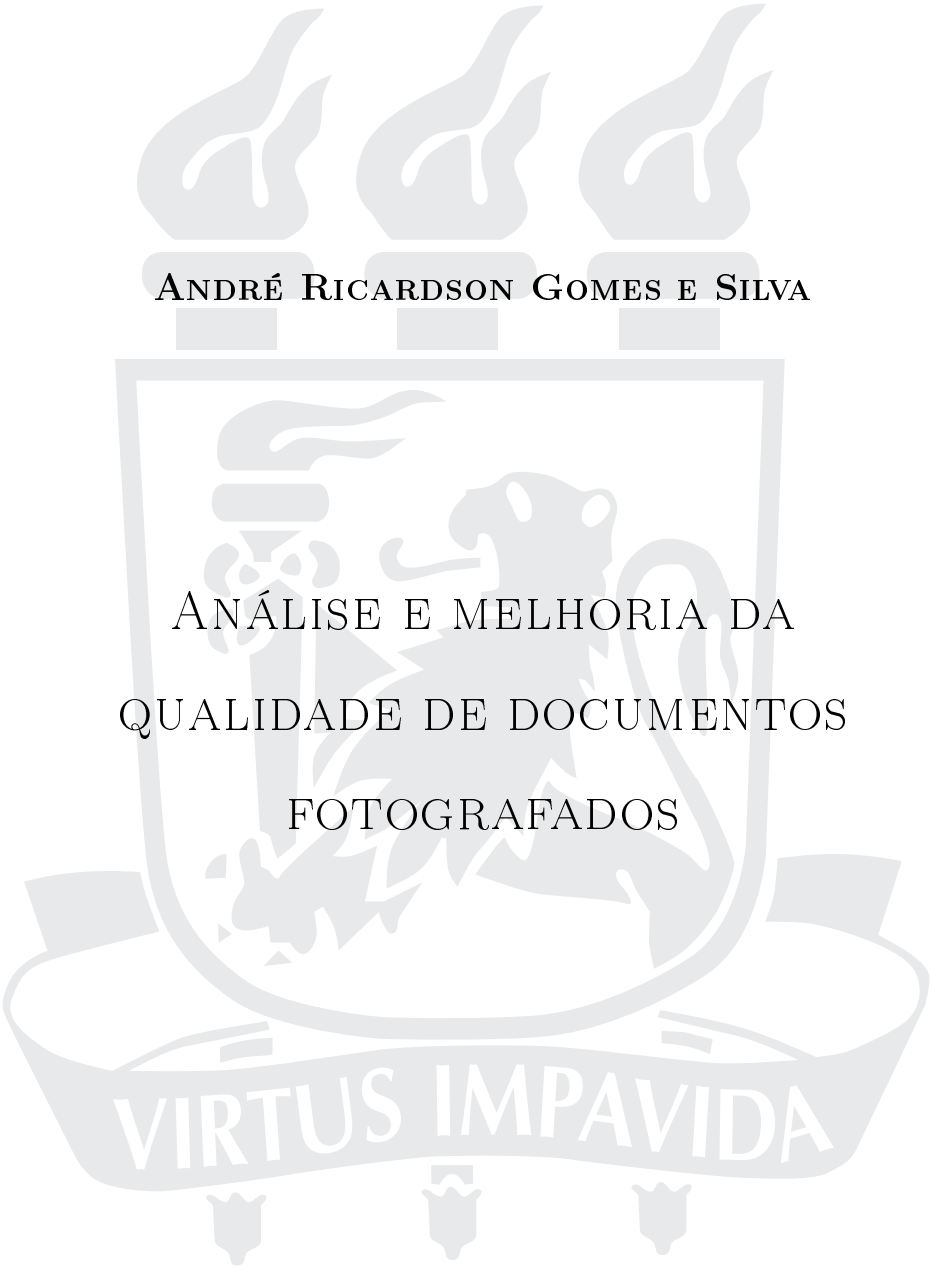
Universidade Federal de Pernambuco
Centro de Tecnologia e Geociências
Programa de Pós-graduação em Engenharia Elétrica



André Ricardson Gomes e Silva

Análise e melhoria da qualidade de documentos fotografados

Recife, Novembro de 2006.

André Ricardson Gomes e Silva

Análise e melhoria da qualidade de documentos fotografados

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco como parte dos requisitos para obtenção do grau de **Mestre em Engenharia Elétrica**

Orientador: Prof. Rafael Dueire Lins, Ph.D.

Recife, Novembro de 2006.

S586a **Silva, André Ricardson Gomes e.**
Análise e melhoria da qualidade de documentos fotografados. - Recife: O Autor, 2006.
136 folhas. : il. ; fig., tabs.

Dissertação (Mestrado) Universidade Federal de Pemambuco. CTG. Engenharia Elétrica, 2006.

Inclui bibliografia.

1. Engenharia elétrica. 2. Algoritmos - Desenvolvimento. 3. Documentos digitalizados - Análise. 4. Fotografia digital. 5. Documentos fotografados - Melhoria da qualidade. I. Título.

621.3 CDD (22.ed.)

UFPE
BCTG/2007 -014

Universidade Federal de Pernambuco

Pós-Graduação em Engenharia Elétrica

PARECER DA COMISSÃO EXAMINADORA DE DEFESA DE TESE DE MESTRADO ACADÊMICO DE

ANDRÉ RICARDSON GOMES E SILVA

TÍTULO

"ANÁLISE E MELHORIA DA QUALIDADE DE DOCUMENTOS FOTOGRAFADOS"

A comissão examinadora composta pelos professores: RAFAEL DUEIRE LINS, DES/UFPE, RICARDO MENEZES CAMPELLO DE SOUZA, DES/UFPE e FRANCISCO MADEIRO BERNARDINO JÚNIOR, DEI/UNICAP, sob a presidência do primeiro, consideram o candidato **ANDRÉ RICARDSON GOMES E SILVA** **APROVADO**.

Recife, 22 de novembro de 2006.

JOAQUIM FERREIRA MARTINS FILHO
Coordenador do PPGEE

RAFAEL DUEIRE LINS
Orientador e Membro Titular Interno

FRANCISCO MADEIRO BERNARDINO JÚNIOR
Membro Titular Externo

RICARDO MENEZES CAMPELLO DE SOUZA
Membro Titular Interno

Agradecimentos

Aqui expresso meus sinceros agradecimentos às pessoas que contribuíram direta e indiretamente para o desenvolvimento dessa dissertação. Em especial agradeço:

- à minha família, que sempre me apoiou incondicionamente em todos os momentos, em especial à minha mãe Terezinha Gomes da Silva
- ao Prof. Rafael Dueire Lins pela aceitação como orientado no mestrado, por me mostrar o valor do trabalho científico, pela motivação, pela disponibilidade, e principalmente por ter acreditado em mim no desenvolvimento desse trabalho
- à Banca Examinadora composta pelos professores Rafael Dueire Lins, Ricardo Campello de Souza e Francisco Madeiro Bernardino Júnior pelas contribuições no desenvolvimento dessa dissertação através de suas sugestões
- a Gabriel França Silva pela ajuda e sugestões
- aos amigos da pós-graduação, que compartilharam as dificuldades nas disciplinas e no desenvolvimento dessa dissertação. Em especial Andrei Formiga, Bruno Ávila, João Marcelo Silva e Márcio Lima pela contribuição direta no desenvolvimento desse trabalho.

André Ricardson Gomes e Silva

Universidade Federal de Pernambuco

22 de Novembro de 2006

Resumo da Dissertação apresentada à UFPE como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica

Análise e melhoria da qualidade de documentos fotografados

André Ricardson Gomes e Silva

Novembro/2006

Orientador: Prof. Rafael Dueire Lins, Ph.D.
Área de Concentração: Comunicações
Palavras-chaves: câmeras digitais, análise de imagens de documentos, correção de distorção, OCR.
Número de páginas: 137

A crescente disponibilidade de alto desempenho, baixo preço e portabilidade tem tornado as câmeras digitais uma alternativa viável para a aquisição de imagens de documentos. Câmeras digitais são fáceis de utilizar e transportar. Hoje, telefones celulares, palm tops, e outros dispositivos eletrônicos já incorporam câmeras digitais. Devido à sua rápida disseminação é urgente o desenvolvimento de técnicas e algoritmos capazes de processar documentos, melhorando a qualidade da imagem, possibilitando a sua melhor leitura, armazenamento, rápida transmissão via redes de computadores, transcrição automática, etc. Imagens capturadas através de câmeras digitais portáteis podem perder o foco, possuem distorções de perspectiva e distorções causadas pela lente, assim como composição complexa da imagem e interação com o conteúdo e plano de fundo, gerando o surgimento de bordas. As técnicas utilizadas em documentos adquiridos com *scanners* tradicionais nos servem como referência, mas não podem ser utilizadas diretamente em imagens adquiridas através de câmeras digitais.

Esta dissertação analisa alguns dos fatores influentes na qualidade de documentos digitalizados através do uso de câmeras fotográficas digitais portáteis e apresenta algoritmos capazes de corrigir alguns desses fatores.

Abstract of Dissertation presented to UFPE as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering

Analysis and improvement of the quality of photographed documents

André Ricardson Gomes e Silva

November/2006

Supervisor: Prof. Rafael Dueire Lins, Ph.D.
Area of Concentration: Communications
Keywords: digital cameras, document image analysis, distortion correction, OCR
Number of pages: 137

The increasing availability, high performance, low price, and portability have made digital cameras a viable alternative for acquisition of document images. Today, digital cameras are found in mobile telephones, palm tops, and other portable electronic devices. Due to their fast dissemination it is urgent the development of techniques and algorithms capable of processing such images of documents, improving their quality, increasing their readability, decreasing the space needed for storage, allowing their fast transmission through computer networks, automatic transcription, etc. The techniques used in traditional scanner-based documents may be used as a reference, but they cannot be used directly in digital camera-acquired images. Images captured through portable cameras have distortions of perspective and distortions caused by the lens. The interaction between the document and its mechanical support cause the appearance of borders, etc.

This dissertation analyzes some of the factors that have influence on the quality of documents digitalized through the use of portable digital cameras and introduces algorithms capable of correcting some of those factors.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

dpi	pontos por polegada (*dots per inch*)
BMP	Formato de armazenamento de imagens (*Bitmap*)
JPEG	Formato de armazenamento de imagem (*Joint Photographic Experts Group*)
R	Componente vermelha da cor do pixel
G	Componente verde de cor do pixel
B	Componente azul de cor do pixel
RGB	Sistema de representação de cores (*Red-Green-Blue*)
OCR	Reconhecimento Óptico de Caracteres (*Optical Character Recognition*)
CCD	Dispositivo de Carga Acoplada (*Charged Coupled Device*)
CCI	Sensor de Contato de Imagem (*Contact Image Sensor*)
Mpixel	megapixels, um megapixel corresponde a um milhão de pixels

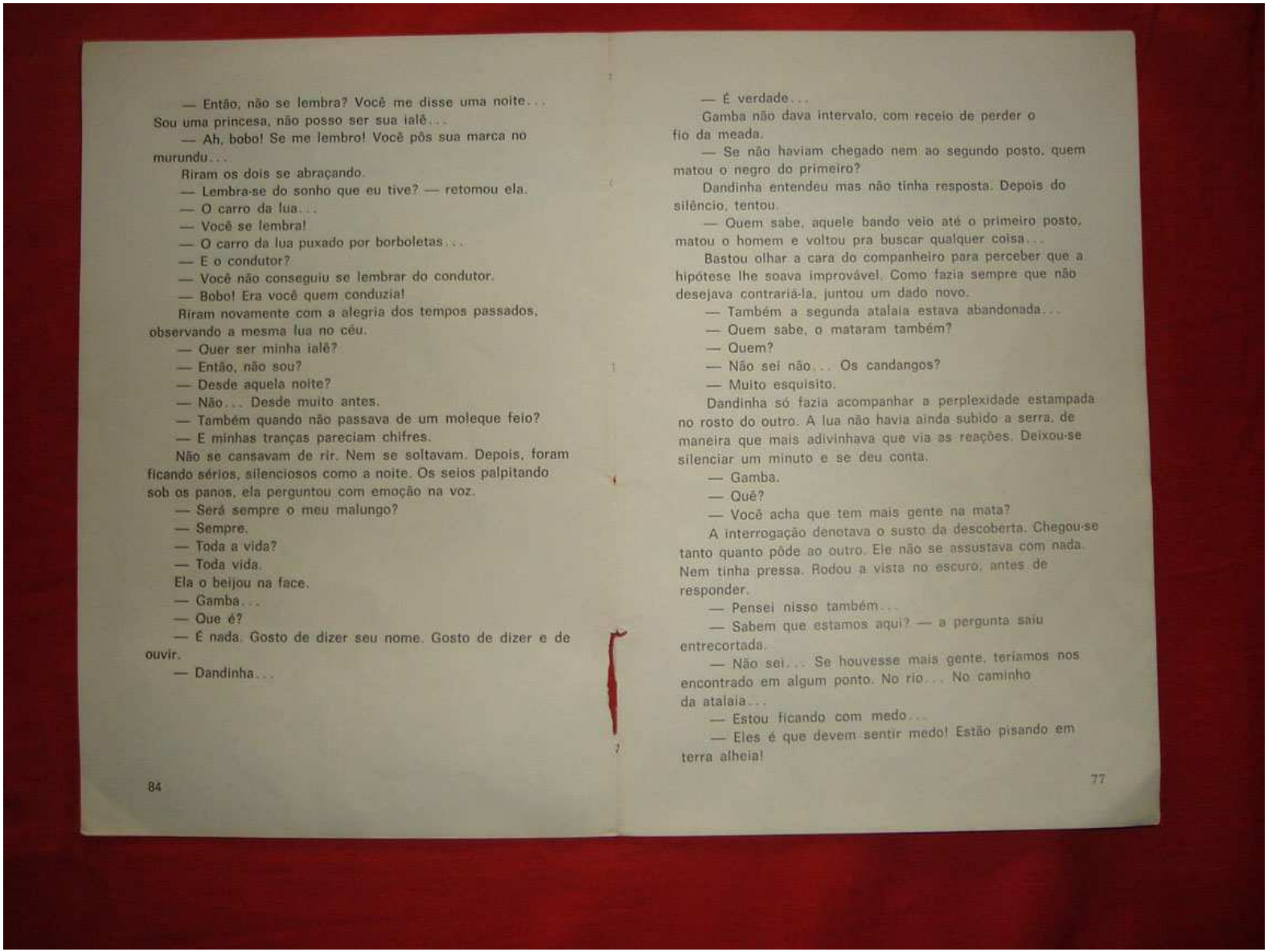
Sumário

capítulo 1

Introdução

A ampla difusão das câmeras digitais portáteis causada pela queda dos preços e grande aumento da qualidade das imagens geradas por esses dispositivos ocasionou a sua utilização em aplicações não originalmente previstas: profissionais e estudantes de diferentes áreas passaram a utilizar câmeras para efetuar digitalização de documentos. Tais documentos vão desde a imagem de um quadro branco com anotações escritas em sala de aula por um professor a processos legais consultados por advogados em fóruns. Entretanto, em virtude de câmeras digitais não terem sido projetadas para digitalização de documentos, sua aplicação nesse âmbito possui limitações. O uso já difundido de tal aplicação fez surgir uma nova área de pesquisa, que avança para a solução dos diferentes problemas característicos da aquisição de imagens por esses dispositivos. O estudo para esse tipo de imagem assemelha-se ao estudo realizado em imagens adquiridas por *scanners*, mas apresenta características e problemáticas próprias. Embora cada solução tenha dificuldades específicas, todos estes visam prover câmeras capazes de produzir imagens de fácil leitura. A Figura 1.1 apresenta um documento fotografado à mão livre, sem suporte mecânico para a câmera, caso que vem se tornando comum no ambiente de fotografia de documentos.

É possível observar que o documento apresentado na Figura 1.1 é legível para a maioria das pessoas, embora os caracteres sejam pequenos. Há a presença de borda escura circundando o documento, ligeira inclinação em relação a base da imagem, iluminação irregular, ligeiramente mais clara ao centro, e curvatura nas suas extremidades. A Figura 1.2 apresenta o mesmo documento da Figura 1.1 ocupando toda a área útil da página, tendo sua borda sido removida e perspectiva corrigida. Devido ao aumento da resolução do documento na página, percebe-se



— Então, não se lembra? Você me disse uma noite... Sou uma princesa, não posso ser sua ialê...

— Ah, bobo! Se me lembro! Você pôs sua marca no murundu...

Riram os dois se abraçando.

— Lembra-se do sonho que eu tive? — retomou ela.

— O carro da lua...

— Você se lembra!

— O carro da lua puxado por borboletas...

— E o condutor?

— Você não conseguiu se lembrar do condutor.

— Bobo! Era você quem conduzia!

Riram novamente com a alegria dos tempos passados, observando a mesma lua no céu.

— Quer ser minha ialê?

— Então, não sou?

— Desde aquela noite?

— Não... Desde muito antes.

— Também quando não passava de um moleque feio?

— E minhas tranças pareciam chifres.

Não se cansavam de rir. Nem se soltavam. Depois, foram ficando sérios, silenciosos como a noite. Os seios palpitando sob os panos, ela perguntou com emoção na voz.

— Será sempre o meu malungo?

— Sempre.

— Toda a vida?

— Toda vida.

Ela o beijou na face.

— Gamba...

— Que é?

— É nada. Gosto de dizer seu nome. Gosto de dizer e de ouvir.

— Dandinha...

84

— É verdade...

Gamba não dava intervalo, com receio de perder o fio da meada.

— Se não haviam chegado nem ao segundo posto, quem matou o negro do primeiro?

Dandinha entendeu mas não tinha resposta. Depois do silêncio, tentou.

— Quem sabe, aquele bando veio até o primeiro posto, matou o homem e voltou pra buscar qualquer coisa...

Bastou olhar a cara do companheiro para perceber que a hipótese lhe soava improvável. Como fazia sempre que não desejava contrariá-la, juntou um dado novo.

— Também a segunda atalaia estava abandonada...

— Quem sabe, o mataram também?

— Quem?

— Não sei não... Os candangos?

— Muito esquisito.

Dandinha só fazia acompanhar a perplexidade estampada no rosto do outro. A lua não havia ainda subido a serra, de maneira que mais adivinhava que via as reações. Deixou-se silenciar um minuto e se deu conta.

— Gamba.

— Quê?

— Você acha que tem mais gente na mata?

A interrogação denotava o susto da descoberta. Chegou-se tanto quanto pôde ao outro. Ele não se assustava com nada. Nem tinha pressa. Rodou a vista no escuro, antes de responder.

— Pensei nisso também...

— Sabem que estamos aqui? — a pergunta saiu entrecortada.

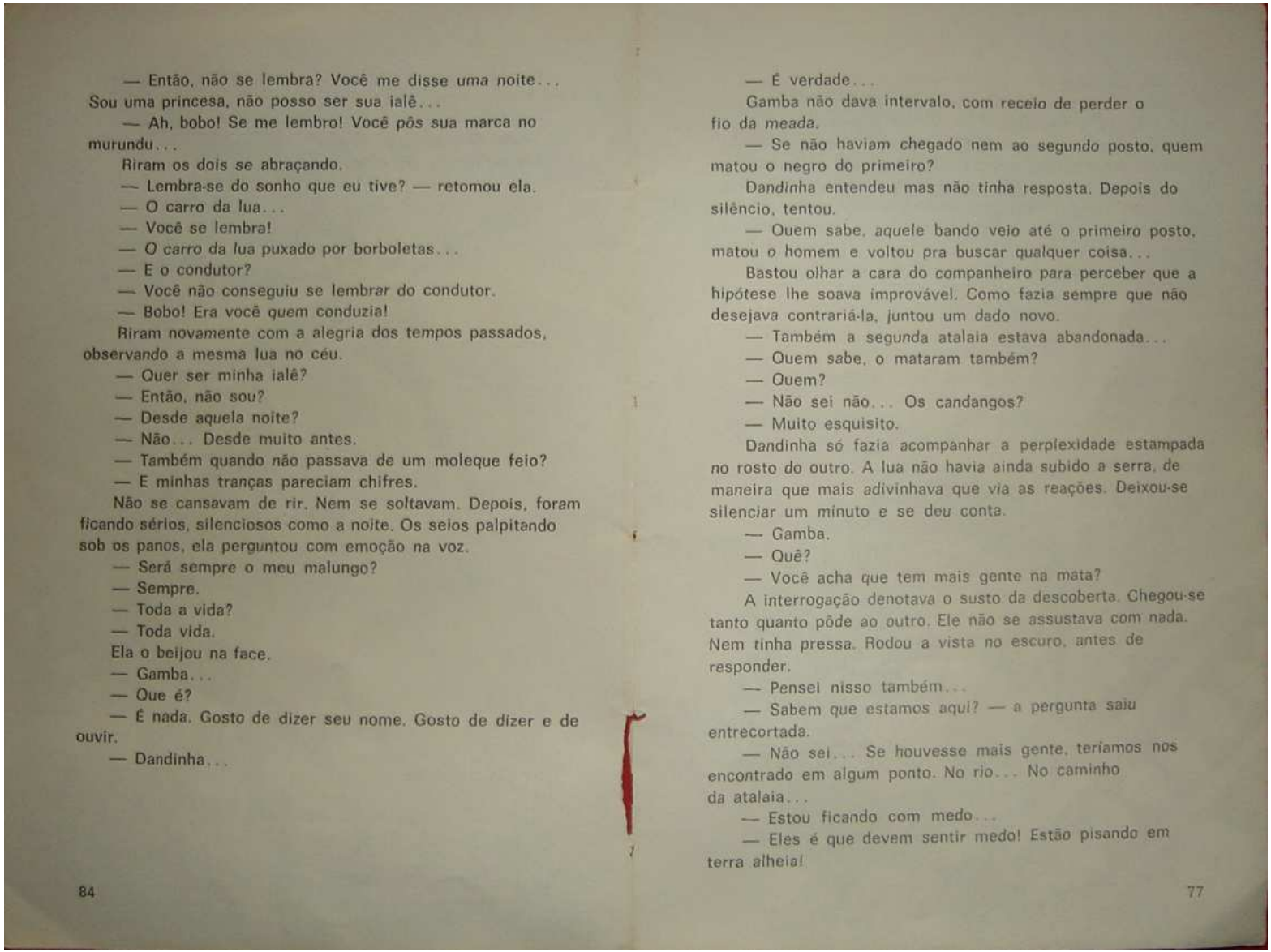
— Não sei... Se houvesse mais gente, teríamos nos encontrado em algum ponto. No rio... No caminho da atalaia...

