642	1	0,16
I João 5: 1-6	724	724	4	0,55
I Pedro 1: 3-8	723	723	6	0,83
I Samuel 2: 1-5	680	676	6	0,88
Jeremias 17: 5-8	604	603	2	0,33
Lucas 6: 43-48	672	667	6	0,89
Mateus 5:1-8	660	660	3	0,45
Mateus 6: 1- 5	737	737	2	0,27
Mateus 7: 7-14	730	730	2	0,27
Provérbios 3: 1-8	675	675	4	0,59
Provérbios 3: 13-20	694	689	5	0,72
Salmos 19:1-7	701	701	1	0,14
Salmos 25: 8-15	688	687	6	0,87
Salmos 34: 1-8	651	653	5	0,77

Tabela 6.2.Dados dos processamentos do programa BR Braille utilizando o scanner da marca Genius ColorPage-Life Pro de folhas escritas com reglete e punção.

SCANNER Genius ColorPage-Life Pro				
Folhas	Número de Caracteres da Folha Original	Número de Caracteres do Texto Traduzido	Número de Erros da Tradução	Erros (%)
Deuteronômio 28:1-7	717	717	2	0,28
I João 1: 5-9	645	645	3	0,46
I João 5: 1-6	724	724	3	0,41
I Pedro 1: 3-8	723	723	12	1,70
I Samuel 2: 1-5	680	680	7	1,03
Jeremias 17: 5-8	604	604	2	0,33
Lucas 6: 43-48	672	675	6	0,89
Mateus 5:1-8	660	668	11	1,67
Mateus 6: 1- 5	737	742	6	0,81
Mateus 7: 7-14	730	730	3	0,41
Provérbios 3: 1-8	675	675	3	0,44
Provérbios 3: 13-20	694	696	8	1,15
Salmos 19:1-7	701	708	5	0,71
Salmos 25: 8-15	688	689	3	0,44
Salmos 34: 1-8	651	651	8	1,23

Tabela 6.3. Dados dos processamentos do programa BR Braille utilizando o *scanner* da marca Boeder Sm@rtScan Slim Edition de folhas escritas com reglete e punção.

SCANNER Boeder Sm@rtScan Slim Edition				
Folhas	Número de Caracteres da Folha Original	Número de Caracteres do Texto Traduzido	Número de Erros da Tradução	Erros(%)
Deuteronômio 28:1-7	717	724	34	4,74
I João 1: 5-9	645	645	12	1,86
I João 5: 1-6	724	724	24	3,31
I Pedro 1: 3-8	723	728	31	4,29
I Samuel 2: 1-5	680	681	25	3,65
Jeremias 17: 5-8	604	607	16	2,45
Lucas 6: 43-48	672	673	31	4,61
Mateus 5:1-8	660	667	32	4,85
Mateus 6: 1- 5	737	735	34	4,61
Mateus 7: 7-14	730	730	9	1,23
Provérbios 3: 1-8	675	682	21	3,11
Provérbios 3: 13-20	694	693	27	3,89
Salmos 19:1-7	701	702	18	2,57
Salmos 25: 8-15	688	681	10	1,45
Salmos 34: 1-8	651	651	26	3,99

As Tabelas 6.1, 6.2 e 6.3 contêm o título dos textos em Braille, os número de caracteres dessas folhas originais, o número de caracteres dos textos traduzidos, e o número de erros da tradução, e o correspondente em porcentagem.

Sendo que os erros em porcentagem são dados por:

$$Erros (\%) = \frac{Número\ de\ Erros}{Número\ de\ Caracteres\ da\ Folha\ Original} \quad (1)$$

O número de erros considerado é soma dos caracteres que são trocados mais os caracteres adicionados e/ou excluídas durante o processo de tradução do texto.

$$Média\ dos\ Erros\ (\%) = \frac{\sum Erros(\%)}{Número\ de\ Folhas} \quad (2)$$

A média de erros (%) obtida com os dados da Tabela 6.1 é de $0,58 \pm 0,08$ %, sendo que para todas as folhas digitalizadas neste *scanner* a porcentagem de erro foi menor que 1%. Com o *scanner* da Tabela 6.2, a média de erros foi de $0,80 \pm 0,12$ %, e o da Tabela 6.3 de $3,37 \pm 0,31$ %.

O texto “Mateus 5”, da folha Mateus 5:1-8 digitalizada no *scanner* da Tabela 6.3, apresentou 4,85% de erro, que foi a maior porcentagem obtida durante o processo de tradução (Figura 6.1).