— Estou ficando com medo...

— Eles é que devem sentir medo! Estão pisando em terra alheia!

77

Figura 1.1: Ilustração de documento fotografado à mão livre.



— Então, não se lembra? Você me disse *uma noite*... Sou uma princesa, não posso ser sua ialê...

— Ah, bobo! Se me lembro! Você *pôs* sua marca no murundu...

Riram os dois se abraçando.

— Lembra-se do sonho que eu tive? — retomou ela.

— O carro da lua...

— Você se lembra!

— O carro da lua puxado por borboletas...

— E o condutor?

— Você não conseguiu se lembrar do condutor.

— Bobo! Era você *quem* conduzia!

Riram novamente com a alegria dos tempos passados, observando a mesma lua no céu.

— Quer ser minha ialê?

— Então, não sou?

— Desde aquela noite?

— Não... Desde muito antes.

— Também quando não passava de um moleque feio?

— E minhas tranças pareciam chifres.

Não se cansavam de rir. Nem se soltavam. Depois, foram ficando sérios, silenciosos como a noite. Os seios palpitando sob os panos, ela perguntou com emoção na voz.

— Será sempre o meu malungo?

— Sempre.

— Toda a vida?

— Toda vida.

Ela o beijou na face.

— Gamba...

— Oue é?

— É nada. Gosto de dizer seu nome. Gosto de dizer e de ouvir.

— Dandinha...

— É verdade...

Gamba não dava intervalo, com receio de perder o fio da *meada*.

— Se não haviam chegado nem ao segundo posto, quem matou o negro do primeiro?

Dandinha entendeu mas não tinha resposta. Depois do silêncio, tentou.

— Quem sabe, aquele bando veio até o primeiro posto, matou o homem e voltou pra buscar qualquer coisa...

Bastou olhar a cara do companheiro para perceber que a hipótese lhe soava improvável. Como fazia sempre que não desejava contrariá-la, juntou um dado novo.

— Também a segunda atalaia estava abandonada...

— Quem sabe, o mataram também?

— Quem?

— Não sei não... Os candangos?

— Muito esquisito.

Dandinha só fazia acompanhar a perplexidade estampada no rosto do outro. A lua não havia ainda subido a serra, de maneira que mais adivinhava que via as reações. Deixou-se silenciar um minuto e se deu conta.

— Gamba.

— Quê?

— Você acha que tem mais gente na mata?

A interrogação denotava o susto da descoberta. Chegou-se tanto quanto pôde ao outro. Ele não se assustava com nada. Nem tinha pressa. Rodou a vista no escuro, antes de responder.

— Pensei nisso também...

— Sabem que estamos aqui? — a pergunta saiu entrecortada.

— Não sei... Se houvesse mais gente, teríamos nos encontrado em algum ponto. No rio... No caminho da atalaia...

— Estou ficando com medo...

— Eles é que devem sentir medo! Estão pisando em terra alheia!

Figura 1.2: Documento da Figura 1.1 após recorte de borda e correção de perspectiva.

que a Figura 1.2 é mais legível que a Figura 1.1. Além disso, o tamanho da imagem reduziu de 325KB para 263KB, ambas utilizando compressão JPEG com perda e fator de qualidade $Q = 50$. A impressão da Figura 1.2 consumiu 77,3% a menos de toner que a Figura 1.1, considerando-se impressão em preto-e-branco, uma vez que o número de pixels escuros foram reduzidos nessa proporção.

No caso de documentos burocráticos (documentos tipografados tipicamente utilizados em escritórios) é imprescindível que os caracteres da imagem sejam legíveis, e pode-se encontrar grande quantidade de aplicações que requerem a transposição do formato de imagem para o formato de texto editável. O Reconhecimento Óptico de Caracteres (OCR - *Optical Character Recognition*) é um aplicativo desenvolvido para transcrever imagens de textos escritos em textos editáveis em linguagem de máquina ou traduzir imagens de caracteres em um esquema padrão de codificação que os tenha em representação (ASCII ou Unicode). Reconhecimento de caracteres é uma das principais aplicações a que se destinam os documentos digitalizados, tendo sua pesquisa ativa desde o final dos anos 1950. Inicialmente o desenvolvimento desta aplicação foi tido como simples, entretanto estima-se que muitas décadas ainda se passarão antes que o computador possa ter a mesma precisão dos seres humanos. O problema no reconhecimento de caracteres em documentos tipografados adquiridos com técnicas de digitalização que permitam alta resolução é tido largamente como um problema resolvido, enquanto os problemas envolvendo documentos adquiridos com baixa resolução, como é o caso da maioria das câmeras digitais, ainda são temas muito ativos em pesquisas.

Muitas variáveis podem influenciar nas dificuldades do processo de obtenção de imagens digitais de boa qualidade, como: parâmetros de digitalização (brilho, contraste, resolução, número de cores, entre outros), inclinação do documento, distorções geométricas e de perspectiva, perda de foco, entre outros. Com base nessas variáveis, pesquisadores têm implementado modelos de degradação de documentos [3]. Documentos digitalizados de baixa qualidade requerem nova digitalização, e esta definição de qualidade satisfatória ou insatisfatória é comumente determinada de forma subjetiva por operadores responsáveis pela manipulação destas imagens, tornando este processo dispendioso e falho. Pesquisas são realizadas tanto para medir a qualidade de imagens digitalizadas quanto para a melhoria da qualidade, seguindo fatores variados e dependentes dos critérios do pesquisador.

1.1 Objetivos

Nesta dissertação tem-se por objetivo o desenvolvimento de algoritmos visando a melhoria das imagens de documentos fotografados utilizando-se câmeras digitais portáteis. São analisados algoritmos para binarização (transformação de imagens coloridas ou monocromáticas (escala de cinza) em imagens binárias) e propostos algoritmos para remoção de bordas e correção de perspectiva. Estes resultados buscam prover melhoria na qualidade de documentos fotografados, e abrir caminho para a solução de outros fatores indesejados causados durante a aquisição de documentos por câmeras digitais portáteis. Serão feitas análises comparativas com as técnicas de processamento de imagens adquiridas através de *scanners*, visto que essas técnicas encontram-se largamente estudadas e o *scanner* ainda é a principal fonte de aquisição de documentos em formato digital. Também serão feitas análises das características próprias de documentos fotografados e a adequação de métodos utilizados em imagens de um modo geral, assim como aplicação de filtros.

Será apresentado também um estudo da qualidade de transcrição de documentos digitalizados via câmeras fotográficas digitais portáteis por OCR, para desta forma, estimar-se a qualidade de caracteres e identificar os principais fatores influentes nesta medição.

1.2 Organização da dissertação

A dissertação está dividida em seis capítulos, sendo o primeiro esta introdução.

O capítulo 2 descreve sumariamente os princípios do funcionamento de *scanners* e câmeras fotográficas digitais, citando as vantagens e os problemas ocorridos durante a digitalização de documentos em ambos dispositivos.

O capítulo 3 apresenta um algoritmo para remoção da borda que serve de suporte mecânico para documentos fotografados. A remoção de tal borda faz-se importante principalmente para a aplicação de outros algoritmos a esse documento e redução do espaço para armazenamento e transmissão.

No capítulo 4, realiza-se um estudo comparativo entre documentos digitalizados através de *scanners* e câmeras fotográficas digitais em diferentes resoluções, buscando-se estimar a qualidade dos documentos digitalizados pelas câmeras, baseando-se em transcrições providas por uma ferramenta comercial de OCR.

O capítulo 5 apresenta alguns algoritmos clássicos de binarização, a aplicação desses algo-

ritmos a documentos fotografados e análise desses documentos após binarização, através de transcrição por uma ferramenta comercial de OCR.

No capítulo 6, apresenta-se um novo algoritmo para correção de perspectiva de imagens de documentos burocráticos adquiridos por câmeras digitais.

O capítulo 7 apresenta uma visão geral das contribuições desta dissertação, bem como linhas de continuidade ao trabalho aqui apresentado.

capítulo 2

Dispositivos para digitalização de documentos

Este capítulo introduz a motivação e mostra como ocorre a aquisição de imagens de documentos através de dispositivos. Isto inclui a introdução de alguns dispositivos e discussões sobre vantagens em relação a *scanners*, aplicações e desafios para análise de documentos digitalizados através do uso de câmeras digitais. A seção que segue apresenta alguns conceitos para melhor compreensão desta dissertação.

2.1 Imagens e documentos

Documentos são definidos como uma representação de informação armazenada ou registrada de alguma forma. Esta informação pode estar presente em um papel, uma tela ou tocada por um auto-falante ou algum outro dispositivo de saída e sua essencial representação pode ser de qualquer forma e incluir dados de qualquer meio [50]. Documentos independentes podem ser compostos, daí uma rede de informação pode também ser considerada um documento.

Documentos burocráticos são documentos em papel, constituídos essencialmente por texto (impresso, manuscrito ou ambos), podendo apresentar tabelas e figuras, sendo esses os documentos analisados nesta dissertação por compor quase 100% dos documentos de escritórios.

Imagens podem ser descritas como toda e qualquer visualização gerada pelo ser humano, enquanto imagens digitais são representações em duas dimensões de uma imagem, constituídas

por elementos finitos (ou pontos) chamados pixels[1].

Documentos digitalizados são imagens digitais criadas a partir de documentos, de forma que seja possível a reprodução total ou parcial da informação contida neles.

2.2 Processamento de documentos

O processo de digitalização de documentos burocráticos é hoje efetuado através do uso de *scanners*. Esta é uma área já bem estudada e serve de comparação com os métodos e problemáticas encontrados nos processamentos de imagens adquiridas pela utilização de câmeras fotográficas digitais portáteis; entretanto, na maior parte dos casos, os algoritmos desenvolvidos para processamento de imagens de documentos digitalizados por *scanners* não podem ser aplicados diretamente a documentos adquiridos através de câmeras digitais portáteis com resultados satisfatórios. Os documentos digitalizados por *scanners* são normalmente papéis impressos ou documentos escritos à mão, como jornais, cartas, folhetos, entre outros. Assume-se que parte substancial da imagem seja texto, embora desenhos e figuras possam fazer parte da composição dessa. Portanto, a digitalização de tais documentos constitui uma fonte de informação para análises comparativas com os mesmos documentos digitalizados por câmeras digitais portáteis.

Os trabalhos relacionados a textos adquiridos por câmeras e análise de documentos podem ser caracterizados de três formas: de acordo com o tipo de texto analisado, com a aplicação a ser utilizada, ou mesmo pelo tipo de dispositivo utilizado [16]. A maioria dos trabalhos relacionados a imagens de documentos adquiridos por câmeras tem sido realizada na área de "processamento de imagem", embora também existam trabalhos em "imagens de documentos estruturados".

O processamento de imagens de texto é uma aplicação específica que busca reconhecer texto como parte de um conteúdo visual. O objetivo geral é prover capacidade de capturar informação destinada à comunicação visual humana e utilizá-la para diferentes tarefas.

Inicialmente projetadas para aquisição de imagens complexas (como fotografias de pessoas, paisagens e animais), devido à sua popularização, as câmeras digitais portáteis têm sido utilizadas para diferentes atividades, como a fotografia de quadros-negro em sala de aula por alunos, ou de um documento histórico durante visita a um museu, ou no dia-a-dia de advogados na aquisição de documentos em fóruns de advocacia. A utilização de câmeras para

[1] O nome pixel é uma abreviação do termo em inglês *picture element*.

digitalização de documentos constitui um meio rápido para a aquisição e disseminação de informação, possibilitando e motivando a sua aplicação na digitalização de documentos.

2.3 A digitalização de documentos por *scanners*

Scanners digitais têm sido os dispositivos dominantes na aquisição de imagens de documentos na última década e muitos novos tipos de *scanners* foram desenvolvidos para várias aplicações envolvendo documentos. Tipicamente os *scanners* são encontrados em quatro formas:

- *Scanners* de mesa são o tipo mais versátil e comumente utilizado, encontrados em escritórios e residências. Neste tipo de *scanner* a digitalização ocorre em um documento por vez. Sua descrição mais detalhada será apresentada neste capítulo.
- *Scanners* com alimentação automática são semelhantes aos *scanners* de mesa, exceto que neste tipo de *scanner* os documentos movem-se, enquanto o mecanismo de digitalização permanece fixo.
- *Scanners* de mão utilizam basicamente a mesma tecnologia dos *scanners* de mesa, mas depende do usuário para mover o mecanismo de digitalização sobre o documento. Tipicamente não produz imagens de boa qualidade, entretanto podem ser úteis na captura rápida de textos.
- *Scanners* de produção utilizam mecanismos mais complexos para aquisição das imagens, buscando melhor qualidade além de velocidade. Normalmente são utilizados para digitalização de grandes acervos bibliográficos.

Basicamente, *scanners* de mesa são compostos de uma lâmpada, lentes, espelhos, filtros e um sensor[2], que pode ser uma matriz CCD (Dispositivo de Carga Acoplada) ou um sensor de contato (CIS - *contact image sensor*), responsável por captar as imagens. No processo de captura, o documento é posto sobre uma superfície de vidro, e a tampa do *scanner* é fechada (vide Figura 2.1). Esta tampa provê um plano de fundo uniforme que os aplicativos baseados no *scanner* podem ter como referência para identificação dos limites do documento. Uma lâmpada é utilizada para iluminar o documento, enquanto o mecanismo completo (espelhos, lentes, filtros e sensor) se move, passando por toda a superfície de vidro. A imagem do

[2]Um sensor pode ser aqui definido como um dispositivo eletrônico que detecta a reflexão da luz no área que será transformada em imagem.

documento é refletida por espelhos, cujo último espelho reflete a imagem em lentes, esta lente foca a imagem através de um filtro no sensor. Tal processo é ilustrado na Figura 2.2 e na Figura 2.3, onde a seta vermelha indica o sentido de deslocamento do mecanismo, e o documento encontra-se com a face para baixo.

Figura 2.1: Ilustração de um scanner *de mesa.*

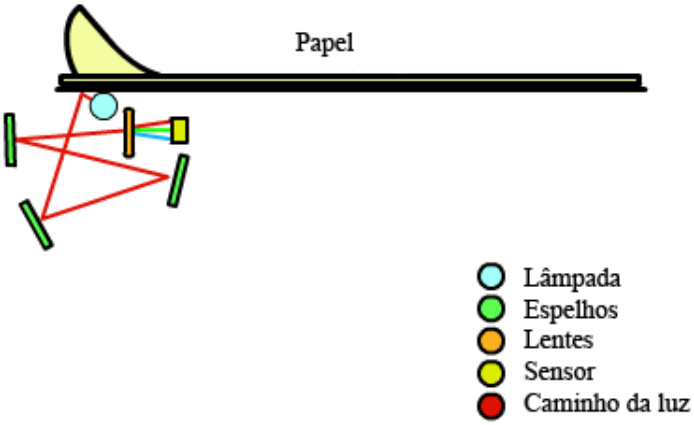


Figura 2.2: Processo de digitalização por scanners.

A velocidade dos *scanners* pode ser tanto alta, digitalizando várias páginas por segundo, como lenta, requerendo interação do usuário a cada página. Tal velocidade é dependente também do tamanho do documento e da resolução[3] utilizada para digitalização. Robôs podem

[3]Resolução é o termo adotado para definir o número de pixels que compõem a imagem por unidade de comprimento (pixels por centímetro, pixels por polegada, etc.).

Figura 2.3: Ilustração da digitalização por scanner.

ser utilizados, tornando possível a mudança de página de um livro automaticamente. Com resoluções que ultrapassam 2400dpi e baixo preço, o *scanner* tornou-se muito popular entre os usuários de computadores pessoais. As imagens adquiridas por tais *scanners* são mais adequadas para o processamento de documentos, dado que o ambiente de digitalização é bem conhecido e pode-se controlar facilmente sua resolução. Visto que *scanners* adquirem adequadamente imagens digitais mesmo de páginas degradadas, estes não são tipicamente considerados como um problema significante no processo de análise de documentos.

2.4 A digitalização de documentos por câmeras fotográficas digitais

Assim como câmeras fotográficas convencionais, as câmeras fotográficas digitais também possuem um conjunto de lentes responsáveis por focar a luz para criação de imagens. Mas ao invés de armazenar a luz em um filme, as câmeras fotográficas digitais armazenam eletronicamente. Tal processo é possível através do uso de um ou mais sensores, responsáveis pela conversão de luz em cargas elétricas. Os sensores mais comumente utilizados são CCD.

Uma das dificuldades na aquisição de imagens por câmeras fotográficas digitais ocorre no processo de armazenamento das cores. São três os métodos mais utilizados:

- No primeiro método são utilizados diferentes sensores para as cores vermelho, verde e azul (componentes do sistema RGB). Tal processo é eficiente, entretanto as câmeras que o utilizam possuem alto custo;

- No segundo método um separador de cores envia uma cor por vez para o sensor, no entanto a imagem precisa estar parada por um período maior, inviabilizando sua utilização na maioria das aplicações;
- No terceiro método utiliza-se apenas um sensor, onde cada ponto da imagem no CCD recebe uma cor, posteriormente são realizadas interpolações para definição da cor final do ponto na imagem, tal processo pode ser observado na Figura 2.4. A distribuição mais comum destas cores ao longo da imagem é o filtro de Bayer, ilustrado na Figura 2.5. Este é o método utilizado na maioria das câmeras fotográficas digitais.

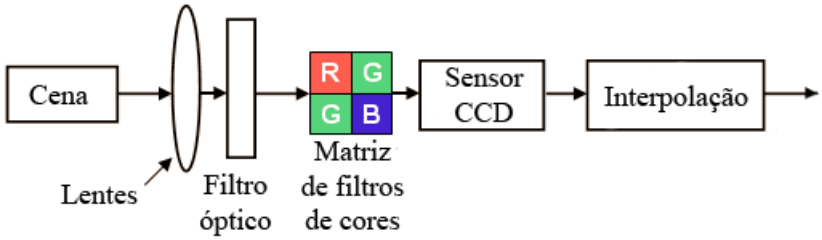


Figura 2.4: A formação das imagens em câmeras fotográficas digitais portáteis.

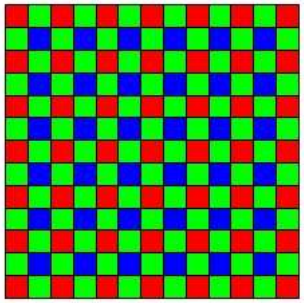


Figura 2.5: Filtro de Bayer.

Na digitalização de documentos a interpolação pode borrar o texto devido à atenuação das altas freqüências, levando à redução da qualidade, assim como aumento nos erros de reconhecimento de caracteres [9].

O processo de captura de câmeras de vídeo digitais e câmeras fotográficas digitais é bem semelhante, sendo vários métodos de análise comuns a ambos, ou seja, as pesquisas envolvendo documentos digitalizados por câmeras de vídeo digitais ou câmeras fotográficas digitais podem apresentar soluções semelhantes para problemas comuns a ambas. Câmeras digitais eficientes no processo de digitalização são muito caras, incluindo as aplicações que as acompanham. A

principal razão para utilização de câmeras de alta precisão é a manipulação de documentos que não podem ser digitalizados por *scanners*, tais como livros grossos e raros, manuscritos históricos raros e papéis quebradiços, então o ambiente é ajustado para proporcionar o melhor desempenho possível da câmera. Tipicamente um ambiente é preparado para tornar um documento parado em uma superfície, a mais plana possível, e possivelmente abaixo de uma camada de vidro. Algumas destas câmeras são acompanhadas por aplicativos inteligentes que podem inclusive restaurar linhas de texto distorcidas próximas ao eixo do livro. Recentemente algumas empresas têm repensado esta idéia para aplicações em ambientes de escritórios. Existem câmeras digitais utilizadas para substituir transparências de projetores, enquanto outras são utilizadas para substituir *scanners* por conveniente captura de imagens [26].