Para verificar se as diferentes marcas influenciavam no número de erros (%) da tradução, foi realizado uma análise de variância monofatorial pareada destes dados. O nível de significância estatística foi estabelecido para $P < 0,01$. Além disso, para comparar o desempenho destes *scanners* entre si, foram feitos testes de contraste por meio do método de comparações múltiplas de Student-Newman-Keuls.

Os resultados indicaram que o equipamento de digitalização influencia diretamente no número de erros (%) de tradução ($P<0,001$). O teste de contraste indicou que não há diferenças significativas entre o *scanner* HP Scanjet 4400c e o Genius ColorPage-Life Pro, porém ambos apresentam uma eficiência significativamente maior que o Boeder Sm@rtScan Slim Edition ($P<0,001$).

Mateus 5
Vendo Jesus as multidões,
subiu ao monte, e, como se
assentasse, aproximara-se
os seus discípulos; 2 e
Ele passou a ensiná-los
dizendo: 3 Bem-aventura-
dos os humildes de espírito
porque deles é o reino dos
céus. 4 Bem-aventurados
os que choram, porque serão
consolados. 5 Bem-aven-
turados os mansos, porque
herdarão a terra.
6 Bem-aventurados os que
têm fome e sede de justiça,
porque serão fartos.
7 Bem-aventurados os mi-
sericordiosos, porque alcan-
çarão misericórdia.
8 Bem-aventurados os lim-
pos de coração, porque ve-
rão a Deus. 9 Bem-aven-
turados os pacificadores,
porque serão chamados
filhos de Deus.

Figura 6.1. Texto traduzido pelo BR Braille da folha (Mateus 5:1-8) digitalizada no *scanner* da Tabela 6.3. As marcas em vermelho indicam as letras traduzidas com erro, sendo que a letra marcada em amarelo corresponde a erro na folha Braille original.

6.2. Programa Com e Sem o Código da Rotação

Neste item, será mostrada a importância da rotação que faz parte do pré-processamento. As Tabelas 6.4 e 6.5 mostram os dados dos processamentos realizados com o código de rotação habilitado e desabilitado respectivamente.

Tabela 6.4. Dados dos processamentos do programa BR Braille completo utilizando o *scanner* da marca HP modelo Scanjet 4400c para digitalizar folhas escritas na impressora Braille.

SCANNER HP SCANJET 4400C				
Código da Rotação Habilitado				
Folhas	Tempo (s)	Número de Caracteres da Folha Original	Número de Erros da Tradução	Erros(%)
Deuteronômio 28:1-7	284	743	6	0,81
I João 1: 5-9	246	680	12	1,74
I João 5: 1-6	259	722	13	1,80
I Pedro 1: 3-8	291	787	26	3,3
I Samuel 2: 1-5	293	755	10	1,32
Jeremias 17: 5-8	200	592	10	1,69
Lucas 6: 43-48	229	667	10	1,50
Mateus 5:1-8	213	640	4	0,63
Mateus 6: 1- 5	274	754	17	2,25
Mateus 7: 7-12	251	674	20	2,97
Provérbios 3: 1-8	248	684	19	2,78
Provérbios 3: 13-20	281	710	16	2,25
Salmos 19:1-7	282	716	13	1,81
Salmos 25: 8-15	259	723	5	0,69
Salmos 34: 1-8	249	680	0	0

Tabela 6.5. Dados dos processamentos do programa BR Braille com rotação desabilitada o utilizando o *scanner* da marca Scanner HP Scanjet 4400c para digitalizar folhas escritas na impressora Braille.

SCANNER HP SCANJET 4400C				
Código da Rotação Desabilitado				
Folhas	Tempo (s)	Número de Caracteres da Folha Original	Número de Erros da Tradução	Erros (%)
Deuteronômio 28:1-7	58	743	5	0,67
I João 1: 5-9	53	680	11	1,62
I João 5: 1-6	58	722	12	1,66
I Pedro 1: 3-8	59	787	29	3,68
I Samuel 2: 1-5	54	625	370	59,2
Jeremias 17: 5-8	45	592	11	1,86
Lucas 6: 43-48	50	684	151	22,1
Mateus 5:1-8	49	640	4	0,63
Mateus 6: 1- 5	56	754	17	2,25
Mateus 7: 7-12	49	699	633	91,13
Provérbios 3: 1-8	52	685	20	2,92
Provérbios 3: 13-20	54	604	405	67,05
Salmos 19:1-7	56	626	381	60,86
Salmos 25: 8-15	56	723	29	4,01
Salmos 34: 1-8	55	680	10	1,47

Este mesmo texto processado no programa BR Braille completo torna-se legível (Figura 6.3), com um erro de 2,97 %, sendo que o erro apresentado nesta mesma folha processado sem a rotação gerou um erro de 91,13% (texto apresentado na Figura 6.2).

Mateus 7

7 Pedi, e dar-se-vos-á;
buscai e achareis; batei, e
abrir-vos-á. 8 Pois todo o
que pede recebe; o que busca
encontra; e, a quem, bate,
abrir-se-lhe-á. 9 Ou qual
dentre vós é o homem que, se
forventura k filho lhe pedir
pão, lhe dará pedra? 10 Ou se
se lhe pedir um peixe, lhe
dará uma cobra? 11 Ora, se
vós, que sois maus, sabeis dar
boas dádivas aos vossos
filhos, quanto mais vosso
Pai, que está nos céus, dará
boas coisas aos que lhe
pedirem? 12 São quanto
vós, quereis que os homens
vós façam, assim fazei-os vós
também a eles; porque esta é a
lei e os profetas. Entrai
pela porta estreita (larga é a

Figura 6.3. Texto “Mateus 7” (folha Mateus 7: 7-12) processado com o programa BR Braille com o código de rotação habilitado. As marcas vermelhas indicam onde estão as letras não foram corretamente traduzidas.

Nos textos onde as linhas não estão inclinadas, a habilitação ou não do código de rotação não faz diferença. Como exemplo pode-se observar o texto Deuteronômio 28:1-7 (Tabelas 6.4 e 6.5) cujos erros foram semelhantes nas duas versões. Com auxílio da teste t de Student, e considerando $P < 0,01$ como nível de significância estatística, foi verificada que a utilização do código de rotação no programa BR Braille influencia diretamente na qualidade do texto traduzido ($P < 0,001$).

6.3. Resoluções Utilizadas para Digitalização do Programa

Este item tem como finalidade mostrar a diferença entre o processamento dos textos que foram digitalizados em duas diferentes resoluções, 100 e 200 dpi. O programa está limitado a funcionar com estas duas resoluções.

Nas Tabelas 6.6 e 6.7, é interessante analisar o tempo, dado em segundos, e o número de erros gerados pelo processamento.

Tabela 6.6. Dados do processamento do programa BR Braille utilizando para digitalização de folhas escritas com reglete e punção, o *scanner* da marca Scanner HP Scanjet 4400c com resolução igual a 100 dpi.

SCANNER HP SCANJET 4400C				
Folhas	Resolução	Tempo (s)	Número de Caracteres da Folha Original	Número de Erros da Tradução
Deuteronômio 28:1-7	100	71	725	0
I João 1: 5-9	100	59	645	1
I João 5: 1-6	100	81	723	0
I Pedro 1: 3-8	100	71	728	2
I Samuel 2: 1-5	100	72	682	0
Jeremias 17: 5-8	100	55	608	0
Lucas 6: 43-48	100	64	667	2
Mateus 5:1-8	100	63	660	0
Mateus 6: 1- 5	100	71	742	1
Mateus 7: 7-14	100	70	736	1
Provérbios 3: 1-8	100	64	682	0
Provérbios 3: 13-20	100	67	696	0
Salmos 19:1-7	100	76	708	0
Salmos 25: 8-15	100	76	689	0
Salmos 34: 1-8	100	71	674	1

Tabela 6.7. Dados do processamento do programa BR Braille utilizando para digitalização de folhas escritas com reglete e punção, o *scanner* da marca Scanner HP Scanjet 4400c com resolução igual a 200 dpi.

SCANNER HP SCANJET 4400C				
Folhas	Resolução	Tempo (s)	Número de Caracteres da Folha Original	Número de Erros
Deuteronômio 28:1-7	200	283	719	7
I João 1: 5-9	200	241	642	1
I João 5: 1-6	200	298	724	4
I Pedro 1: 3-8	200	293	723	6
I Samuel 2: 1-5	200	261	676	6
Jeremias 17: 5-8	200	221	603	2
Lucas 6: 43-48	200	281	667	6
Mateus 5:1-8	200	257	660	3
Mateus 6: 1- 5	200	288	737	2
Mateus 7: 7-14	200	319	730	2
Provérbios 3: 1-8	200	256	675	4
Provérbios 3: 13-20	200	282	689	5
Salmos 19:1-7	200	277	701	1
Salmos 25: 8-15	200	283	687	6
Salmos 34: 1-8	200	262	653	5

Embora estes dois valores de resolução (100 e 200 dpi) sejam possíveis, se for desejado um tempo de processamento menor, recomenda-se o uso de 100 dpi na digitalização das imagens. Como pode ser visto na Tabela 6.6 o número de erros resultantes com a resolução de 100 dpi também é menor do que com 200 dpi, isto porque a rotulação (etapa de pré-processamento) é feita através de um percentual da resolução, portanto, com uma resolução maior pode-se obter um número maior de exclusões indesejáveis de *pixels*.