A disseminação de câmeras digitais portáteis por parcelas diversas da população mundial expandiu os horizontes da digitalização de documentos, substituindo o uso doméstico de potenciais aquisições de *scanners*. Esta expansão ocorre ainda mais rápido com a utilização de câmeras digitais em aparelhos celulares. A mais importante característica dessas câmeras é a sua flexibilidade: elas podem ser tão pequenas quanto um cartão bancário, à prova d'água, transportadas para qualquer lugar e de fácil utilização.

Porém, no caso de imagens de documentos essas condições de trabalho flexíveis apresentam necessidades de desenvolvimento de algoritmos capazes de lidar com o processamento necessário para melhoria dessas imagens, de forma a torná-las mais adequadas às novas aplicações. A variabilidade do ambiente de trabalho de câmeras digitais torna os desafios encontrados bem diferentes dos desafios encontrados em *scanners* ou das câmeras industriais. Atualmente as câmeras digitais destinadas ao uso de pessoas comuns atingem mais de 16 Mpixels, com resolução de 4992 x 3328 pixels (largura x altura), isto seria suficiente para capturar documentos em tamanho A4 (21 x 29,7 cm) a uma resolução superior a 300 dpi, adequada à análise de imagem de documentos [31]. Embora a maior parte dos dispositivos sejam ainda encontrados até 7 Mpixels, nos próximos anos é provável que a aquisição de câmeras de maiores resoluções pela população em geral seja cada vez mais comum. A Tabela 2.1 apresenta uma breve comparação entre *scanners* e câmeras digitais.

2.4.1 Desafios encontrados na digitalização

Atuais softwares de processamento de imagens produzem bons resultados para documentos sem ruídos, porém exigem algum nível de treinamento dos usuários e não possuem filtros espe-

Tabela 2.1: Comparação entre scanners *e câmeras digitais.*

	Scanners	**Câmeras Digitais**
Resolução	150–600 dpi	50–600 dpi
Superfície	Plana	arbitrária
Distorção	Mínima	Perspectiva e óptica
Iluminação	Adequada	Difícil de controlar
Plano de fundo	Conhecido	Complexo
Velocidade	Lenta	Rápida
Foco	Fixo	variável
Portabilidade	Ruim	Boa
Usabilidade	Baixa	Alta
Tamanho	Grande	Compacto

cíficos para a melhoria de documentos fotografados por câmeras fotográficas digitais portáteis. A maior parte dos desafios inclui:

a) Iluminação inadequada: câmeras digitais portáteis têm menos controle das condições de iluminação do que *scanners*. Iluminação inadequada é comum devido principalmente a dois fatores: o ambiente físico (sombras, reflexões, fluorescência) e respostas inadequadas dos dispositivos. Tais complicações ocorrem mesmo quando a iluminação ambiente é controlada. Os *flashes* das câmeras digitais portáteis foram desenvolvidos para iluminar o alvo de fotografia há mais de 1,5 metros de distância da câmera, enquanto a fotografia de documentos exige maior proximidade da câmera devido à busca por maior resolução; essa distância é próxima a 40 centímetros. Devido a esse fato é freqüente que fotografias de documentos exibam áreas com luminosidade irregular;

b) Distorção de perspectiva: ocorre devido à utilização livre das câmeras digitais portáteis sem o auxílio de um suporte mecânico, o que faz com que o plano do documento não seja paralelo ao plano da câmera. Como resultado, os caracteres mais distantes da base da imagem tornam-se menores e as linhas paralelas verticais não mais são preservadas. Mesmo pequenas distorções de perspectiva são suficientes para causar problemas significantes em softwares comerciais de OCR [26]. Para *scanners*, em geral documentos são perfeitamente alinhados com a superfície de vidro, não apresentando distorções de perspectiva, excetuando-se os casos em que não seja possível devido ao volume do documento, como por exemplo livros grossos;

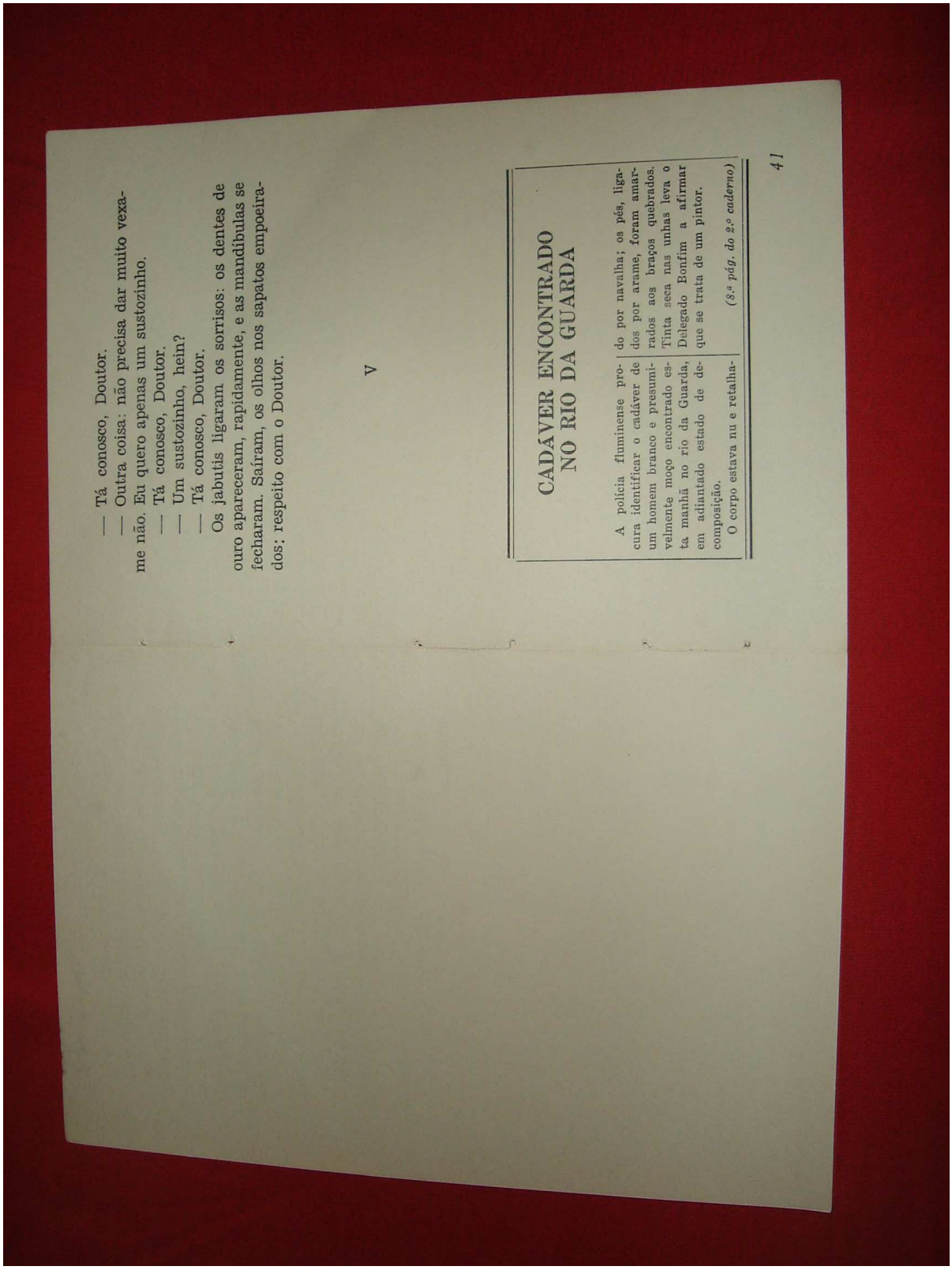
c) Distorções de lente (efeito esférico): as lentes produzem curvaturas do centro da imagem

para as bordas, fazendo com que à medida que o documento aproxime-se da lente, torne a distorção maior. Este tipo de distorção causa dificuldade na segmentação da imagem, devido ao fato de o documento não mais representar um quadrilátero;

d) Planos de fundo complexos: freqüentemente o suporte mecânico sobre o qual está o documento também é capturado na imagem com o documento fotografado. Se o documento não tiver contorno regular, isto o tornará difícil de segmentar (processo pelo qual uma imagem é subdividida em sua regiões ou objetos constituintes), mesmo se uma parte do plano de fundo apresentado na imagem for uniforme;

e) Zoom e foco: muitos dispositivos digitais são projetados para operar em uma grande variedade de distâncias, tornando o foco um fator significante. A pequenas distâncias, comuns para captação de imagens de documentos, mesmo pequenas mudanças de perspectiva podem causar focos inadequados. O método de captação de cores pela câmera também pode atenuar as freqüências altas, causando a perda de foco [9];

Na Figura 2.6, temos um documento (20,7 x 20 cm) fotografado com uma câmera digital da marca Sony, modelo DSC-S40, a 4.1 Mpixels (2348 x 1684 pixels) com *flash*, compactado no formato JPEG, apresentando distorções de perspectiva, curvatura e plano de fundo complexo. A Figura 2.7 apresenta o mesmo documento digitalizado através de *scanner* da marca Hewlett-Packard, modelo 5300c, em cores, a 100dpi, com 1169 x 850 pixels, em formato bitmap sem compressão, apresentando pequena inclinação, erro comum na digitalização e facilmente corrigível, enquanto livre dos problemas citados sobre a Figura 2.6.



— Tá conosco, Doutor.

— Outra coisa: não precisa dar muito vexame não. Eu quero apenas um sustozinho.

— Tá conosco, Doutor.

— Um sustozinho, hein?

— Tá conosco, Doutor.

Os jabutis ligaram os sorrisos: os dentes de ouro apareceram, rapidamente, e as mandíbulas se fecharam. Saíram, os olhos nos sapatos empoeirados; respeito com o Doutor.

V

CADÁVER ENCONTRADO NO RIO DA GUARDA

A polícia fluminense procura identificar o cadáver de um homem branco e presumivelmente moço encontrado esta manhã no rio da Guarda, em adiantado estado de decomposição.

O corpo estava nu e retalhado por navalha; os pés, ligados por arame, foram amarrados aos braços quebrados. Tinta seca nas unhas leva o Delegado Bonfim a afirmar que se trata de um pintor.

(8.ª pág. do 2.º caderno)

41

Figura 2.6: Documento digitalizado através de câmera fotográfica digital portátil Sony DSC-S40, 4.1 Mpixels, com flash.

— Tá conosco, Doutor.
— Outra coisa: não precisa dar muito vexame não. Eu quero apenas um sustozinho.
— Tá conosco, Doutor.
— Um sustozinho, hein?
— Tá conosco, Doutor.
Os jabutis ligaram os sorrisos: os dentes de ouro apareceram, rapidamente, e as mandíbulas se fecharam. Saíram, os olhos nos sapatos empoeirados; respeito com o Doutor.

V

CADÁVER ENCONTRADO NO RIO DA GUARDA

A polícia fluminense procura identificar o cadáver de um homem branco e presumìvelmente moço encontrado esta manhã no rio da Guarda, em adiantado estado de decomposição.
O corpo estava nu e retalhado por navalha; os pés, ligados por arame, foram amarrados aos braços quebrados. Tinta seca nas unhas leva o Delegado Bonfim a afirmar que se trata de um pintor.
(8.ª pág. do 2.º caderno)

41

Figura 2.7: Documento da Figura 2.6 digitalizado através de scanner *HP, modelo 5300c, a 100dpi em true color.*

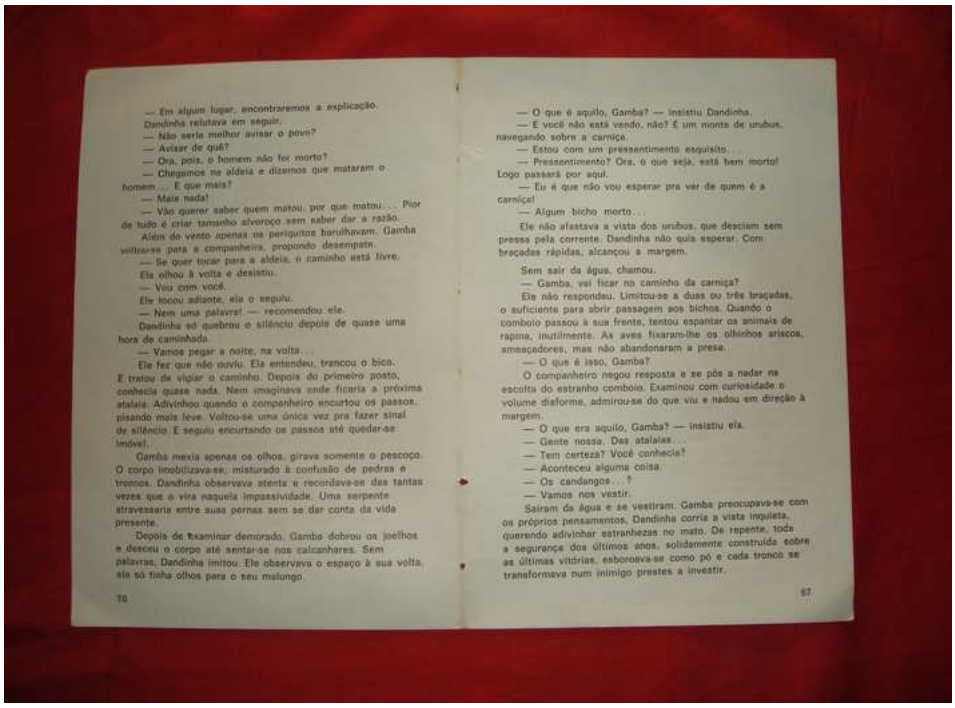
capítulo 3

Remoção de bordas

Bordas podem ser definidas como áreas inerentes ao ambiente no qual o documento foi digitalizado, que circundam a imagem do documento, não contendo informação relevante. O aparecimento de bordas não é inerente exclusivamente à digitalização de documentos utilizando câmeras digitais portáteis. O documento apresentado na Figura 2.7 possui borda clara, perceptível nas regiões inferior, superior e laterais direita e esquerda da imagem. Tais bordas correspondem à área da bandeja do *scanner* não ocupada pelo documento.

Bordas necessitam de espaço adicional para armazenamento das imagens, maior ocupação da largura de banda na transmissão de imagens via rede de comunicação de dados, diminuem a qualidade na visualização em monitores de computadores devido à inserção de informação desnecessária e dificultam sua segmentação, dentre outros problemas [12]. As Figuras 3.1 a 3.3 ilustram as dificuldades impostas pela borda na binarização do documento, através da aplicação do algoritmo da-Silva-Lins-Rocha [7] de binarização automática antes e após o recorte da borda. O problema apresentado no uso desse algoritmo também será obtido em todos algoritmos que tenham como base a análise estatística da imagem, pois a borda irá alterar a estatística da filtragem, resultando em perda de informação. No caso de documentos fotografados por câmeras digitais portáteis, o aparecimento de bordas é freqüente, pois à mão livre a imagem adquirida geralmente inclui parte do suporte mecânico sobre o qual encontra-se o documento.

A remoção de bordas deve ser executada automaticamente, pois tal remoção dependente de intervenção requer usuários com conhecimento sobre como removê-las, tornando o processo lento [45]. Os algoritmos para remoção de bordas devem conter o mínimo possível de



— Em algum lugar, encontraremos a explicação.
Dandinha relutava em seguir.
— Não seria melhor avisar o povo?
— Avisar de quê?
— Ora, pois, o homem não foi morto?
— Chegamos na aldeia e dizemos que mataram o homem... E que mais?
— Mais nada!
— Vão querer saber quem matou, por que matou... Pior de tudo é criar tamanho alvoroço sem saber dar a razão.
Além do vento apenas os periquitos barulhavam. Gamba voltou-se para a companheira, propondo desempate.
— Se quer tocar para a aldeia, o caminho está livre.
Ela olhou à volta e desistiu.
— Vou com você.
Ele tocou adiante, ela o seguiu.
— Nem uma palavra! — recomendou ele.
Dandinha só quebrou o silêncio depois de quase uma hora de caminhada.
— Vamos pegar a noite, na volta...
Ele fez que não ouviu. Ela entendeu, trancou o bico. E tratou de vigiar o caminho. Depois do primeiro posto, conhecia quase nada. Nem imaginava onde ficaria a próxima atalaia. Adivinhou quando o companheiro encurtou os passos, pisando mais leve. Voltou-se uma única vez pra fazer sinal de silêncio. E seguiu encurtando os passos até quedar-se imóvel.
Gamba mexia apenas os olhos, girava somente o pescoço. O corpo imobilizava-se, misturado à confusão de pedras e troncos. Dandinha observava atenta e recordava-se das tantas vezes que o vira naquela impassividade. Uma serpente atravessaria entre suas pernas sem se dar conta da vida presente.
Depois de examinar demorado, Gamba dobrou os joelhos e desceu o corpo até sentar-se nos calcanhares. Sem palavras, Dandinha imitou. Ele observava o espaço à sua volta, ela só tinha olhos para o seu malungo.

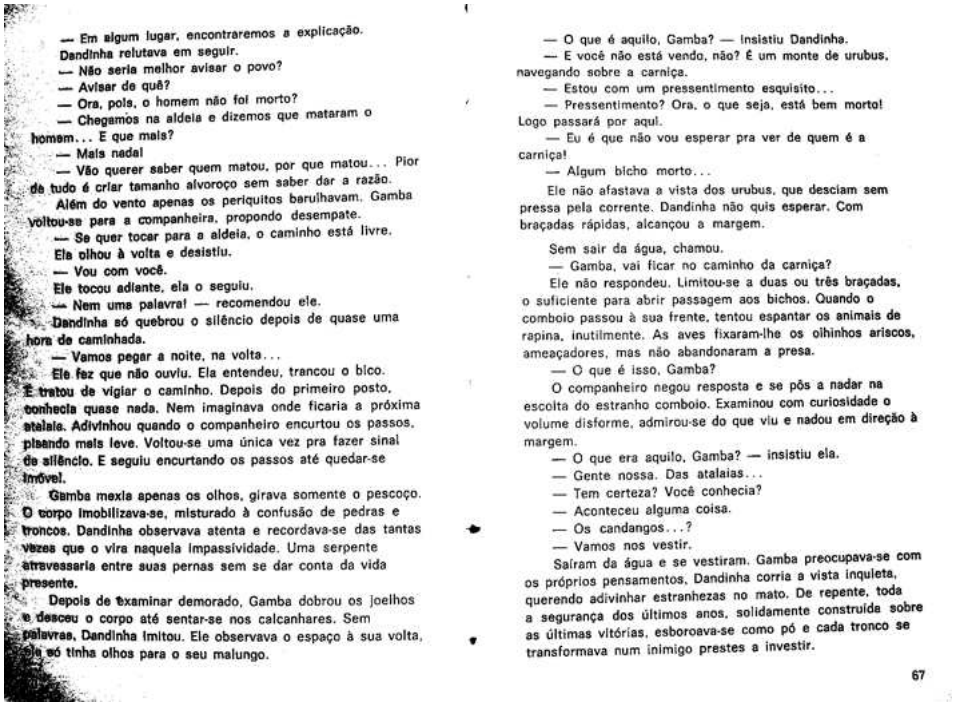
70

— O que é aquilo, Gamba? — insistiu Dandinha.
— E você não está vendo, não? É um monte de urubus, navegando sobre a carniça.
— Estou com um pressentimento esquisito...
— Pressentimento? Ora, o que seja, está bem morto! Logo passará por aqui.
— Eu é que não vou esperar pra ver de quem é a carniça!
— Algum bicho morto...
Ele não afastava a vista dos urubus, que desciam sem pressa pela corrente. Dandinha não quis esperar. Com braçadas rápidas, alcançou a margem.
Sem sair da água, chamou.
— Gamba, vai ficar no caminho da carniça?
Ele não respondeu. Limitou-se a duas ou três braçadas, o suficiente para abrir passagem aos bichos. Quando o comboio passou à sua frente, tentou espantar os animais de rapina, inutilmente. As aves fixaram-lhe os olhinhos ariscos, ameaçadores, mas não abandonaram a presa.
— O que é isso, Gamba?
O companheiro negou resposta e se pôs a nadar na escolta do estranho comboio. Examinou com curiosidade o volume disforme, admirou-se do que viu e nadou em direção à margem.
— O que era aquilo, Gamba? — insistiu ela.
— Gente nossa. Das atalaias...
— Tem certeza? Você conhecia?
— Aconteceu alguma coisa.
— Os candangos...!
— Vamos nos vestir.
Saíram da água e se vestiram. Gamba preocupava-se com os próprios pensamentos, Dandinha corria a vista inquieta, querendo adivinhar estranhezas no mato. De repente, toda a segurança dos últimos anos, solidamente construída sobre as últimas vitórias, esborosva-se como pó e cada tronco se transformava num inimigo prestes a investir.

67

Figura 3.1: Documento fotografado original.

Figura 3.2: Binarização automática sem recorte da borda.



— Em algum lugar, encontraremos a explicação.
Dandinha relutava em seguir.
— Não seria melhor avisar o povo?
— Avisar de quê?
— Ora, pois, o homem não foi morto?
— Chegamos na aldeia e dizemos que mataram o homem... E que mais?
— Mais nada!
— Vão querer saber quem matou, por que matou... Pior de tudo é criar tamanho alvoroço sem saber dar a razão.
Além do vento apenas os periquitos barulhavam. Gamba voltou-se para a companheira, propondo desempate.
— Se quer tocar para a aldeia, o caminho está livre.
Ela olhou à volta e desistiu.
— Vou com você.
Ele tocou adiante, ela o seguiu.
— Nem uma palavra! — recomendou ele.
Dandinha só quebrou o silêncio depois de quase uma hora de caminhada.
— Vamos pegar a noite, na volta...
Ele fez que não ouviu. Ela entendeu, trancou o bico. E tratou de vigiar o caminho. Depois do primeiro posto, conhecia quase nada. Nem imaginava onde ficaria a próxima atalaia. Adivinhou quando o companheiro encurtou os passos, pisando mais leve. Voltou-se uma única vez pra fazer sinal de silêncio. E seguiu encurtando os passos até quedar-se imóvel.
Gamba mexia apenas os olhos, girava somente o pescoço. O corpo imobilizava-se, misturado à confusão de pedras e troncos. Dandinha observava atenta e recordava-se das tantas vezes que o vira naquela impassividade. Uma serpente atravessaria entre suas pernas sem se dar conta da vida presente.
Depois de examinar demorado, Gamba dobrou os joelhos e desceu o corpo até sentar-se nos calcanhares. Sem palavras, Dandinha imitou. Ele observava o espaço à sua volta, e só tinha olhos para o seu malungo.

— O que é aquilo, Gamba? — insistiu Dandinha.
— E você não está vendo, não? É um monte de urubus, navegando sobre a carniça.
— Estou com um pressentimento esquisito...
— Pressentimento? Ora, o que seja, está bem morto! Logo passará por aqui.
— Eu é que não vou esperar pra ver de quem é a carniça!
— Algum bicho morto...
Ele não afastava a vista dos urubus, que desciam sem pressa pela corrente. Dandinha não quis esperar. Com braçadas rápidas, alcançou a margem.
Sem sair da água, chamou.
— Gamba, vai ficar no caminho da carniça?
Ele não respondeu. Limitou-se a duas ou três braçadas, o suficiente para abrir passagem aos bichos. Quando o comboio passou à sua frente, tentou espantar os animais de rapina, inutilmente. As aves fixaram-lhe os olhinhos ariscos, ameaçadores, mas não abandonaram a presa.
— O que é isso, Gamba?
O companheiro negou resposta e se pôs a nadar na escolta do estranho comboio. Examinou com curiosidade o volume disforme, admirou-se do que viu e nadou em direção à margem.
— O que era aquilo, Gamba? — insistiu ela.
— Gente nossa. Das atalaias...
— Tem certeza? Você conhecia?
— Aconteceu alguma coisa.
— Os candangos...?
— Vamos nos vestir.
Saíram da água e se vestiram. Gamba preocupava-se com os próprios pensamentos, Dandinha corria a vista inquieta, querendo adivinhar estranhezas no mato. De repente, toda a segurança dos últimos anos, solidamente construída sobre as últimas vitórias, esboroava-se como pó e cada tronco se transformava num inimigo prestes a investir.

67

Figura 3.3: Binarização automática após recorte da borda.

restrições, pois os usuários tendem a fotografar documentos nos mais variados ambientes, culminando nos mais variados efeitos indesejáveis. Esse problema encontra-se bem estudado em imagens de documentos digitalizados por *scanners* [45] [12], servindo como referência para o desenvolvimento de algoritmos em câmeras digitais.

3.1 Características de documentos fotografados

As características de imagens adquiridas através de fotografias encontram-se bem estudadas na literatura [13] [17], tais como a transformação de planos realizada pela câmera, distorções, rotações e ruídos nos sensores de captação de imagens. Os desafios que causam dificuldades na criação de algoritmos para processamento de documentos fotografados estão diretamente relacionados à necessidade de intervenção humana para fornecer parâmetros que possibilitem a execução adequada. Por exemplo, para a remoção de bordas seria necessário marcar a área a ser removida para cada imagem e aplicar-se um algoritmo de recorte da imagem, tornando o processo dispendioso e impreciso.

Portanto, faz parte deste estudo analisar as características encontradas em documentos fotografados e utilizá-las para a remoção de bordas, requerendo o mínimo de intervenção do

usuário, fazendo-se necessário adquirir um banco de imagens que possibilite realizar comparações entre estas e buscar estabelecer semelhanças e diferenças.

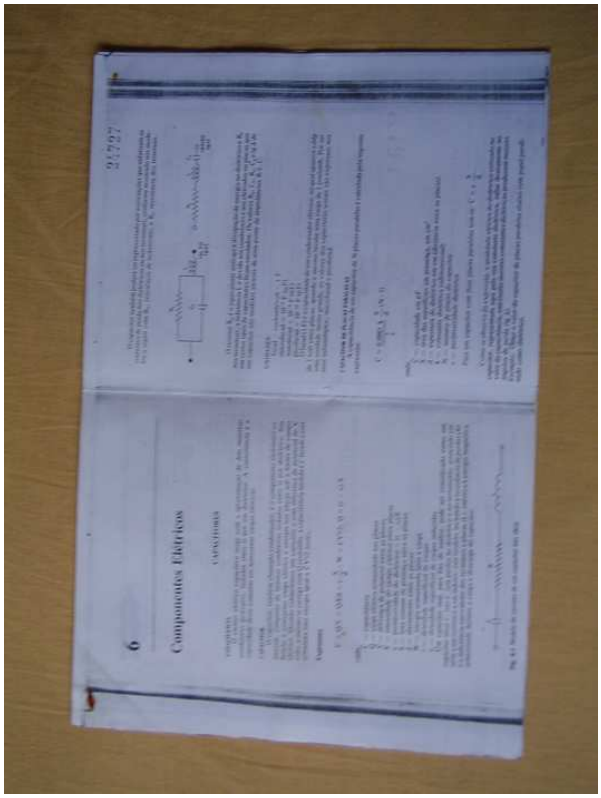
Para a pesquisa descrita nesta dissertação foi criado um banco de imagens contendo 400 fotografias, com resolução de 3.2 Mpixels (2048 x 1536 pixels). Os documentos utilizados para compor o banco de imagem foram fotografados com 16 milhões de cores (formato conhecido como *true color*), contendo páginas de livros, listas telefônicas, folhetos, apostilas e certificados, fotografados sem a ajuda de suporte para a câmera. Dentre os documentos utilizados havia papéis de diferentes cores, figuras e tabelas. Os papéis constituintes de tais documentos eram translúcidos, não apresentando interferência frente-verso [30] [32]. Os ambientes para digitalização destes documentos também foram variados, conforme se pode observar na Figura 3.4. A Figura 3.5 apresenta os histogramas RGB das imagens contidas na Figura 3.4. Através dos histogramas podemos observar a grande variação na composição das imagens, e que a grande variação de cores na borda e nas figuras que possam vir a fazer parte das imagens tornam difícil a análise das bordas através dos histogramas.

A análise das características inerentes a este banco de imagens foi realizada através da observação das componentes R, G e B (referentes ao padrão RGB), atentando para suas variações em pixels consecutivos, ao longo dos documentos e principalmente nos contornos destes. Verificou-se também as características típicas de figuras, fontes, papéis e planos de fundo, assim como as variações de brilho de acordo com as adversidades dos ambientes.

As principais características observadas e identificadas como influentes no desenvolvimento de algoritmos de remoção de bordas encontram-se descritas nas subseções que seguem.

3.1.1 O efeito da iluminação sobre os documentos

Câmeras digitais são projetadas para trabalhar com imagens complexas, de ambientes tridimensionais, e, em sua maioria, com grande quantidade de cores. A distância dessas câmeras para o objeto a ser fotografado costuma ser superior a um metro de distância, como é o caso de fotos de famílias, de paisagens, etc. A essa distância o efeito do *flash* sobre o objeto apresenta-se disperso, podendo ser considerado desprezível, ou tornar a fotografia visível, em caso de ambientes escuros. Ao se utilizar a câmera digital para aquisição de documentos, usuários costumam deixar pouca sobra do ambiente no qual o documento é fotografado, assim sendo, a distância entre a câmera e o documento costuma ser de 40cm a 60cm. A essa distância, o *flash* causa iluminação irregular no documento, ou seja, causa maior



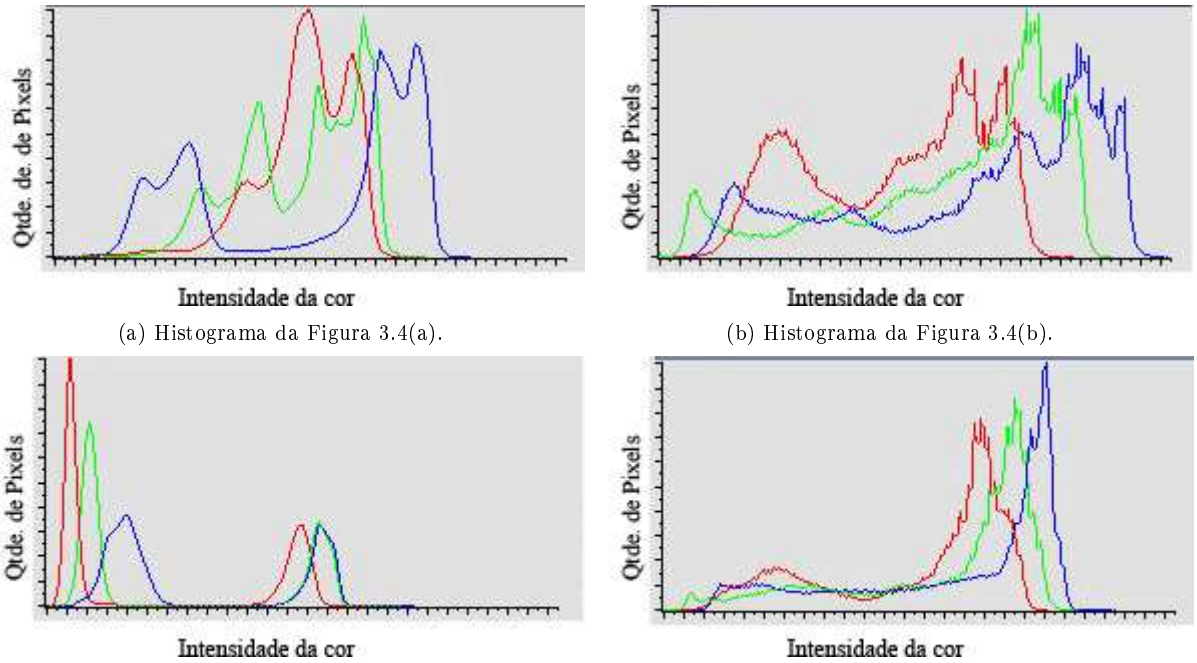
(a) Tamanho 246,09KB JPEG.

(b) Tamanho 375,6KB JPEG.

(c) Tamanho 140,5KB JPEG.

(d) Tamanho 338,9KB JPEG.

Figura 3.4: Exemplos de documentos com variação do plano de fundo e tipo do documento.



(a) Histograma da Figura 3.4(a).

(b) Histograma da Figura 3.4(b).

(c) Histograma da Figura 3.4(c).

(d) Histograma da Figura 3.4(d).

Figura 3.5: Histogramas das imagens contidas na Figura 3.4.

brilho em determinada região do documento. A distribuição irregular do brilho na imagem limita a aplicação dos algoritmos tradicionais de segmentação.

Pode-se constatar que em fotos com *flash* o brilho próximo ao centro da imagem tende a ser maior, coincidindo com a posição do *flash* na câmera e independente de iluminação lateral natural ou de ambiente, diminuindo gradualmente ao afastar-se desse ponto, conforme pode ser observado na Figura 3.6. Em imagens sem *flash* o ambiente influencia muito na composição da imagem, fazendo com que a região com maior brilho varie de uma imagem para outra.

A distribuição não uniforme de brilho sobre a imagem pode fazer com que a cor do papel nos vértices do documento aproxime-se da cor da fonte na área central do documento, constituindo um problema para binarização destas imagens. Tal fato pode ser observado na Figura 3.7, onde a cor destacada na parte superior foi retirada do papel (R = 111, G = 138, B = 147) e a cor destacada na parte inferior foi retirada da fonte do documento (R = 114, G = 140, B = 141). Além disso, a cor da borda pode aproximar-se muito da cor do papel dificultando a identificação automática dos limites do documento, conforme ilustrado na Figura 3.8. É possível observar que nesses casos o contorno do documento não é bem definido.

Figura 3.6: Região aproximada de maior atuação do flash.

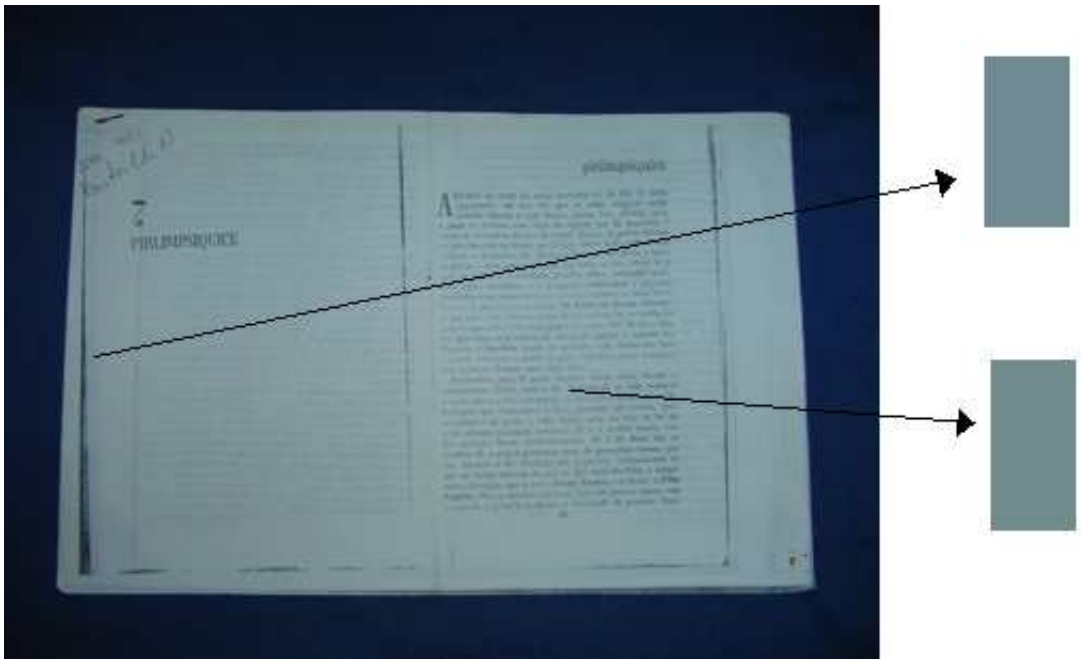


Figura 3.7: Cores de papel e fonte com valores próximos.

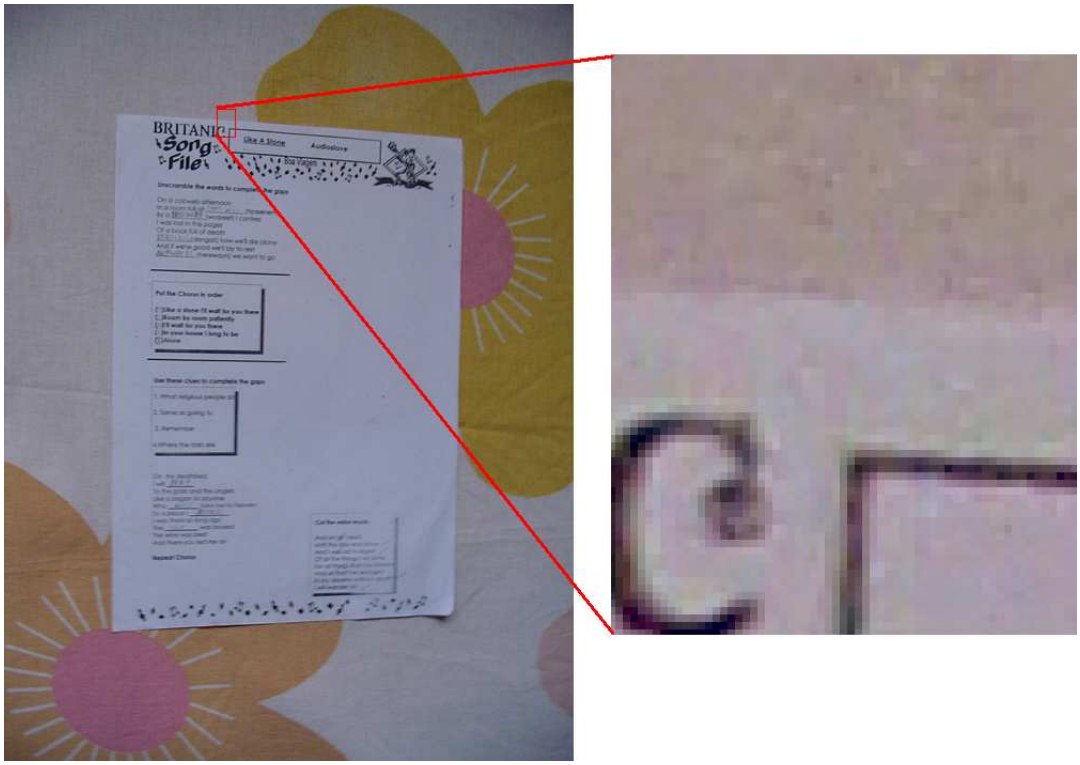


Figura 3.8: Cores de papel e borda com valores próximos.

3.1.2 Os contornos em documentos fotografados

Contornos são pixels onde o brilho na imagem muda abruptamente e denotam locais na imagem que correspondem aos limites de um objeto. No caso de documentos há contornos em volta do papel, das fontes, das tabelas e fora das figuras que possam compô-lo, podendo também haver contornos em áreas internas. Para a remoção de bordas é comum a utilização da identificação do contorno do papel para a definição da área da imagem a ser recortada.

Pesquisas na detecção de contornos têm focado primariamente na intensidade das imagens em escala de cinza [25]. Observou-se que a aplicação de algoritmos de remoção de bordas apenas em escala de cinza limita a atuação destes, posto que é comum a ocorrência de cores visualmente diferentes possuindo valores próximos quando realizada a conversão para escala de cinza.

A captação adequada de cores por *scanners* oferece maior intensidade dos contornos, enquanto nas imagens adquiridas por câmeras digitais a intensidade destes demonstrou ser muito dependente do foco, além de sofrer o efeito de interpolação descrito no Capítulo 2. Desta forma, torna-se necessário o estabelecimento de algumas limitações para definição de

um algoritmo de remoção automática de borda, referentes à diferença de cores entre o papel e a borda.

3.1.3 Conteúdo de documentos

Documentos em geral podem possuir mais de um tipo de conteúdo, como tabelas, desenhos, fotografias, gráficos, entre outros. Desta forma, os algoritmos para remoção automática de bordas devem identificar o conteúdo do documento e diferenciá-lo da borda da imagem, para assim realizar a remoção da borda sem exclusão de informação dos documentos.

Pode-se observar que nas imagens resultantes da digitalização de documentos burocráticos o conteúdo dominante é o papel do documento, e que, excetuando-se os casos em que há imagens dentro do documento, a informação (fontes e tabelas) ocupa cerca de 5% a 6% da área da imagem referente ao documento (dados obtidos experimentalmente). Tal informação contribui no desenvolvimento de algoritmos para tratamento de documentos fotografados, pois restringindo-se a localização do documento pode-se estimar a cor do papel, mais freqüente do que a cor da informação.

3.2 Descrição de um novo algoritmo

Em virtude da grande variação na composição das bordas, optou-se pela identificação dos padrões do documento, para a partir de então proceder-se com a identificação dos limites do papel. Observou-se que pixels consecutivos pertencentes ao papel apresentam pequenas variações nas três componentes RGB, sendo estas variações crescentes ou decrescentes. De acordo com as observações, foi identificado que para o banco de imagens adquirido, os pixels vizinhos pertencentes ao papel apresentavam variação de até 16 níveis em duas componentes, enquanto a terceira componente poderia variar em até o dobro da variação dos outros dois níveis.