As Figuras 6.4 e 6.5 mostram qual é a diferença entre um texto traduzido com 0 % de erro e este mesmo texto com 0,28 % de erro.

Salmos 19
1 Os céu proclamam a glória de Deus, e o firmamento anuncia as obras das suas mãos. 2 Um dia discursa a outro dia, e uma noite revela conhecimento a outra noite. 3 Não há linguagem nem há palavras, e deles não se houve nenhum som; 4 no entanto, por toda a terra se faz ouvir a sua voz, e as suas palavras até aos confins do mundo. Aí, pôs uma tenda para o sol, 5 o qual, como noivo que sai dos seus aposentos, se regozija como herói, a percorrer o seu caminho. 6 Principia numa extremidade dos céus, e até à outra vai o seu percurso; e nada refoge ao seu calor. 7 A lei do Senhor é perfeita, e restaura a alma; o testemunho fo Senhor é fiel e dá sabedoria aos símplices.

Figura 6.4 . Texto traduzido pelo BR Braille (digitalização 1- Tabela 6.8), com 0 % de erro, sendo que a letra marcada corresponde a erro na folha Braille original.

Salmos 19

1 Os cé[] proclamam a gló-
ria de Deus, e o firmamen-
to anuncia as obras das qu-
as mãos. 2 Um dia discor-
sa a outro dia, e uma noite
revela conhecimento a outra
noite. 3 Não há linguagem
nem há palavras, e deles
não se houve nenhum som;
4 no entanto, por toda a
terra se faz ouvir a sua
voz, e as suas palavras até
aos confins do mundo. Aí,
pôs uma tenda para o sol,
5 o qual, como noivo que
sai dos seus aposentos, se
regozija como herói, a per-
correr o seu caminho. 6
Principia numa extremidade
dos céus, e até à outra vai
o seu percurso; e nada refo-
ge ao seu calor. 7. A lei
do Senhor é perfeita, e
restaura a alma; o testemu-
nho fo Senhor é fiel e dá
sabedoria aos simplices.

Figura 6.5 . Texto traduzido pelo BR Braille (digitalização 6- Tabela 6.8), com 0,28 % de erro, sendo que a letra marcada em amarelo corresponde a erro na folha Braille original e as letras em vermelho, os erros causados pelo programa tradutor.

Portanto, com o *scanner* HP Scanjet 4400c utilizado, ou com um de igual qualidade, verificando-se todas as restrições adotadas em relação à folha original utilizada, e tendo o cuidado de alinhar a mesma ao retângulo formado pelo vidro do *scanner*, consegue-se uma tradução com uma velocidade aproximada de 60 segundos e com um número de erros bastante pequeno. Isto pode ser confirmado pelos resultados mostrados na Tabela 6.8, onde

são mostradas dez digitalizações do mesmo texto Braille (folha Salmos 34: 1-8), utilizando a resolução de 100 dpi.

Tabela 6.8. Erros de tradução do programa BR Braille utilizando o *scanner* da marca Scanner HP Scanjet 4400c com resolução igual a 100 dpi para 10 digitalizações de uma mesma folha escrita com reglete e punção.

SCANNER HP SCANJET 4400C		
<i>Folha: Salmo 19</i>		
<i>Resolução: 100dpi</i>		
Digitalização	Número de Erros	Erros (%)
1	0	0,00
2	1	0,14
3	1	0,14
4	0	0,00
5	1	0,14
6	2	0,28
7	1	0,14
8	2	0,28
9	0	0,00
10	1	0,14

Capítulo 7 – Conclusões e Trabalhos Futuros

Neste capítulo são retomados o tema e o objetivo deste trabalho, bem como as soluções propostas e executadas e os resultados obtidos. Na seção 7.1 são brevemente abordados os temas de cada capítulo do texto. Na seção 7.2 são apresentadas as conclusões do trabalho desenvolvido. Na seção 7.3 são discutidas as contribuições resultantes deste trabalho, e na seção 7.4 são apresentadas sugestões de trabalhos futuros.