Devido às distorções geométricas e de perspectiva presentes na digitalização de documentos por câmeras digitais, a remoção de bordas não pode ser totalmente eficiente, caso contrário poderá haver corte da informação, conforme pode ser observado nas Figuras 3.9, 3.10 e 3.11.

As etapas a seguir descrevem um algoritmo proposto, que busca remover o máximo possível das bordas de forma a preservar a área referente à imagem do documento propriamente dita.

Etapa 1 Cálculo da moda no centro da imagem

Levando em consideração o fato de a informação ocupar espaço muito menor do que o

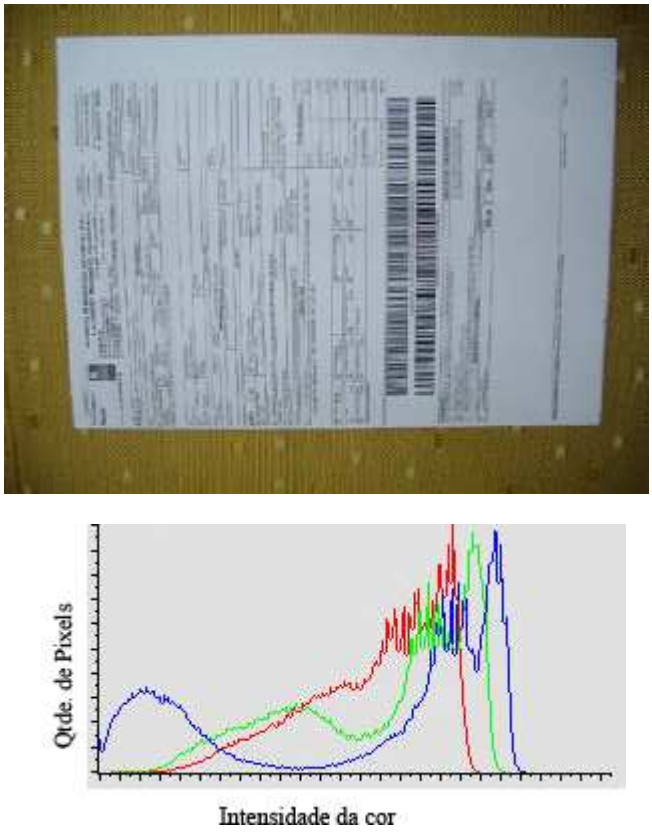


Figura 3.9: Documento fotografado original (3.2 Mpixels com flash*).*

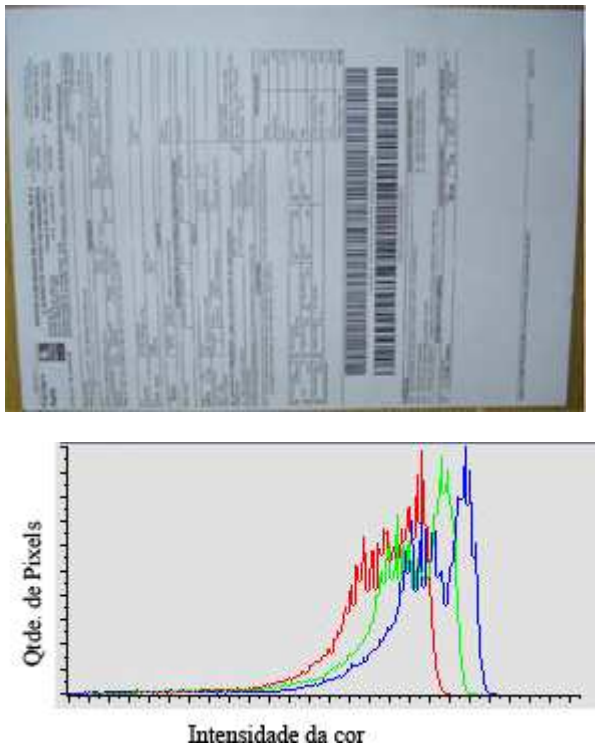


Figura 3.10: Recorte de borda com bordas remanescentes.

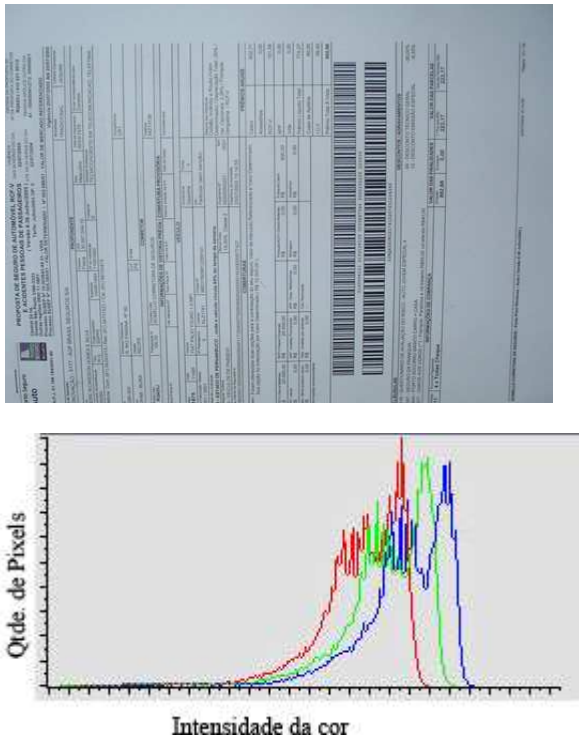


Figura 3.11: Recorte de borda com perda da informação.

papel do documento, calculou-se o pixel que mais se repete na região central da imagem, correspondente a 1/9 do total da imagem. A decisão pela moda nesta região foi definida com base no brilho causado pelo *flash* ser mais intenso, e pela região mais provável de o documento estar localizado na imagem.

Observando-se as variações identificadas entre pixels vizinhos pertencentes ao papel, este pixel serviu como referência para a busca pelos limites do papel. Assumiu-se que o documento sempre preenche a maior parte da região central da imagem (Figura 3.12).

Etapa 2 Estimativa dos limites do documento

Baseando-se na moda da região central da imagem, e utilizando-se o conhecimento das variações entre pixels vizinhos, buscou-se encontrar os limites do papel do documento, partindo do centro da imagem e em direção à área externa ao documento até que atinja os limites da imagem. Durante a varredura o pixel mais recentemente classificado como papel, α, e a moda do documento, β, são utilizados como referências para determinar se o próximo pixel deve ser classificado como papel ou não-papel.

Observa-se que no padrão *true color* cada componente possui 8 bits cujos valores podem variar de 0 a 255. Duas tolerâncias são estabelecidas, a primeira determina um limite no

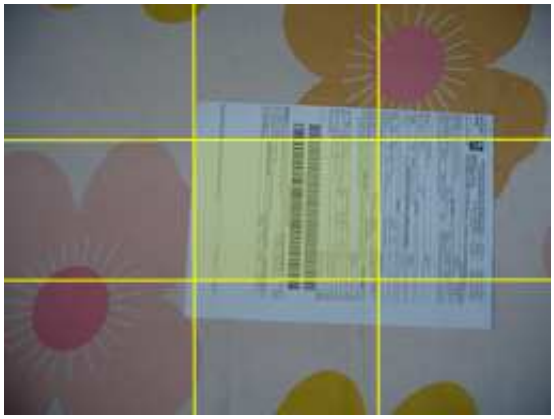


Figura 3.12: Remoção de bordas - etapa 1.

Figura 3.13: Remoção de bordas - etapa 2.

qual apenas uma das componentes do pixel atual pode ultrapassar quando comparada com a mesma componente em α e β. Esta primeira tolerância assume o valor 32 para o banco de imagens adquirido. A segunda tolerância define um valor no qual as duas outras componentes podem variar, possuindo o valor 16. Caso mais de uma componente exceda esse limite, o pixel é classificado como informação ou borda.

Estas tolerâncias determinam a flexibilidade do algoritmo, podendo ser ajustadas para diferentes conjuntos de imagens na obtenção de melhores resultados. Desta forma, quatro pontos são identificados como limites dos documentos (a_0=$\{x_0$,$y_0\}$, a_1=$\{x_1$,$y_1\}$, a_2=$\{x_2$,$y_2\}$ e a_3=$\{x_3$,$y_3\}$) (Figura 3.13).

Etapa 3 Cálculos para melhoria da precisão

Devido à variação de luz, que é freqüente ao utilizar-se o *flash* da câmera, a etapa 2 deste algoritmo pode não ser precisa. Ao aproximar-se do limite do documento, de dentro para fora, percebe-se que o brilho pode apresentar variações mais abruptas (superiores a 20%). Baseando-se na estimativa dos quatro pontos identificados como limites dos documentos, calcula-se a moda na região próxima a cada um destes pontos, em um limite estabelecido por 15 pixels em direção à parte exterior do documento e 60 pixels em direção à parte interior do documento, partindo do ponto estimado, e cujos outros dois limites são dados pelos dois pontos encontrados em varredura ortogonal a este ponto. Por exemplo, a moda que contém o ponto a_0, tem como limites os pontos y_1, y_3, $x_0 - 15$, $x_0 + 60$, ou seja, b_0=$\{x_0 - 15$,$y_3\}$, b_1=$\{x_0 - 15$,$y_1\}$, b_2=$\{x_0 + 60$,$y_1\}$, b_3=$\{x_0 + 60$,$y_3\}$ (Figura 3.14).

O valor de 15 pixels em direção à parte exterior do documento foi definido buscando-se assimilar valores do papel com menor iluminação, enquanto o valor de 60 pixels em di-

reção ao centro do documento garante que a moda será pertencente ao papel, e não à borda, visto que esta pode apresentar composição de cores simples. Tais valores foram determinados baseando-se em valor percentual à quantidade de pixels das imagens adquiridas e apresentaram bons resultados para a resolução de 3.2 Mpixels, sendo necessário rever esses valores para diferentes resoluções.

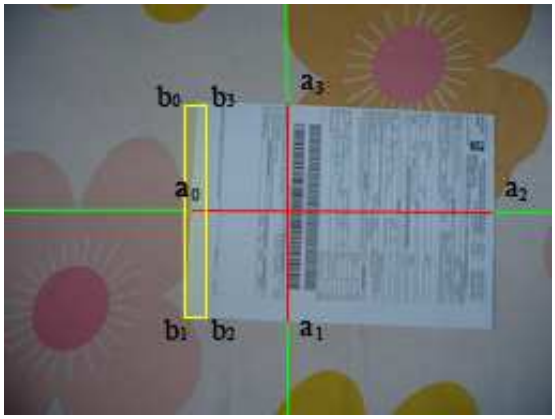


Figura 3.14: Remoção de bordas - etapa 3.

Etapa 4 Encontrando os valores para corte da imagem

Com os valores das modas das regiões próximas aos pontos estimados como limites do documento, executa-se varredura partindo de um dos limites inferiores do documento - altura ou largura - dependendo da região do documento, em direção aos limites superiores, classificando os pixels como borda ou papel. A cada pixel classificado como papel, efetua-se o deslocamento de um pixel em direção à região externa e retrocedem-se cinco pixels em direção ao limite inferior atuante. Com este retrocesso busca-se evitar irregularidades nos limites dos documentos, como buracos ou papel irregularmente recortado. Esta verificação é realizada dois pixels por vez para aumentar a confiabilidade, onde através das observações chegou-se à definição da tolerância de 32 níveis para todos canais. O último pixel classificado como papel indica o limite do documento, vertical ou horizontal, tal que s_k=$\{x_k,y_k\}$. Isto é, para verificar-se o limite esquerdo, parte-se da coordenada (x_0,0) em direção a (x_k,y_{max}). O ponto de verificação deslocar-se-á verticalmente. Caso os pontos $\{x_0,y_n\}$ e $\{x_0+1,y_n\}$ sejam identificados como papel, então os próximos pixels a serem verificados seriam $\{x_0\text{-}1,y_n\text{-}5\}$ e $\{x_0,y_n\text{-}5\}$ (Figura 3.15). Este algoritmo é conservativo, ou seja, após estimados os quatro limites do documento, verificam-se as menores e maiores alturas e larguras, para então ser realizado o corte da

imagem. O resultado pode ser observado na Figura 3.16.

Figura 3.15: Remoção de bordas - etapa 4.

Figura 3.16: Remoção de bordas - resultado.

3.3 Resultados e limitações

Testou-se o algoritmo apresentado em 400 imagens, adquirindo-se tempo médio de processamento de 1.2 segundos por imagem, utilizando-se microcomputador com processador Intel Pentium IV, 2.4GHz e 512MB de memória RAM. Devido a não terem sido encontrados algoritmos de remoção de bordas aplicáveis a documentos fotografados ou a imagens coloridas anteriores a este, não foi possível realizar comparações de eficiência.

Das 400 imagens filtradas por este algoritmo apenas duas apresentaram resultados insatisfatórios. A primeira apresentando grande quantidade de borda remanescente (Figura 3.17) e a segunda apresentando corte do documento (Figura 3.18). Entretanto, os documentos utilizados nos dois casos eram constituídos de papel de baixa qualidade e a superfície apresentava reflexo do *flash* tornando alguns pontos da borda de cor semelhante à cor da moda na região central do documento ou da moda na região próxima à margem inferior do documento.

O formato de arquivo utilizado por câmeras fotográficas digitais portáteis é do tipo JPEG. O algoritmo de compressão JPEG permite manter maior qualidade da imagem através de uma variável conhecida como fator de qualidade (Q), que assume valores de 1 a 100, um maior fator de qualidade resulta em menor compressão. Como o objetivo das câmeras digitais é fornecer imagens com a maior quantidade de detalhes possível, o fator de qualidade nas câmeras digitais portáteis pode muitas vezes ser ajustado. Para câmeras fotográficas digitais portáteis não embutidas em outros dispositivos, o fator de qualidade máximo assume valores acima de 90 [23] [15].

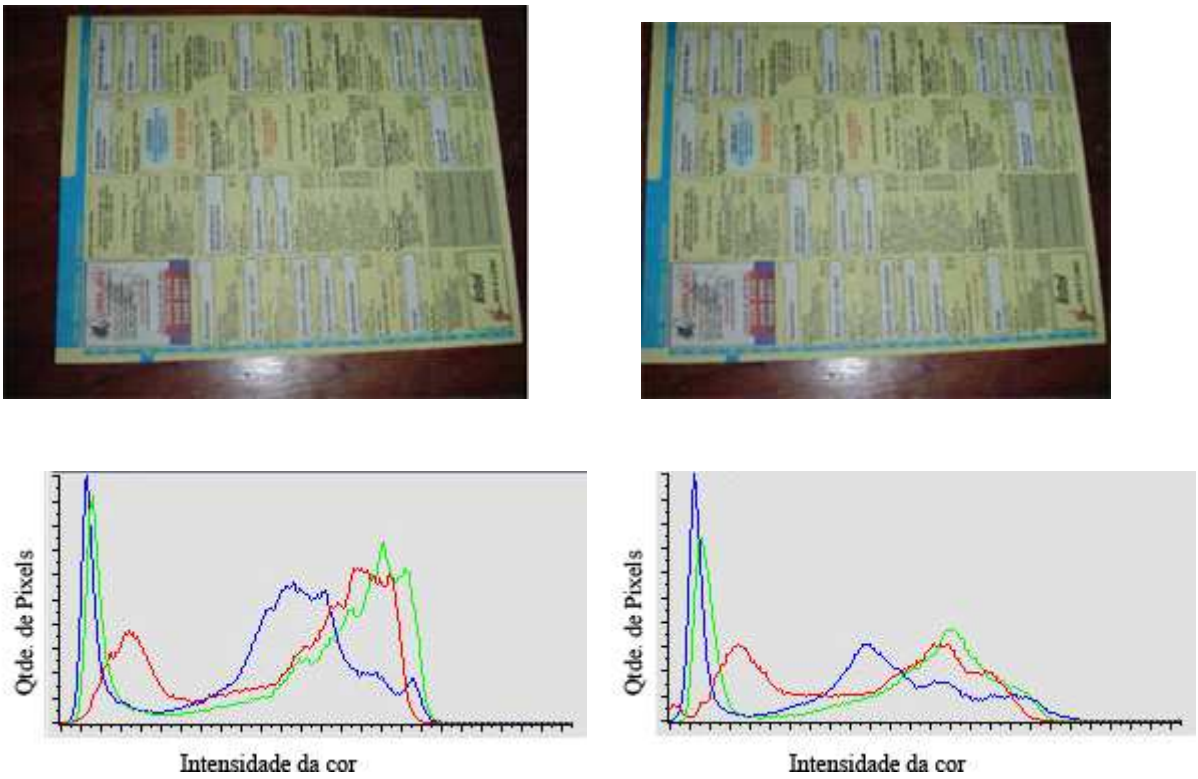


Figura 3.17: Erro por sobra de bordas.

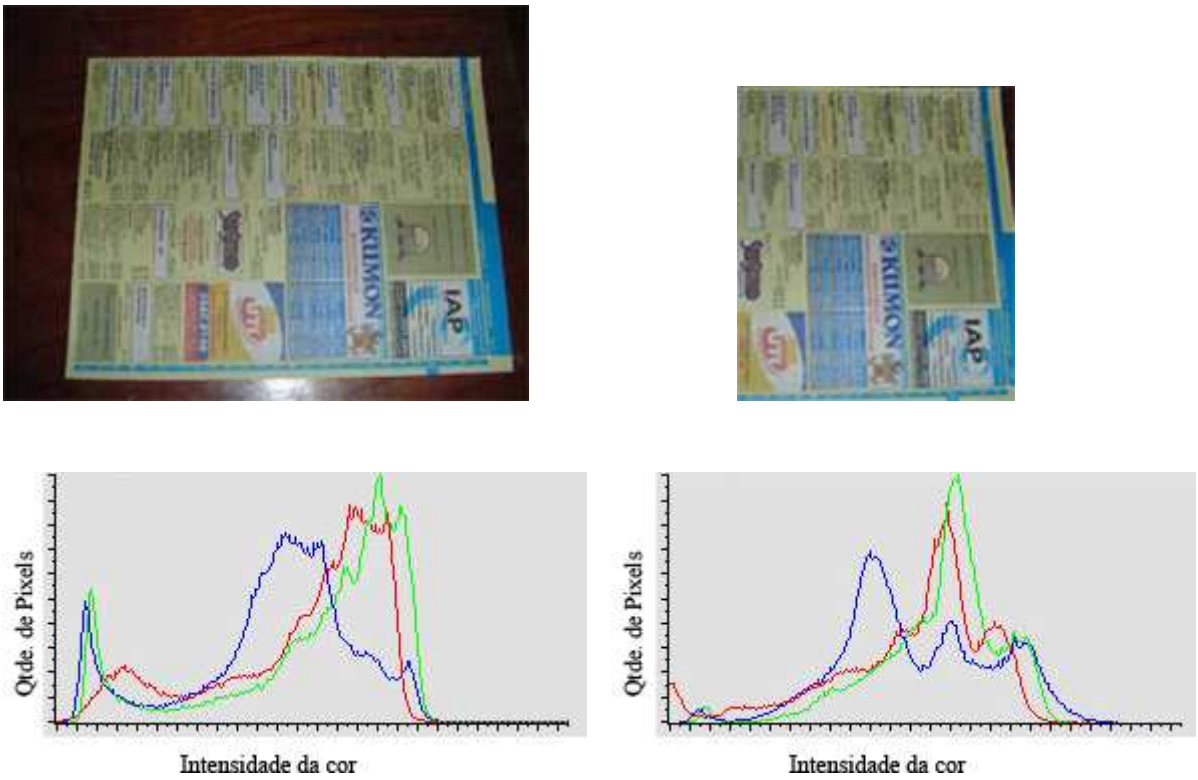
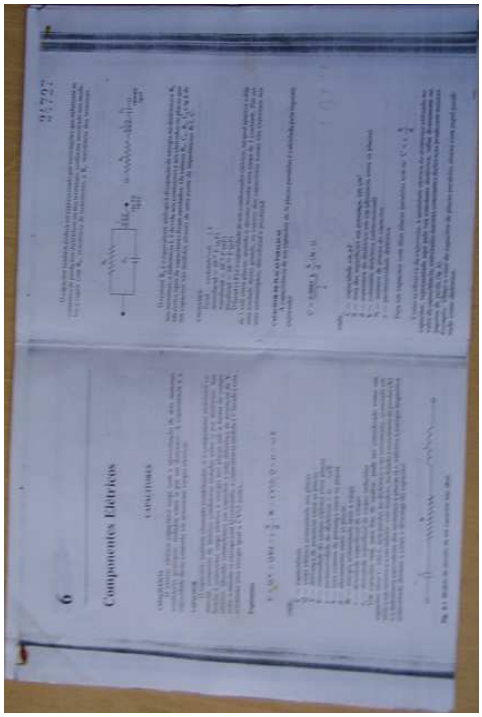


Figura 3.18: Erro por corte do documento.