7.1. Sobre o Texto

Na introdução, foi apresentado o problema abordado pelo trabalho, que foi a realização de um programa de computador para ajudar o deficiente visual alfabetizado em Braille, a se comunicar, através da escrita, com pessoas que não conhecem este sistema de leitura e escrita. Ou seja, ampliar a integração do cego, seja no ambiente escolar ou no doméstico, facilitando o seu dia-a-dia.

Foi mostrado, na introdução, que o programa é de fácil utilização além de não ser necessário nenhum equipamento sofisticado, requerendo, somente, para sua utilização um micro-computador (configuração mínima: Pentium 100 com 32 M de RAM) e um *scanner*.

Além disto, algumas recomendações são importantes no momento da digitalização: a maneira adequada da colocação do papel, com o texto Braille contido no vidro do *scanner* e a seleção da área efetivamente abrangida pelo texto Braille, retirando as eventuais marcas causadas pela reglete e bordas com ruído. Vale ressaltar ainda a importância da preservação dos pontos Braille.

No Capítulo 2 o Sistema Braille foi apresentado, com uma breve explanação da história do Braille, e dos caracteres alfanuméricos que compõem o chamado alfabeto Braille.

No Capítulo 3, foram descritas algumas características encontradas no *scanner* bem como alguns formatos de imagem, principalmente os encontrados nos *scanners*, mostrando assim, que o formato BMP é mais adequado para sua utilização pelo BR Braille.

Dando a base teórica sobre o processamento de imagem digital, que fornece a sequência da utilização dos algoritmos que foram implementados na confecção do programa BR Braille, tem-se o quarto capítulo.

O quinto capítulo mostra os algoritmos utilizados e também uma explicação da importância da utilização de cada etapa do processamento das imagens Braille digitalizadas.

E finalmente, no sexto capítulo, foram realizados diversos testes divididos em três etapas. A primeira etapa tinha como objetivo principal verificar se o tipo do *scanner* influenciava no resultado obtido pelo BR Braille. Como foi observado que um dos *scanners* apresentou um resultado superior aos demais, partiu-se então para a segunda etapa cujo objetivo era comprovar se havia ou não a necessidade de implementar a rotação na fase de pré-processamento e a diferença de tempo de processamento para imagens digitalizadas com 100 e 200 dpi. Em sua terceira etapa foram obtidos resultado do BR Braille quando aplicado a um único texto que foi digitalizado dez vezes o que deu subsidio para concluir que o BR Braille cumpriu de forma bastante satisfatória o seu objetivo.

7.2. Conclusões

Em relação à proposta do trabalho, o resultado obtido foi satisfatório, inclusive, no teste utilizando o *scanner* em que os erros obtidos foram os mais significativos, o texto ainda assim ficava bastante legível.

Observou-se ainda que o desempenho do *software* está diretamente relacionado ao modo como a folha Braille foi digitalizada e ao seu estado de conservação. Desta forma, se forem seguidas todas as recomendações no processo de digitalização, consegue-se obter uma tradução legível, possibilitando que as pessoas não alfabetizadas em Braille possam ler os textos escritos em Braille em papel, pelos cegos.

O equipamento de escrita utilizado pelo portador de deficiência não interfere na qualidade dos resultados obtidos: o que vai influenciar é o quanto de inclinação as linhas do texto possam apresentar. Para isto, é importante somente que se tenha o cuidado ao escrever, na colocação das folhas ou na manutenção dos mecanismos de inserção do papel

nos equipamentos de escrita Braille. Além do mais, é necessário, para que a tradução continue legível, que o relevo dos pontos Braille esteja preservado.

Quanto ao tempo de processamento, ele não é o ideal para um texto digitalizado com 200 dpi, chegando a atingir mais de 4 minutos como foi mostrado no Capítulo 6. Portanto recomenda-se utilizar imagens digitalizadas em 100 dpi, onde o tempo de processamento do programa BR Braille não atinge 60 segundos.

O processamento digital de imagens Braille foi realizado em etapas ou módulos, como por exemplo, o pré-processamento e a segmentação. É possível se introduzir uma modificação para complementar ou melhorar o programa, sem trabalho excessivo, pois fica mais simples saber onde estão acontecendo os maiores problemas: como por exemplo, fazer a alteração da rotação para que o algoritmo aceite maiores inclinações das linhas, ou a otimização do tempo gasto no pré-processamento, alterando ou substituindo principalmente o operador da transformação h-mínima. Isto indica um caminho a se seguir para futuras melhorias nas próximas versões do programa.

7.3. Contribuições

Uma das contribuições é que se atingiu o objetivo de ampliar a possibilidade de comunicação da pessoa cega com pessoas que não lêem em Braille, dando a elas maior autonomia no seu dia-a-dia.

Uma outra contribuição é que, embora exista uma tecnologia similar no mercado internacional, ela não é voltada para o português do Brasil e, além do mais, o BR Braille será um programa *freeware*. Dessa forma, o programa pode ter um grande alcance em sua utilização, pois permite seu uso sem nenhum ônus e também é possível sua utilização com equipamentos usualmente encontrados em ambientes escolares e domésticos, de custo acessível.

A possibilidade de realizar a tradução de textos Braille digitalizados em qualquer *scanner* é uma outra vantagem do BR Braille frente aos OBRs importados, cujo uso é restrito a determinados modelos e marcas de scanners. Isto certamente contribuirá para sua aceitação pelos usuários.

7.4. Sugestões de Trabalhos Futuros

Para melhoria da velocidade de processamento, é possível realizar a alteração do algoritmo utilizado para a transformação H-Mínima, por um algoritmo de otimização como a Transformada Imagem Floresta H-Mínima (IFT H-Mínima).

Com relação à rotação é possível modificar o método utilizado para o cálculo do ângulo, para que o programa funcione adequadamente com uma maior inclinação das linhas Braille, ou do papel como um todo.

Também, ainda em relação à rotação, pode-se utilizar outro cálculo diferente do Método dos Mínimos Quadrados.

Pode-se ainda realizar estudos da viabilidade de se implementar outros algoritmos de rotação que diminuam o tempo total de processamento do programa, já que como visto no Capítulo 6, esta etapa provoca uma perda de velocidade no desempenho do programa.

E por fim, é necessário acompanhar a implementação do uso do BR Braille numa escola em que alunos cegos estejam inseridos, e também num ambiente doméstico, para avaliar a facilidade de uso e a aceitação do programa.

Bibliografia

A HISTÓRIA DE LOUIS BRAILLE. *Disponível em:* <URL: http://www.sac.org.br/APR_BR1.htm> Acesso em: 19 de novembro de 2002.

Araújo Júnior, F.P. (1996). *Um Algoritmo Morfológico para Processamento Fotomosaico*. Tese de Mestrado. IMECC – UNICAMP, Campinas

Carvalho, J.O.F. (2001). *Soluções Tecnológicas para Viabilizar o Acesso do Deficiente visual à Educação a Distância no Ensino Superior*. Tese de Doutorado. FEEC-UNICAMP, Campinas.

Charnet R., Freire, C.A.L., Charnet E.M.R., Bonvino H.. (1999). *Análise de Modelos de Regressão Linear - com Aplicações*, Editora UNICAMP.

Facon, J. (1993). *Processamento e Análise de Imagens*. In: *Processamento e Análise de Imagens*, VI Escola Brasileiro-Argentina de Informática.

Foley, J.D.; van Dam A.; Feiner, S. K. e Hughes, J.F. (1990). *Computer Graphics – Principles and Practice*, Addison-Wesley Publishing Company, Inc.

1FORMATOS DE IMAGEM. *Disponível em:*<URL:<http://mega.ist.utl.pt/~ic-cg/programa/livro/FormatosdeImagem.pdf>> Acesso em: 07 de dezembro 2002.

FUNDAÇÃO HILTON ROCHA (1987). *Ensaio sobre a Problemática da Cegueira – Prevenção, Recuperação e Reabilitação*, ed. Senado Federal, Brasília – D.F.

Gomes, J. e Velho, L. (1994). *Computação Gráfica: Imagem*, Série de Computação e Matemática - IMPA/SBM.

Gonzalez, R. C. e Woods, R. E. (2000). *Processamento de Imagens Digitais*, Edgard Blücher Ltda.

Haralick, R. M. e Nadadur, D. (2000). *Recursive Binary Dilation and Erosion Using Digital Line Structuring Elements in Arbitrary Orientations*. IEEE Transactions on Image Processing, vol.9, nº 5, p. 749-759, Maio.

Jesus, E. O. (1997). *SRIDE - Sistema Reconhecedor de Imagens de Diagramas Elétricos*, Tese de Doutorado, DCA/FEEC/UNICAMP, Campinas.