No ambiente de documentos digitalizados é permitido o uso de um fator de compressão maior, como é o caso dos aplicativos de digitalização de documentos por *scanners*. Portanto, para a análise da redução no tamanho do arquivo, as imagens, antes e após a remoção das bordas, foram comprimidas em formato JPEG padrão ($Q = 50$). Comparando-se as imagens com borda e sem borda chegou-se à média de 40% de redução no tamanho das imagens.

O algoritmo é limitado a atuar apenas em documentos digitalizados em ambientes de cores distintas do seu papel. Desta forma, é possível ajustar as tolerâncias para a remoção da borda, permitindo a retirada da borda mesmo em imagens cujo ambiente de digitalização proporcionou grande variação de brilho, desde que a cor do papel nas regiões de menor brilho não tornem-se tão semelhantes às cores da borda quanto das demais cores do papel do documento na região de maior brilho. Superfícies que refletem a luz do *flash* podem causar a falha do algoritmo, pois algumas regiões destas superfícies aparecem brancas na imagem, podendo vir a coincidir ou aproximar-se muito da cor do papel do documento. Os documentos também devem possuir margens de, no mínimo, dois pixels, devido ao processo descrito na etapa 4 [11]. A Figura 3.19 mostra os resultados após a aplicação do algoritmo às imagens constantes na Figura 3.4. Os histogramas da Figura 3.19 são apresentados na Figura 3.20, onde comparando-se essa figura com a Figura 3.5 observa-se o impacto da remoção da borda pela redução da quantidade e da intensidade das cores. A maioria das técnicas de filtragem atuam sobre arquivos não compactados, também conhecidos como arquivos de mapa de bits ou de rastreamento, onde cada componente de cor de pixel corresponde a um ponto numa matriz. No sistema operacional Windows o formato de mapa de bits mais difundido é o BMP, aqui utilizado. A conversão dos arquivos JPEG para os equivalentes BMP foi efetivada utilizando-se o software Picaview32, versão 1.3 [49]. Os códigos-fonte dos algoritmos utilizados nesta pesquisa se encontram no apêndice A desta dissertação. Cópias digitais dos códigos-fonte e imagens de teste são encontrados no apêndice B.



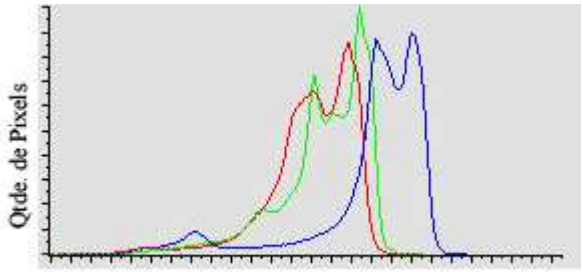
(a) Tamanho 175,69KB JPEG.

(b) Tamanho 282,9KB JPEG.

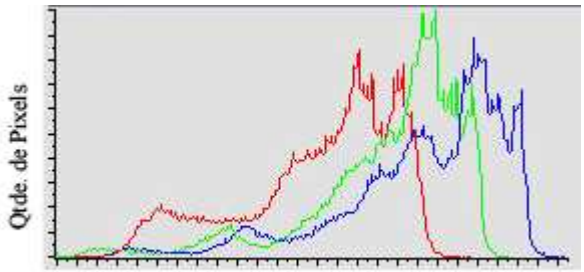
(c) Tamanho 78,9KB JPEG.

(d) Tamanho 263,3KB JPEG.

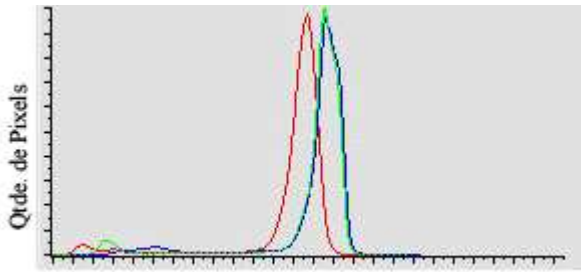
Figura 3.19: Resultado dos exemplos mostrados na Figura 3.4.



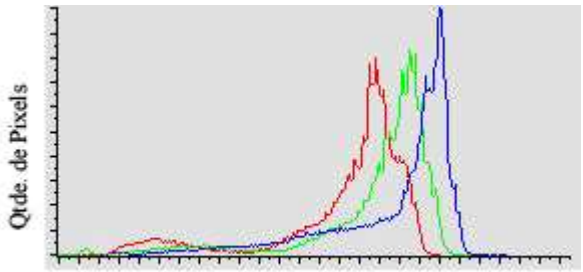
(a) Histograma da Figura 3.19(a).



(b) Histograma da Figura 3.19(b).



(c) Histograma da Figura 3.19(c).



(d) Histograma da Figura 3.19(d).

Figura 3.20: Histograma das imagens contidas na Figura 3.19.

capítulo 4

Uma análise da qualidade da transcrição de documentos fotografados

O termo qualidade em imagens é muito abrangente. Pode-se encontrar tanto pesquisas realizadas com base na percepção humana para identificar os fatores que influenciam na qualidade de imagens, quanto pesquisas baseadas em características das imagens em relação a padrões pré-estabelecidos. Mesmo restringindo o grupo de imagens a documentos digitalizados (datilografados ou tipografados), ainda não se pode determinar um método que realize esta medição com eficácia. Como a transcrição de documentos é essencial no âmbito de processamento de documentos, ferramentas de OCR podem servir para medição de qualidade, não como índice, mas como fator de comparação: quando o mesmo documento é digitalizado por *scanner* e câmera fotográfica digital, caso suas transcrições possuam semelhante quantidade de erros, estima-se que estes documentos digitalizados possuem qualidade semelhante. Convém observar que este método de análise limita-se a documentos datilografados ou impressos, pois documentos manuscritos produzem alta taxa de erro na transcrição por ferramentas de OCR.

4.1 Metodologia de medição de qualidade

No âmbito de documentos fotografados, cinco problemas têm sido citados como principais fatores prejudiciais à análise desses documentos: iluminação irregular, presença de bordas, inclinações, distorções geométricas e distorções de perspectiva. Para medir-se o grau de influência desses cinco problemas, faz-se necessário estabelecer critérios para estimar-se a qualidade destas imagens. Conforme cita-se previamente, no capítulo 2 desta dissertação, imagens adquiridas por *scanners* podem possuir bordas ou inclinações, entretanto estes problemas são facilmente eliminados, portanto, podem ser utilizadas como conjunto de referência para medição da qualidade, assumindo-se que as imagens adquiridas pelo *scanner* apresentam boa qualidade.

Devido à extrema complexidade na atribuição de índices de qualidade, buscou-se apenas analisar a qualidade das saídas providas pela transcrição de documentos através de ferramentas comerciais de OCR. Para aquisição de documentos transcritos utilizou-se a ferramenta Omnipage Professional 15.0 [51], devido à sua disponibilidade e alto desempenho [31].

Para estas medições foram utilizados 50 documentos, totalizando 13.532 palavras e 2.095.596 caracteres, digitalizando-os a 100 dpi, 150 dpi e 300 dpi, através de *scanner* Hewlett-Packard, modelo 5300c, e 3.2 Mpixels e 4.1 Mpixels através de duas câmeras digitais. Observe-se que foram consideradas palavras as seqüências de caracteres de número igual ou superior a três caracteres. Os resultados, após transcritos, tiveram seus erros classificados, em termos de palavras e caracteres [1].

4.2 Legibilidade e subjetividade

Devido ao sistema visuo-neural dos seres humanos ser extremamente complexo, ainda não há consenso no método estabelecido pelo cérebro para identificação de caracteres. Evidências dos trabalhos nos últimos 20 anos indicam que os seres humanos utilizam letras de uma palavra para identificá-la como um todo [24]. Três categorias de modelos de reconhecimento de palavras são mais utilizados: o modelo de contorno das palavras, que sugere que palavras são reconhecidas como unidades completas, o modelo serial, que afirma que palavras são lidas letra-a-letra e o modelo paralelo de reconhecimento de letras, o qual sugere que letras de uma palavra são reconhecidas simultaneamente, e a informação das letras são utilizadas para reconhecer a palavra, sendo este o mais utilizado.

Apesar da complexidade dos algoritmos utilizados por ferramentas de OCR, ainda há muito a ser implementado para que estas sejam aptas a transpor documentos com níveis altos de ruído, distorções, baixa resolução, dentre outros problemas com os quais o olho humano lida com facilidade. Além disso, ferramentas de OCR atuam apenas em documentos binários [36], realizando esta binarização no caso de a entrada ser uma imagem colorida. Comparando com imagens binárias, sabe-se que a escala de cinza melhora a qualidade da imagem ao olho humano [34], portanto, com a popularização de câmeras digitais, busca-se a atuação de ferramentas de OCR em documentos em escala de cinza, de modo a lidar melhor com a baixa resolução [36].

A subjetividade tem papel importante na medição da qualidade de imagens. Em digitalização de acervos bibliográficos é comum a presença de um operador que qualifica a imagem como apta ou não, armazenando ou solicitando nova digitalização [38]. Um experimento realizado entre alunos do Centro de Informática da Universidade Federal de Pernambuco (CIn - UFPE), envolvendo 80 alunos, classificando 369 imagens, mostra a relação entre o reconhecimento e classificação correta pelos alunos e o reconhecimento por uma ferramenta de OCR (Tabela 4.1). As imagens foram extraídas de fotografias com 4.1 Mpixels, e digitalizadas com *scanner* a 150 dpi e 300 dpi de resolução. Tais imagens contendo caracteres isolados, palavras ou frases, inserindo-se ruído de sal e pimenta[1] e/ou borrando a imagem antes de realizar a binarização em três níveis.

Tabela 4.1: Reconhecimento do OCR em relação à classificação humana de qualidade.

Classificação	**Reconhecimento OCR**
Excelente	38,09%
Boa	19,05%
Regular	33,33%
Ruim	4,76%
Péssimo	0%

A Tabela 4.1 leva à conclusão de que os resultados obtidos no reconhecimento de caracteres por ferramentas comerciais de OCR não podem ser utilizados como medição objetiva de qualidade, mas que seu reconhecimento com alta taxa de acerto é indicador de que o documento provavelmente é de boa qualidade.

A busca por um índice objetivo de qualidade tem levado pesquisadores a montar modelos

[1] O ruído de sal e pimenta é caracterizado por pontos brancos e pretos que surgem aleatoriamente na imagem binária. Estes pontos aleatórios aplicados a uma imagem lembram sal e pimenta, daí o nome do ruído.

de degradação em textos e sugerir valores [3] [4] [20] [47]. Muitos desses modelos são difíceis de serem validados dados seus referenciais variados para a extração de indicadores de qualidade, havendo poucos métodos para validação [19]. Esta busca por um método efetivo de medição de qualidade em documentos digitalizados tem procurado prover transposição mais eficiente por ferramentas de OCR. É comum ter como medição de qualidade a extração de características dos caracteres e ruídos presentes na imagem [2].

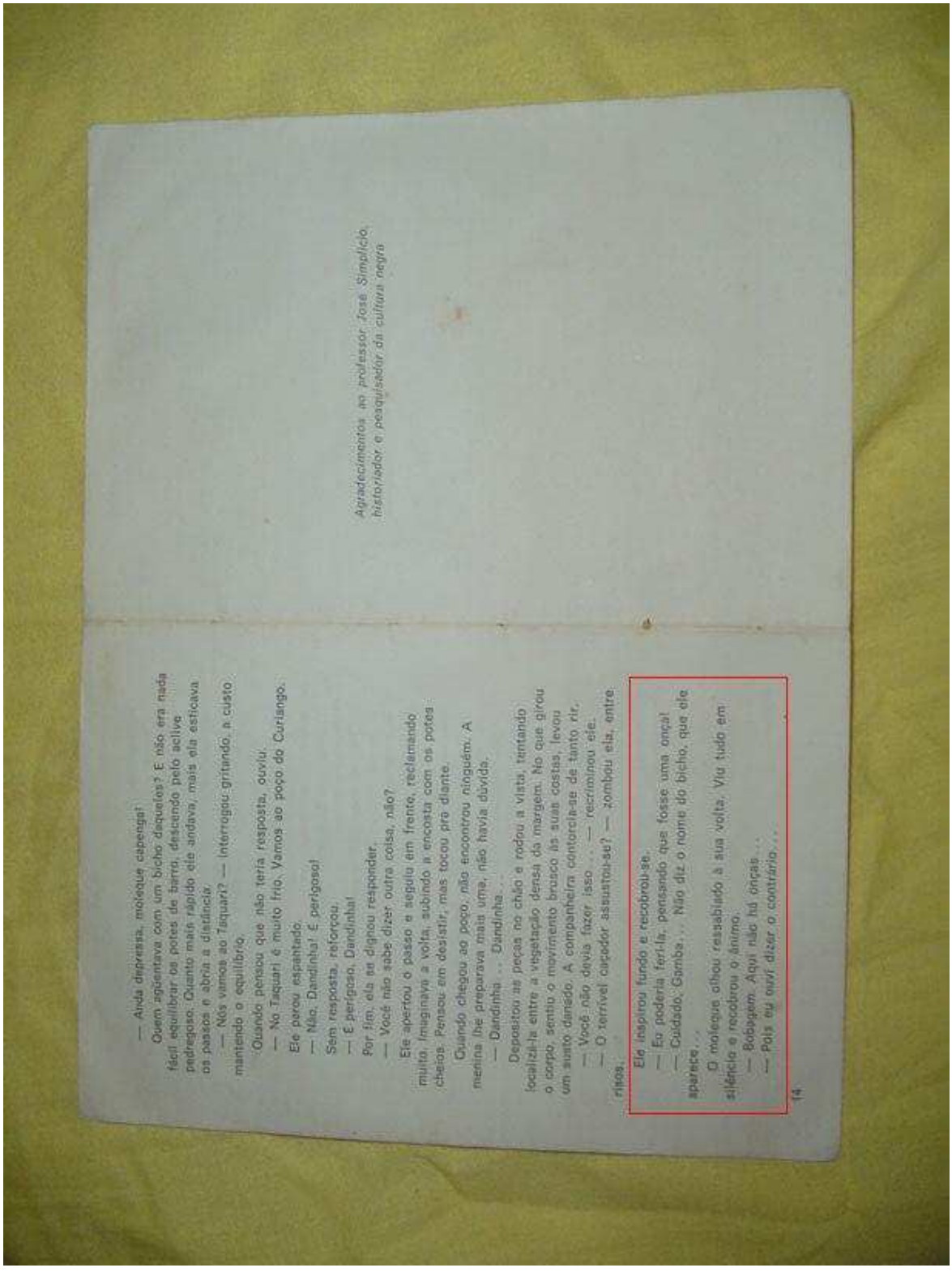
4.3 Pré-processamento e seus resultados

O pré-processamento dos documentos digitalizados por câmeras digitais busca melhorar a legibilidade por humanos, além de prover menor espaço para armazenamento e transmissão via rede de computadores, e melhorar a qualidade da saída em uma transcrição de imagem para texto. Nas subseções que seguem, são feitas medições e comparações, buscando identificar a influência de cada fator na qualidade da imagem.

4.3.1 *Flash* e iluminação inadequada

Apesar de os documentos obtidos tipicamente mostrarem diferentes regiões de brilho para um observador humano, e que isto impõe dificuldades para segmentação, binarização e remoção de bordas, a ferramenta de OCR utilizada não demonstra maiores erros em regiões de maior ou menor brilho desde que seja preservado contraste entre a fonte do documento e seu papel. Isto leva à conclusão que o algoritmo de binarização utilizado pela ferramenta de OCR é adaptativo.

Experimentos demonstraram que em ambientes com pouca iluminação, onde o efeito do *flash* torna-se facilmente visível, encontra-se maior quantidade de erros de transposição, levando à conclusão que embora o algoritmo para binarização utilizado pela ferramenta de OCR seja adaptativo, não é suficientemente eficiente para contornar o problema criado pela iluminação irregular. A região da imagem que apresenta maior atuação do *flash*, e conseqüentemente melhor iluminação, apresenta menores erros durante a transcrição, sendo que o *flash* demonstra ser ineficiente na aplicação ao documento como um todo. A Figura 4.1 ilustra a transposição de um trecho de menor brilho em um documento fotografado.




```
le inspirou fundo e recobrou-se.          a
-       Eu poderia feri-la, pensando que fosse uma onça!
-       Cuidado. Gamba... Não diz o nome do bicho, que ele
1:( moleque olhou ressabiado à sua volta. Viu tudo em
Hi^^ e recobrou o ânimo.
-       Bobagem. Aqui não há onças...
-       iDis eu ouvi dizer o contrário,
```

Figura 4.1: Exemplo de transcrição em região de menor brilho.

4.3.2 Correção da inclinação

Mesmo em documentos digitalizados por *scanners* é comum o surgimento de inclinação do documento devido ao fato de que nem sempre o documento é posto corretamente na superfície do *scanner*, seja pelo operador ou pela alimentação automática do *scanner*.

Para seres humanos a rotação de imagens é desagradável e introduz dificuldade na leitura do texto. No caso de reconhecimento de textos por computadores, a inclinação de documentos representa a inserção de diversos elementos prejudiciais à visão computacional, tais como maior espaço para armazenamento e maior captação de erros em reconhecimento e na transcrição de documentos por ferramentas de OCR [45]. Esses elementos fazem com que a correção da inclinação seja uma fase comum ao pré-processamento em qualquer ambiente de processamento de imagens.

No caso de documentos digitalizados por câmeras digitais sem suporte mecânico, este problema é existente em quase todos os documentos. A medição da inclinação nos documentos analisados resultou em inclinações sempre menores que 2 graus. Para esta medição, foram ignoradas as distorções geométricas, introduzidas pelas curvaturas das lentes das câmeras. Os vértices inferiores foram utilizados para traçar-se uma reta, da qual foi possível extrair-se esta inclinação.

A maioria das ferramentas comerciais de OCR compensam automaticamente inclinações de até 15 graus [27]. As distorções geométricas ou de perspectiva podem gerar pequenas variações no ângulo de inclinação do texto, em diferentes regiões. Nos documentos estudados esta variação foi inferior a 0,2 graus, e portanto, não gerando erros relevantes no OCR, seja comparado no documento como um todo ou linha-a-linha.

4.3.3 Remoção de bordas

As desvantagens e problemas introduzidos pela presença de bordas em documentos digitalizados são os mesmos para *scanners* e câmeras digitais, embora a remoção de bordas nesses dois casos não seja relacionada. O fator principal é que o ambiente de digitalização por câmeras digitais é imprevisível.

A presença de bordas em documentos adquiridos por câmeras digitais afeta diretamente os algoritmos de segmentação das ferramentas de OCR, gerando aumento na quantidade de erros em palavras e caracteres. Essa quantidade de erros varia dependendo da complexidade da borda. Para melhor análise dos documentos, realizou-se a remoção de bordas dos documentos

como fase de pré-processamento. Devido à presença de distorções de perspectiva, o método utilizado para a remoção de bordas faz com que vestígios das bordas permaneçam na imagem. Estas bordas vestigiais foram substituídas pela cor mais freqüente na área central da imagem, conforme ilustrado nas Figuras 4.2 e 4.3. Através do histograma apresentado nas duas imagens percebe-se a redução na quantidade de cores na imagem cuja borda foi removida. A escolha da cor de substituição deu-se em virtude das técnicas conhecidas de binarização de documentos, pois o brilho das cores presentes nas regiões da imagem analisada afetam diretamente o valor do limiar de binarização. Por exemplo, caso as bordas vestigiais fossem substituídas pela cor branca, o nível de limiar para binarização do documento seria mais alto, podendo levar à degradação dos caracteres próximos.

4.3.4 Distorções geométricas

O formato esférico das lentes de câmeras digitais e a proximidade do documento no momento da digitalização introduzem distorções em linhas retas que não são observadas em *scanners*. Nos documentos analisados foram verificadas as distâncias entre o limiar do documento e linhas retas traçadas entre as extremidades. As medições levaram à conclusão que a distorção máxima é correspondente a um valor inferior a 3% do número de pixels em uma linha da imagem, apresentando média de 18 pixels para os documentos fotografados a 3.2 Mpixels e de 23 pixels para os documentos fotografados a 4.1 Mpixels. Devido ao fato dessas distorções estarem espalhadas ao longo dos documentos, não se observou maiores degradações nas extremidades dos documentos, região onde estas distorções são mais acentuadas. A Figura 4.4 ilustra como foi realizada a medição desta distorção, sendo medida a região destacada em verde.

4.3.5 Distorções de perspectiva

A aquisição de documentos por câmeras digitais sem suporte mecânico paralelo ao plano do documento invariavelmente conduz a distorção de perspectiva. O objetivo desta etapa de pré-processamento foi medir a influência desta distorção no resultado provido pela ferramenta de OCR.