Lindley, C.A. (1991). *Practical Image Processing in C - Aquisition, Manipulation and Storage*, John Wiley and Sons, Inc.

Marques Filho, O. e Neto Vieira, H. (1999). *Processamento Digital de Imagens*, Brasport Livros e Multimídia LTDA.

Mennens, J.; Van Tichelen, L.; Francois, G.; e Engelen, J. (1993). *Optical recognition of Braille writing*, J. Document Analysis and Recognition, Proceedings of the Second International Conference on, p. 428 -431.

MOVING THEORY INTO PRATICE – DIGITAL IMAGE TUTORIAL. *Disponível em:* <URL: <http://www.library.cornell.edu/preservation/tutorial/intro/intro-04.html> > Acesso em: 06 de dezembro de 2002.

MOVING THEORY INTO PRATICE – DIGITAL IMAGE TUTORIAL. *Disponível em:* <URL: <http://www.library.cornell.edu/preservation/tutorial/intro/intro-05.html> > Acesso em: 06 de dezembro de 2002.

Nascimento, A. M .R.; Bruno, M. M. G.; Rímoli, O. B.; e Cordeiro, Y. M. C.. (1987). *O Deficiente Visual na Classe Comum*, Coordenadoria de Estudos e Normas Pedagógicas, Secretaria da Educação do Estado de São Paulo – São Paulo.

Netravali, A. N. e Haskell, B. G. (1988). *Digital Pictures - Representation and Compression*, Plenum Press, N.Y.

O SISTEMA BRAILLE. *Disponível em:* <URL: http://www.sac.org.br/APR_BR1.htm>
Acesso em: 19 de novembro de 2002.

Otsu, N. (1979). *A Threshold Selection Method from Gray-Level Histograms*, IEEE Transactions on System, Man, and Cybernetics, vol. SMC-9, nº 1, p. 62-66, January.

Post, F. H. e Hin, A. J. S. (1992). *Advances in Scientific Visualization*, p: 145-157, editora: Springer-Verlag.

Queiros, G. A .F. (1997). *Técnicas de Processamento e Análise de Imagens Aplicadas à Avaliação Morfológica e Morfométrica de Axônios Mielínicos*, Dissertação de Mestrado, DCA/FEEC/UNICAMP, Campinas.

SCANNERS, *Disponível em:* < URL: <http://www.geoc.ufpr.br/~gfoto/pdi/scanner>> Acesso em: 06 de dezembro de 2002.

Secretaria de Educação Especial. *Disponível em:* <URL:<http://www.mec.gov.br/seesp/plan4.shtm>> Acesso em: 18 de dezembro 2002.

Secretaria de Educação Especial. *Disponível em:* <URL:<http://www.mec.gov.br/seesp/plan9.shtm>> Acesso em: 18 de dezembro 2002.

Soares, R. C. (2001). *Estudo de Código de Barras por Análise de Imagens*, Dissertação de Mestrado, DCA/FEEC/UNICAMP, Campinas.

Soille, P. (1999). *Morphological Image Analysis - Principles and Applications*, Editora: Springer-Verlag Berlin Heidelberg.

Shih, F. Y. C. e Mitchell, O. R. (1989). *Threshold Decomposition of Gray-Scale Morphology into Binary Morphology*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, nº 1, p. 31-42, January.

Shih, F. Y. C. e Mitchell, O. R. (1992). *A Mathematical Morphology Approach to Euclidian Distance Transformation*, IEEE Transactions on Image Processing, vol.1, nº 2, p. 197-204, Abril.

THE MATHWORKS INC.. *The Language of Technical Computing, versão 6.1.0.450 Release 12.1, 2001 - "Toolbox de Processamento de Imagem"*. Conjunto de 2 CD-ROMs. Necessário: Windows 2000/ NT.

Torres, G. (2001). *Hardware - Curso Completo*, Axcel Books do Brasil Editora.

White, R.. (1993). *Como Funciona o Computador*, Ziff-Davis Press.

ANEXOS

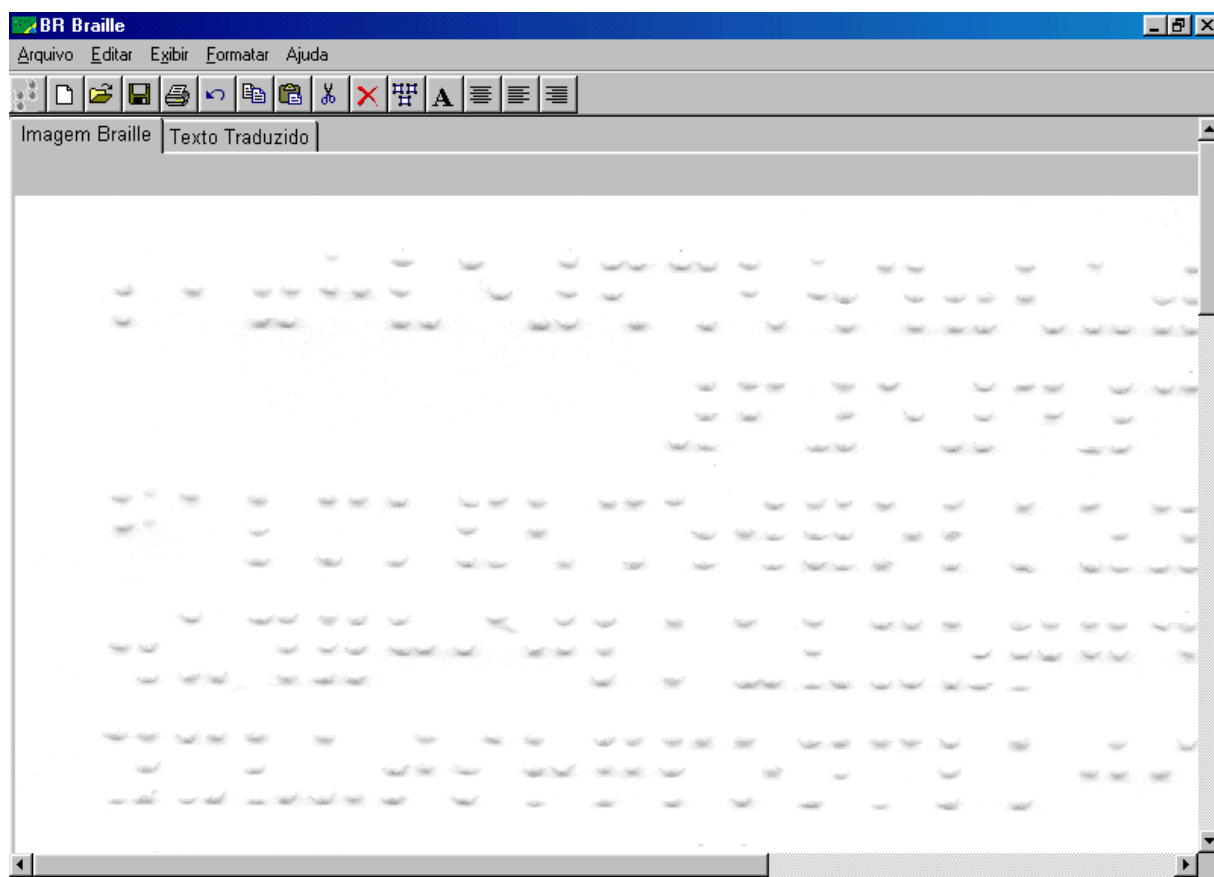
Anexo 1

Logotipo do Programa Tradutor de Textos Braille para Caracteres Alfanuméricos em Português
BR Braille



Anexo 2

Tela do Programa Tradutor de Textos Braille para Caracteres Alfanuméricos em Português BR Braille



Anexo 3

Explicação dos Ícones da Tela do Programa BR Braille



Anexo 4

Instruções para o Uso do Programa BR Braille

1. Digitalizar a folha escrita em Braille com os pontos da cela Braille (alto relevo) voltadas para o vidro do *scanner*. Colocar as folhas de forma que a mesma toque uma das laterais de marcação do vidro do *scanner*, mas tomando o cuidado de que todas as celas Braille estejam inseridas no vidro.
2. Salvar a imagem em BMP.
3. Com a utilização do Paint Brush retirar possíveis pontos de bordas causados pelo reglete e/ou manchas causada por problemas de iluminação do *scanner*.
4. Abrir o programa BR Braille.
5. Utilizar o ícone “Abrir” para a escolha do arquivo que contém a imagem Braille digitalizada.
6. Clicar com o *mouse* no ícone “Traduzir”.
7. Para a próxima página a ser traduzida utilize os seguintes ícones (verificar se foram realizados passos de 1 a 3 para esta folha):
 - 7.1. Selecionar todo o texto.
 - 7.2. Limpar o texto.
 - 7.3. Selecione a Imagem Braille.
 - 7.4. Novo