Para esta análise, após a remoção de bordas, dividiu-se a imagem em quatro partes, medindo-se os erros de caracteres em relação às regiões. Qualquer caracter constante no documento que não constasse na transcrição, que se apresentasse substituído por outro, ou

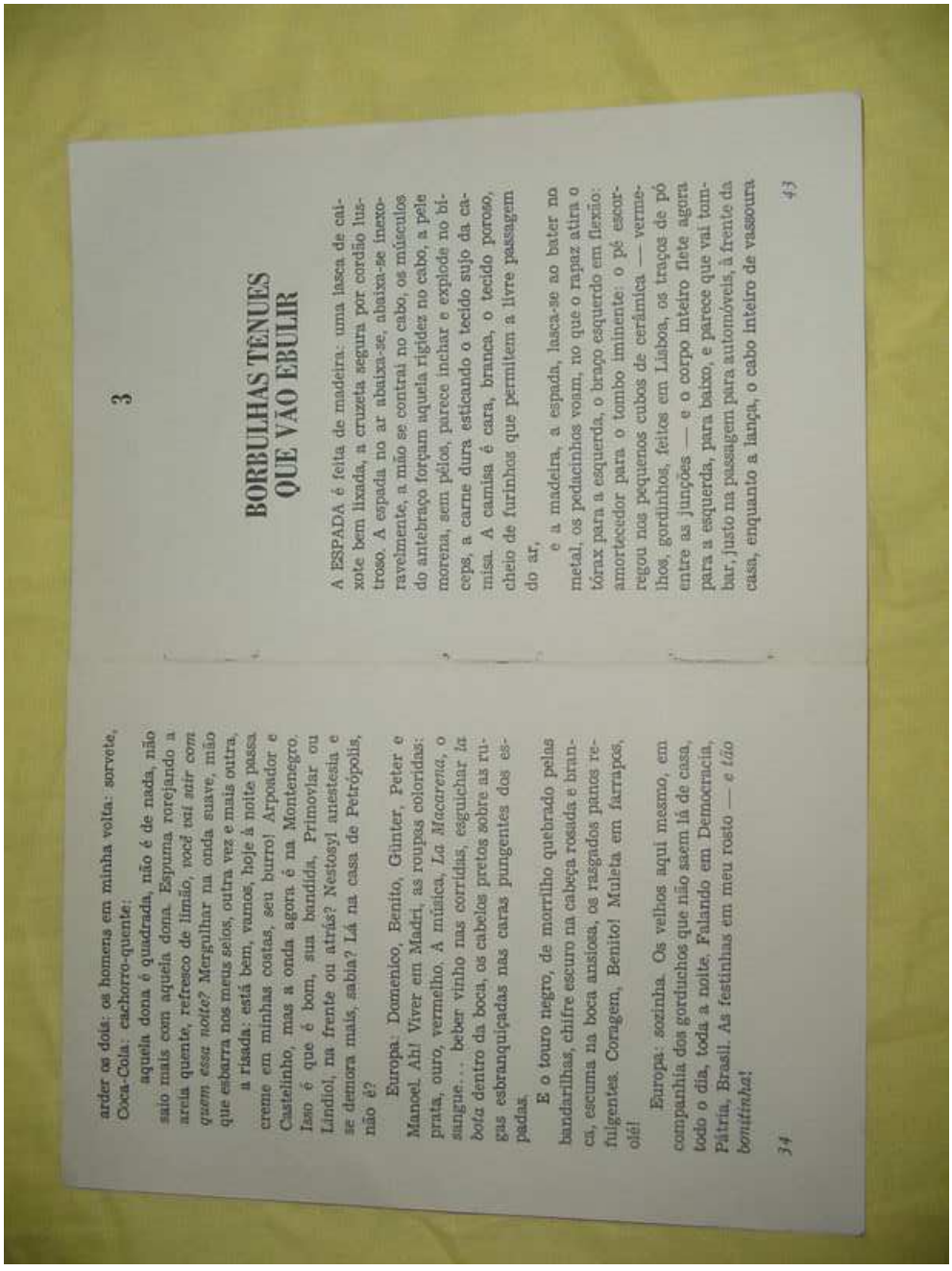
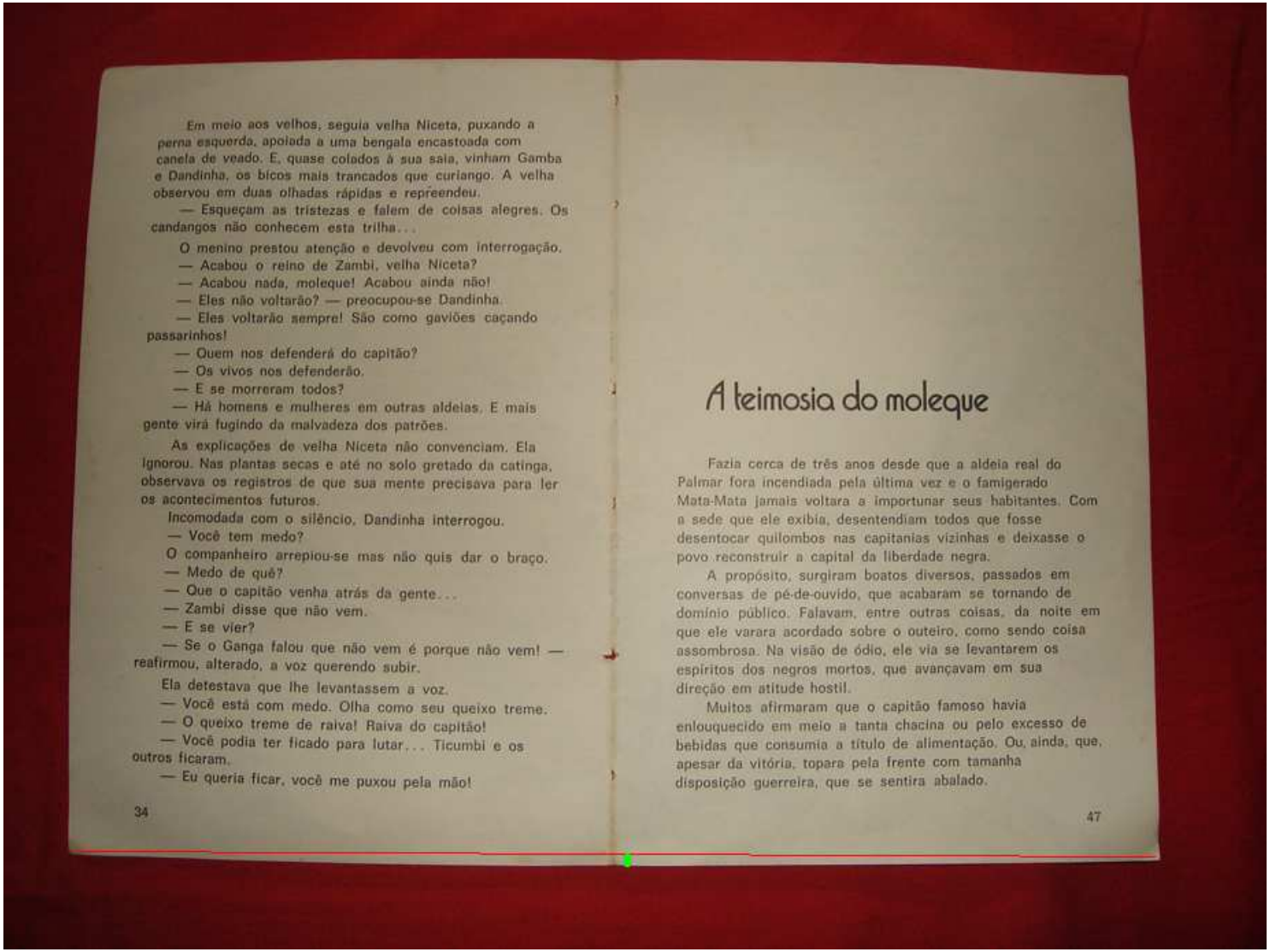


Figura 4.2: Documento digitalizado com borda.

Figura 4.3: Documento recortado com borda remanescente substituída.



Em meio aos velhos, seguia velha Niceta, puxando a perna esquerda, apoiada a uma bengala encastoada com canela de veado. E, quase colados à sua saia, vinham Gamba e Dandinha, os bicos mais trancados que curiango. A velha observou em duas olhadas rápidas e repreendeu.

— Esqueçam as tristezas e falem de coisas alegres. Os candangos não conhecem esta trilha...

O menino prestou atenção e devolveu com interrogação.

— Acabou o reino de Zambi, velha Niceta?

— Acabou nada, moleque! Acabou ainda não!

— Eles não voltarão? — preocupou-se Dandinha.

— Eles voltarão sempre! São como gaviões caçando passarinhos!

— Quem nos defenderá do capitão?

— Os vivos nos defenderão.

— E se morreram todos?

— Há homens e mulheres em outras aldeias. E mais gente virá fugindo da malvadeza dos patrões.

As explicações de velha Niceta não convenciam. Ela ignorou. Nas plantas secas e até no solo gretado da caatinga, observava os registros de que sua mente precisava para ler os acontecimentos futuros.

Incomodada com o silêncio, Dandinha interrogou.

— Você tem medo?

O companheiro arrepiou-se mas não quis dar o braço.

— Medo de quê?

— Que o capitão venha atrás da gente...

— Zambi disse que não vem.

— E se vier?

— Se o Ganga falou que não vem é porque não vem! — reafirmou, alterado, a voz querendo subir.

Ela detestava que lhe levantassem a voz.

— Você está com medo. Olha como seu queixo treme.

— O queixo treme de raiva! Raiva do capitão!

— Você podia ter ficado para lutar... Ticumbi e os outros ficaram.

— Eu queria ficar, você me puxou pela mão!

34

A teimosia do moleque

Fazia cerca de três anos desde que a aldeia real do Palmar fora incendiada pela última vez e o famigerado Mata-Mata jamais voltara a importunar seus habitantes. Com a sede que ele exibia, desentendiam todos que fosse desentocar quilombos nas capitanias vizinhas e deixasse o povo reconstruir a capital da liberdade negra.

A propósito, surgiram boatos diversos, passados em conversas de pé-de-ouvido, que acabaram se tornando de domínio público. Falavam, entre outras coisas, da noite em que ele varara acordado sobre o outeiro, como sendo coisa assombrosa. Na visão de ódio, ele via se levantarem os espíritos dos negros mortos, que avançavam em sua direção em atitude hostil.

Muitos afirmaram que o capitão famoso havia enlouquecido em meio a tanta chacina ou pelo excesso de bebidas que consumia a título de alimentação. Ou, ainda, que, apesar da vitória, topara pela frente com tamanha disposição guerreira, que se sentira abalado.

47

Figura 4.4: Ilustração da distorção geométrica.

que fosse inserido indevidamente na transcrição foi classificado como erro de caracter. O resultado pode ser observado nas Tabelas 4.2 e 4.3. A variância apresentada na Tabela 4.3 é dada pela expressão matemática

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1},$$

em que n é a quantidade total de documentos, x_i representa a quantidade de erros de caracteres na imagem i e $\bar{x}$ a média aritmética dos erros encontrada.

Tabela 4.2: Percentagem dos erros por região da imagem.

Regiões	**3.2 Mpixels**	**4.1 Mpixels**
Superior	55,66%	60,58%
Inferior	44,34%	39,42%
Direita	55,66%	50%
Esquerda	44,34%	50%

Tabela 4.3: Variância da localização dos erros de caracteres.

Regiões	**3.2 Mpixels**	**4.1 Mpixels**
Superior	6,357	4,059
Inferior	3,413	4,059
Direita	5,500	7,146
Esquerda	4,801	2,834

Pode-se concluir através destes resultados que distorções de perspectiva causam aumento na quantidade de erros na transcrição por ferramentas de OCR, influenciando consideravelmente na qualidade do documento.

4.4 Análise cumulativa

Após as análises realizadas no pré-processamento, os erros gerados na transposição das imagens para o formato de texto foram medidos e comparados. Os erros foram classificados como: substituição de caracteres, ausência de caracteres, inserção de caracteres, erros de acentuação, erros de palavra, ausência de palavras, erros de pontuação. Sendo que um erro de caractere causa erro de palavra caso a seqüência de caracteres seja superior a três caracteres.

Foram desconsiderados os erros de palavras para aquelas com quantidade de caracteres inferior a três caracteres para evitar conectivos e caracteres isolados. As Tabelas 4.4 e 4.5

ilustram os resultados obtidos em valor absoluto, enquanto as Tabelas 4.6, 4.7, 4.8 e 4.9 ilustram os resultados estatísticos. É possível observar através da Tabela 4.4 que a quantidade de erros nas transcrições das imagens de documentos adquiridas por câmera digital a 4.1 Mpixels foi menor do que as imagens adquiridas por scanner a 100 dpi, e a quantidade de ausências é inferior à quantidade apresentada nas imagens adquiridas por scanners a 150 dpi. Observa-se através das Tabelas 4.7 e 4.9 que a moda dos erros de transcrições nos documentos fotografados aproximou-se muito da moda apresentada nos documentos adquiridos por scanners, o que demonstra que a aquisição por câmeras digitais é uma forma viável de aquisição, mesmo para a aplicação em ferramentas de OCR, as quais foram desenvolvidas para atuação em documentos digitalizados por scanners. As variâncias dos erros, apresentadas nas Tabelas 4.6 e 4.8, levam à conclusão que o aumento da resolução contribui para melhor distribuição dos erros, assim como a redução dos erros, observada na Tabela 4.4.

Tabela 4.4: Erros de caracteres encontrados nas imagens (valor absoluto).

Tipo do erro	**100 dpi**	**150 dpi**	**300 dpi**	**3.2 Mpixels**	**4.1 Mpixels**
Substituição	74	38	24	104	84
Pontuação	123	3	1	40	12
Acentuação	22	4	10	83	22
Ausência	38	27	5	17	16
Inserção	43	92	19	25	19

Tabela 4.5: Erros de palavras encontrados nas imagens (valor absoluto).

Tipo do erro	**100 dpi**	**150 dpi**	**300 dpi**	**3.2 Mpixels**	**4.1 Mpixels**
Erros de palavras	53	19	20	106	67
Ausência	-	-	-	5	1

A resolução média para aquisição destes documentos foi de 150dpi para as imagens capturadas a 3.2 Mpixels e 170dpi para as imagens com 4.1 Mpixels. O cálculo foi baseado no tamanho do documento físico e no tamanho das imagens após retiradas as bordas. Observa-se que as câmeras utilizadas na aquisição dessas imagens, embora fossem do mesmo fabricante, possuíam especificações diferentes, e que o ambiente no qual as imagens foram adquiridas foram variados.

Estes resultados levam à conclusão de que o uso de câmeras fotográficas digitais, mesmo que em ambientes não-ideais, é uma forma viável de aquisição de informação para transcrição de imagens para textos utilizando softwares comerciais de OCR. Tais softwares não foram

Tabela 4.6: Variância dos erros de caracteres encontrados nas imagens.

Tipo do erro	**100 dpi**	**150 dpi**	**300 dpi**	**3.2 Mpixels**	**4.1 Mpixels**
Substituição	2,948	2,308	0,948	31,367	3,324
Pontuação	16,049	2,820	0,02	2,16	0,227
Acentuação	0,659	0,075	0,694	3,535	1,231
Ausência	3,288	3,288	0,133	18,557	0,467
Inserção	5,469	16,912	1,056	0,867	0,689

Tabela 4.7: Moda dos erros de caracteres encontrados nas imagens.

Tipo do erro	**100 dpi**	**150 dpi**	**300 dpi**	**3.2 Mpixels**	**4.1 Mpixels**
Substituição	0 e 1 (bimodal)	0	0	1, 2 e 3 (trimodal)	1
Pontuação	0	0	0	0	0
Acentuação	0	0	0	1 e 0 (bimodal)	0
Ausência	0	0	0	0	0
Inserção	0	0	0	0	0

Tabela 4.8: Variância dos erros de palavras encontrados nas imagens.

Tipo do erro	**100 dpi**	**150 dpi**	**300 dpi**	**3.2 Mpixels**	**4.1 Mpixels**
Erros de palavras	1,69	0,363	0,816	5,209	2,147
Ausência	0	0	0	12,02	0,02

Tabela 4.9: Moda dos erros de palavras encontrados nas imagens.

Tipo do erro	**100 dpi**	**150 dpi**	**300 dpi**	**3.2 Mpixels**	**4.1 Mpixels**
Erros de palavras	0	0	0	amodal	1
Ausência	0	0	0	0	0

projetados para trabalhar com as adversidades encontradas nas imagens de documentos fotografados. Convém destacar que todos documentos digitalizados utilizados nesta avaliação eram legíveis a olho nu, quando visualizados em monitores de microcomputadores de tamanho igual ou superior a 15"ou quando impressos em impressoras jato de tinta, utilizando papel de tamanho A4, uma imagem por página e em modo colorido.

capítulo 5

A binarização de documentos

A binarização é o processo através do qual se converte uma imagem colorida ou monocromática (escala de cinza) em uma imagem preto-e-branco (binária). Imagens binárias são uma alternativa para documentos sem valor artístico ou iconográficos, visto que ocupam menor espaço para armazenamento e, conseqüentemente, requerem menor tempo para transmissão via rede de comunicação de dados.

Algoritmos de binarização geralmente fazem uso do nível de cinza da imagem como etapa intermediária. O nível de cinza de uma imagem pode ser definido como a distribuição dos tons de cinza em uma imagem monocromática (escala de cinza). A conversão de uma imagem colorida em imagem monocromática pode ser realizada através de diferentes cálculos, inclusive substituindo-se os valores das componentes R (vermelha) e B (azul), pelo valor da componente G (verde), entretanto, a fórmula mais conhecida para o cálculo do tom de cinza([10] *apud* [8]) para determinado pixel é dada por

$$\text{tom\_de\_cinza} = 0,3R + 0,59G + 0,11B.$$

Os algoritmos de binarização podem ser classificados em duas categorias: globais e adaptativos locais. Os algoritmos globais estabelecem um valor de limiar (*threshold*), aplicando esse valor à imagem como um todo. Por exemplo, se o valor de limiar for 128, todos os pixels com tons de cinza acima de 128 tornam-se brancos, enquanto os demais tornam-se pretos. Os algoritmos locais adaptativos possuem um valor inicial de limiar, podendo este valor ser alterado para diferentes regiões do documento, tais regiões são determinadas de acordo com

a implementação de cada algoritmo.

A aplicação de algoritmos de binarização desenvolvidos para atuação em *scanners* não produz bons resultados quando aplicados diretamente a documentos fotografados por câmeras digitais portáteis. A aplicação de algoritmos locais adaptativos produz melhores resultados do que os algoritmos globais.

A necessidade de binarização de documentos digitalizados por *scanners* levou ao desenvolvimentos de muitos algoritmos [44]. Tais algoritmos são utilizados como referência para o desenvolvimento de algoritmos mais eficientes no caso da digitalização por câmeras digitais [41][43]. Alguns algoritmos de binarização são apresentados sucintamente na próxima seção.

5.1 Algoritmos clássicos de binarização

Alguns métodos são utilizados no processo de binarização. O primeiro deles é adquirir um valor de limiar (*threshold*) e aplicar diretamente à imagem em tons de cinza. Um segundo método é o de seleção iterativa, em que um valor de limiar é pré-definido e ajustado ao longo das iterações. Tal valor inicial de limiar é definido como o ponto médio do histograma. O terceiro método aqui apresentado consiste na utilização da entropia da imagem. Entropia foi definida por Claude Shannon como medida de informação [42]. Considere-se E um evento que ocorre com probabilidade $p(E)$. Se E ocorreu, então diz-se que foram recebidas

$$I = log(\frac{1}{p(E)})$$

Se um evento tem probabilidade 1 de ocorrer então não há informação adicional. Entretanto, caso o evento tenha probabilidade baixa, sua ocorrência poderá trazer grande quantidade de informação.

Supondo que existem n possíveis símbolos x e que o símbolo i ocorre com probabilidade $p(x_i)$. Então a entropia associada com a fonte X dos símbolos é

$$H = -\sum_{i=1}^{n} p(x_i) log[p(x_i)],$$

onde a entropia é medida em bits/símbolo para uma fonte binária.

As subseções que seguem apresentam alguns algoritmos clássicos de binarização baseados em entropia e o algoritmo de Otsu, por sua eficiência e por ser um dos mais utilizados.

5.1.1 O algoritmo de Pun

O algoritmo de Pun [37] assume os níveis de cinza da imagem como uma fonte de 256 símbolos, onde todos os símbolos são estatisticamente independentes. A entropia associada aos pixels pretos (H_b) e a entropia associada aos pixels brancos (H_w) são delimitadas pelo valor de corte t. O algoritmo sugere que t seja tal que maximize a função $H = H_b + H_w$. Sendo H_b e H_w dados por

$$H_b = -\sum_{0}^{t} p(i)log(p(i))$$

e

$$H_w = -\sum_{t+1}^{255} p(i)log(p(i))$$

em que $p(i)$ é a probabilidade do pixel$i$ da cor $cor[i]$ na imagem, dada pela razão entre o número de aparições da $cor[i]$ pelo número total de pixels na imagem.

5.1.2 O algoritmo de Kapur, Sahoo e Wong

O algoritmo descrito nessa seção [21] considera o objeto e o fundo da imagem como sendo duas fontes distintas, conseqüentemente, tendo duas distribuições, uma para o objeto e outra para o fundo. A distribuição do objeto é dada por:

$$p(i) = \frac{p_i}{P(t)},$$

para $0 \leq i \leq t$, e a distribuição do fundo é dada por

$$p(i) = \frac{p_i}{1 - P(t)},$$

para $t + 1 \leq i \leq 255$.

Denota-se por H_b a entropia associada aos pixels pretos (objeto) e por H_w a entropia associada aos pixels brancos (fundo). Estas são, respectivamente, calculadas através das equações

$$H_b = -\sum_{0}^{t} p(i)log(p(i))$$

e

$$H_w = -\sum_{t+1}^{255} p(i)log(p(i))$$

5.1.3 O algoritmo de Johannsen e Bille

Esta é outra variação de algoritmo de binarização baseado em entropia [18]. busca-se como *threshold* ótimo o argumento t que minimize a função $S(t) = S_b(t) + S_w(t)$, onde

$$S_b(t) = log(\sum_{i=0}^{t} p_i) + (1/\sum_{i=0}^{t} p_i)[E(p_t) + E(\sum_{i=0}^{t-1} p_i)]$$

e

$$S_w(t) = log(\sum_{i=t}^{255} p_i) + (1/\sum_{i=t}^{255} p_i)[E(p_t) + E(\sum_{i=t+1}^{255} p_i)],$$

sendo $E(p) = -p * log(p)$.

5.1.4 O algoritmo de Yen, Chang e Chang

O algoritmo de Yen, Chang e Chang [48], assim como o algoritmo de Kapur_Sahoo_Wong, considera o objeto e o fundo da imagem como sendo duas fontes de sinais distintas. Com isso temos as mesmas distribuições apresentadas na seção 5.1.2, dadas por:

$$H_b = -\sum_{0}^{t} p(i)log(p(i))$$

e

$$H_w = -\sum_{t+1}^{255} p(i)log(p(i)).$$

É definida uma relação entrópica dada por

$$TC(t) = C_b(t) + C_w(t) = -log(\sum_{i=0}^{t}(\frac{p_i}{P(t)})^2) - log(\sum_{i=t+1}^{255}(\frac{p_i}{1-P(t)})^2)$$

O limiar de corte é o argumento que maximiza a equação acima. As funções $C_b(t)$ e $C_w(t)$ são entropias de Ranyi, com $\rho = 2$.

5.1.5 O algoritmo Mello e Lins

O algoritmo Mello e Lins [30] [32], faz uso do nível de cinza mais freqüente na imagem como um limite t (inicial) para calcular os valores de $H_b(t)$, $H_w(t)$ e H. Nesse algoritmo a entropia é calculada na base N, em que N é o total de pontos da imagem.

$$H = H_b(t) + H_w(t) = -\sum_{i=0}^{t} p_i log_N p_i - \sum_{i=t+1}^{255} p_i log_N p_i.$$

O valor da entropia H é utilizado para a definição de dois fatores multiplicativos, m_w e m_b, de forma que:

$$\begin{cases} m_w = 2 \text{ e } m_b = 3 & \text{se } H \leq 0,25 \\ m_w = 1 \text{ e } m_b = 2,6 & \text{se } 0,25 < H < 0,30 \\ m_w = 1 \text{ e } m_b = 1 & \text{se } H \geq 0,30 \end{cases}$$

O limiar de binarização (T) é dado por:

$$T = 256(m_b H_b + m_w H_w).$$

5.1.6 O algoritmo de da_Silva, Lins e Rocha

O algoritmo de da_Silva-Lins-Rocha [7] também faz uso de entropia, e as etapas são as seguites:

- Calcula-se a entropia H do histograma da imagem em escala de cinza

$$H = -\sum_{i=0}^{255} p_i log_2(p_i),$$

em que $\{p_0, p_1, ..., p_{255}\}$ é uma distribuição *a priori*dada por

$$p_i = \frac{o\ número\ de\ pixels\ com\ escala\ de\ cinza\ de\ nivel\ i\ (0\ a\ 255)}{o\ número\ total\ de\ pixels\ da\ imagem}.$$

- Para cada nível t calcula-se as distribuições *a posteriori*

$$\{P_t = \sum_{i=0}^{t} p_i, 1 - P_t\},$$

enquanto $P_t \leq 0,5$, a entropia associada com a seguinte distribuição:

$$H'(t) = -P_t log(P_t) - (1 - P_t) log(1 - P_t).$$

- Finalmente determina-se o limite que minimiza a expressão dada por

$$|e(t)| = |\frac{H'(t)}{H/log(256)} - \alpha(H/log(256)),$$

em que α é um fator de perda dado por

$$\alpha(H/log(256)) = \begin{cases} -\frac{3}{7} H/log(256) + 0,8 & \text{se } H/log(256) < 0,7 \\ H/log(256) - 0,2 & \text{se } H/log(256) \geq 0,7 \end{cases}$$

5.1.7 O algoritmo de Wu, Songde e Hanqing

Esse algoritmo desenvolvido inicialmente para aplicação em imagens de ultrasom [46] mostrou ser aplicável em imagens de documentos manuscritos históricos [7]. Nesse algoritmo,

assim como o algoritmo de Pun, previamente descrito, também assume que a imagem é formada de duas partes distintas: o objeto e o fundo da imagem. Assim sendo:

$$H_b(t) = -\sum_{i=0}^{t} \frac{p_i}{P_t} ln \frac{p_i}{P_t},$$

$$H_w(t) = -\sum_{i=0}^{255} \frac{p_i}{1-P_t} ln \frac{p_i}{1-P_t},$$

em que

$$P_t = \sum_{i=0}^{t} p_i$$

.

Como última etapa desse algoritmo, busca-se encontrar o valor de nível de cinza que minimize as diferenças entre as entropias do objeto e fundo.

$$t = \min_{t \in G} |H_b(t) - H_w(t)|,$$

em que G representa todos os níveis de cinza presentes na imagem.

5.1.8 O algoritmo de Otsu

Conforme mencionado anteriormente, o algoritmo de Otsu [35] não faz uso de entropia. Esse algoritmo faz uso de medidas da análise de discriminante para definir se os níveis de cinza pertencem ao objeto ou ao fundo. A média e a variância do objeto e do fundo, em função do limite t, podem ser denotadas por

$$m_b(t) = \sum_{i=0}^{t} i(p_i),$$

$$\rho_b^2(t) = \sum_{i=0}^{t} (i - m_b(t))^2 p_i,$$

$$m_w(t) = \sum_{i=t+1}^{255} i(p_i),$$

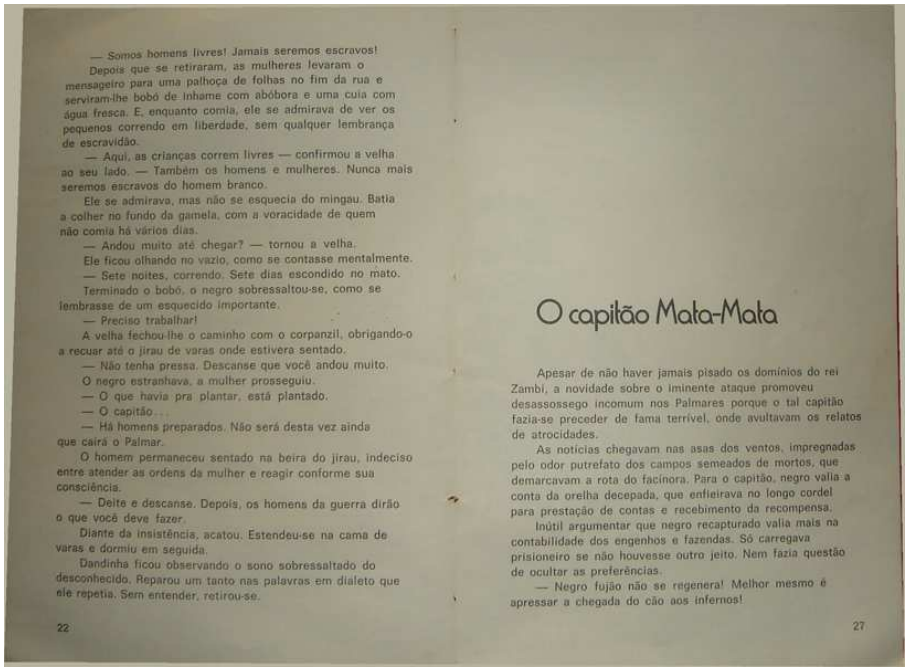
$$\rho_w^2(t) = \sum_{i=t+1}^{255} (i - m_w(t))^2 p_i.$$

O valor do limiar de binarização é o argumento que maximiza

$$\eta(t) = \frac{P(t)(1-P(t))(m_b(t) - m_w(t))^2}{P(t)\rho_b^2(t) + (1-P(t))\rho_w^2(t)}.$$

5.2 Algoritmos de binarização aplicados a documentos fotografados

A maioria das ferramentas de OCR processa as imagens de entrada no formato binário antes de realizar o reconhecimento. Portanto, é importante analisar o desempenho de algoritmos de binarização em documentos fotografados. Realizou-se esta etapa do pré-processamento com o objetivo de analisar possíveis melhorias na qualidade do documento digitalizado, verificando-se sua legibilidade por humanos e a saída provida pela ferramenta de OCR. Foram utilizados os oito algoritmos de binarização apresentados na seção anterior, todos esses do tipo global (aplicando o mesmo valor do limiar de binarização à toda área analisada). Esses algoritmos foram aplicados às imagens dos documentos adquiridos pelas câmeras digitais de forma global e separando-as em três, seis, nove e dezoito regiões. O resultado da aplicação desses algoritmos à imagem contida na Figura 5.1 pode ser observado nas Figuras 5.2 a 5.17.



— Somos homens livres! Jamais seremos escravos!

Depois que se retiraram, as mulheres levaram o mensageiro para uma palhoça de folhas no fim da rua e serviram-lhe bobó de inhame com abóbora e uma cuia com água fresca. E, enquanto comia, ele se admirava de ver os pequenos correndo em liberdade, sem qualquer lembrança de escravidão.

— Aqui, as crianças correm livres — confirmou a velha ao seu lado. — Também os homens e mulheres. Nunca mais seremos escravos do homem branco.

Ele se admirava, mas não se esquecia do mingau. Batia a colher no fundo da gamela, com a voracidade de quem não comia há vários dias.

— Andou muito até chegar? — tornou a velha.

Ele ficou olhando no vazio, como se contasse mentalmente.

— Sete noites, correndo. Sete dias escondido no mato.

Terminado o bobó, o negro sobressaltou-se, como se lembrasse de um esquecido importante.

— Preciso trabalhar!

A velha fechou-lhe o caminho com o corpanzil, obrigando-o a recuar até o jirau de varas onde estivera sentado.

— Não tenha pressa. Descanse que você andou muito.

O negro estranhava, a mulher prosseguiu.

— O que havia pra plantar, está plantado.

— O capitão...

— Há homens preparados. Não será desta vez ainda que cairá o Palmar.

O homem permaneceu sentado na beira do jirau, indeciso entre atender as ordens da mulher e reagir conforme sua consciência.

— Deite e descanse. Depois, os homens da guerra dirão o que você deve fazer.

Diante da insistência, acatou. Estendeu-se na cama de varas e dormiu em seguida.

Dandinha ficou observando o sono sobressaltado do desconhecido. Reparou um tanto nas palavras em dialeto que ele repetia. Sem entender, retirou-se.

22

O capitão Mata-Mata

Apesar de não haver jamais pisado os domínios do rei Zambi, a novidade sobre o iminente ataque promoveu desassossego incomum nos Palmares porque o tal capitão fazia-se preceder de fama terrível, onde avultavam os relatos de atrocidades.

As notícias chegavam nas asas dos ventos, impregnadas pelo odor putrefato dos campos semeados de mortos, que demarcavam a rota do facínora. Para o capitão, negro valia a conta da orelha decepada, que enfileirava no longo cordel para prestação de contas e recebimento da recompensa.

Inútil argumentar que negro recapturado valia mais na contabilidade dos engenhos e fazendas. Só carregava prisioneiro se não houvesse outro jeito. Nem fazia questão de ocultar as preferências.

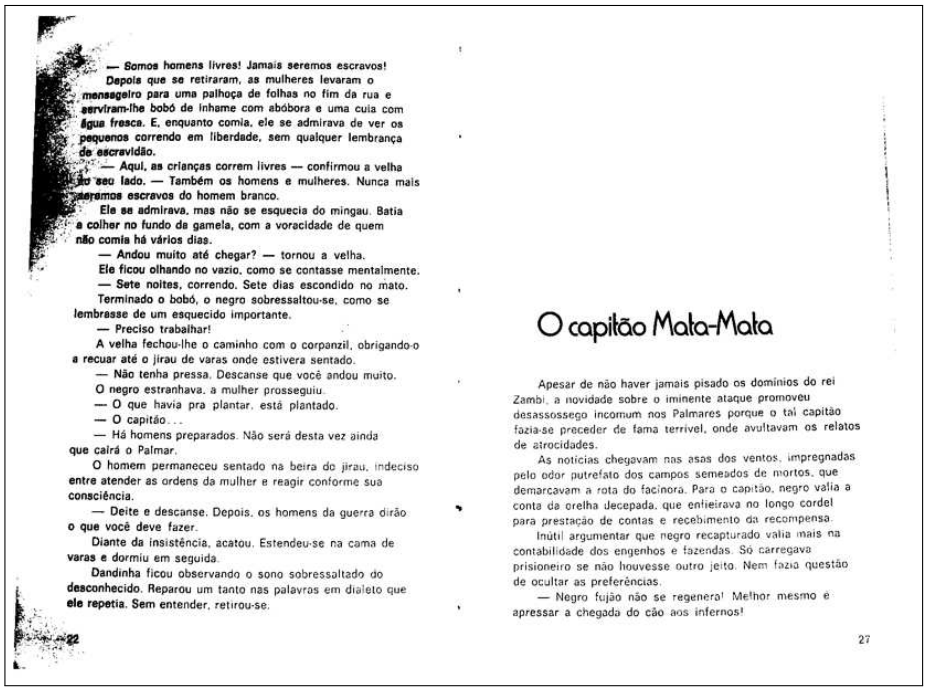
— Negro fujão não se regenera! Melhor mesmo é apressar a chegada do cão aos infernos!

27

Figura 5.1: Documento em true color (24 bits).

Aplicando-se os algoritmos de forma global, o algoritmo que apresentou melhor resultado no reconhecimento de caracteres foi o algoritmo da_Silva-Lins-Rocha [7], enquanto, aplicando-se em regiões do documento, o melhor resultado foi provido pelo algoritmo Kapur-Sahoo-Wong [21] em 18 regiões. É possível observar que na aplicação dos algoritmos de forma global

os algoritmos de Otsu e Kapur-Sahoo-Wong também produziram bons resultados, entretanto as falhas que surgem na imagem devido ao nível mais alto de *threshold* causam aumento na quantidade de erros nas transcrições por ferramentas de OCR.



— Somos homens livres! Jamais seremos escravos!

Depois que se retiraram, as mulheres levaram o mensageiro para uma palhoça de folhas no fim da rua e serviram-lhe bobó de inhame com abóbora e uma cuia com água fresca. E, enquanto comia, ele se admirava de ver os pequenos correndo em liberdade, sem qualquer lembrança de escravidão.

— Aqui, as crianças correm livres — confirmou a velha ao seu lado. — Também os homens e mulheres. Nunca mais seremos escravos do homem branco.

Ele se admirava, mas não se esquecia do mingau. Batia a colher no fundo da gamela, com a voracidade de quem não comia há vários dias.

— Andou muito até chegar? — tornou a velha.

Ele ficou olhando no vazio, como se contasse mentalmente.

— Sete noites, correndo. Sete dias escondido no mato.

Terminado o bobó, o negro sobressaltou-se, como se lembrasse de um esquecido importante.

— Preciso trabalhar!

A velha fechou-lhe o caminho com o corpanzil, obrigando-o a recuar até o jirau de varas onde estivera sentado.

— Não tenha pressa. Descanse que você andou muito.

O negro estranhava, a mulher prosseguiu.

— O que havia pra plantar, está plantado.

— O capitão...

— Há homens preparados. Não será desta vez ainda que cairá o Palmar.

O homem permaneceu sentado na beira do jirau, indeciso entre atender as ordens da mulher e reagir conforme sua consciência.

— Deite e descanse. Depois, os homens da guerra dirão o que você deve fazer.

Diante da insistência, acatou. Estendeu-se na cama de varas e dormiu em seguida.

Dandinha ficou observando o sono sobressaltado do desconhecido. Reparou um tanto nas palavras em dialeto que ele repetia. Sem entender, retirou-se.

22

O capitão Mata-Mata

Apesar de não haver jamais pisado os domínios do rei Zambi, a novidade sobre o iminente ataque promoveu desassossego incomum nos Palmares porque o tal capitão fazia-se preceder de fama terrível, onde avultavam os relatos de atrocidades.

As notícias chegavam nas asas dos ventos, impregnadas pelo odor putrefato dos campos semeados de mortos, que demarcavam a rota do facínora. Para o capitão, negro valia a conta da orelha decepada, que enfieirava no longo cordel para prestação de contas e recebimento da recompensa.

Inútil argumentar que negro recapturado valia mais na contabilidade dos engenhos e fazendas. Só carregava prisioneiro se não houvesse outro jeito. Nem fazia questão de ocultar as preferências.

— Negro fujão não se regenera! Melhor mesmo é apressar a chegada do cão aos infernos!

27

Figura 5.2: Documento da Figura 5.1 binarizado pelo algoritmo daSilva-Lins-Rocha globalmente.

Analisando as imagens binarizadas foi possível observar redução da quantidade de erros por substituições de caracteres no reconhecimento, entretanto aumentou-se o número de erros por inserção de caracteres devido ao ruído inserido na imagem. Conclui-se que os oito algoritmos utilizados não podem ser aplicados diretamente na imagem, mas apresentam resultados promissores na aplicação de algoritmos para pós-processamento, buscando eliminar os ruídos causados durante o processo de binarização. Como pode ser observado as figuras 5.10 e 5.13 há ruído de sal-e-pimenta, assim como a transformação de preto em regiões que deveriam ser brancas, tais ruídos podem ser tratados através da aplicação de filtros, motivando análise futura.

Figura 5.3: Documento da Figura 5.1 binarizado pelo algoritmo Mello-Lins globalmente.

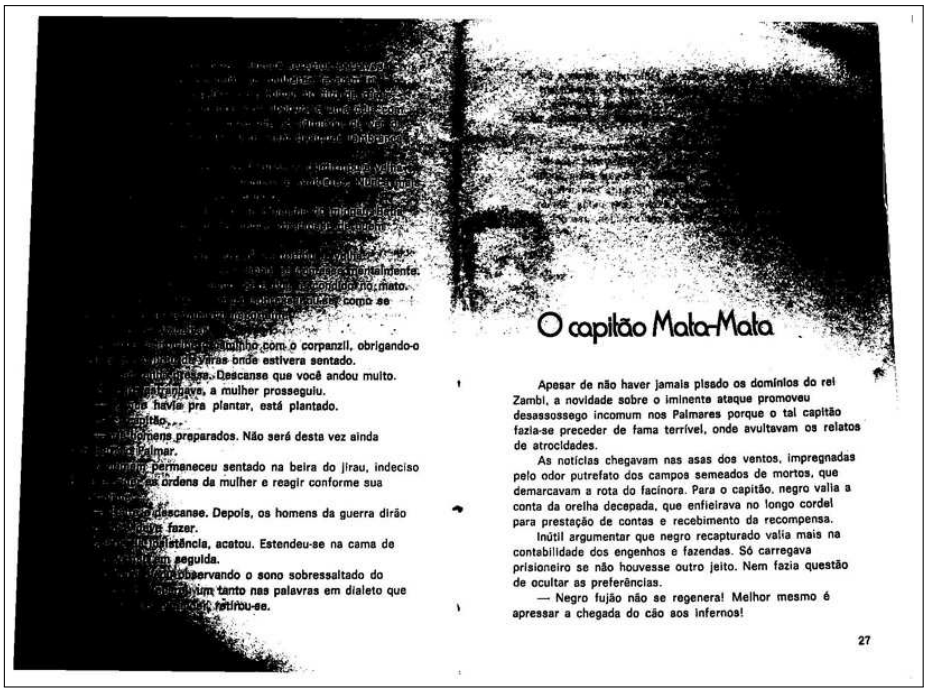
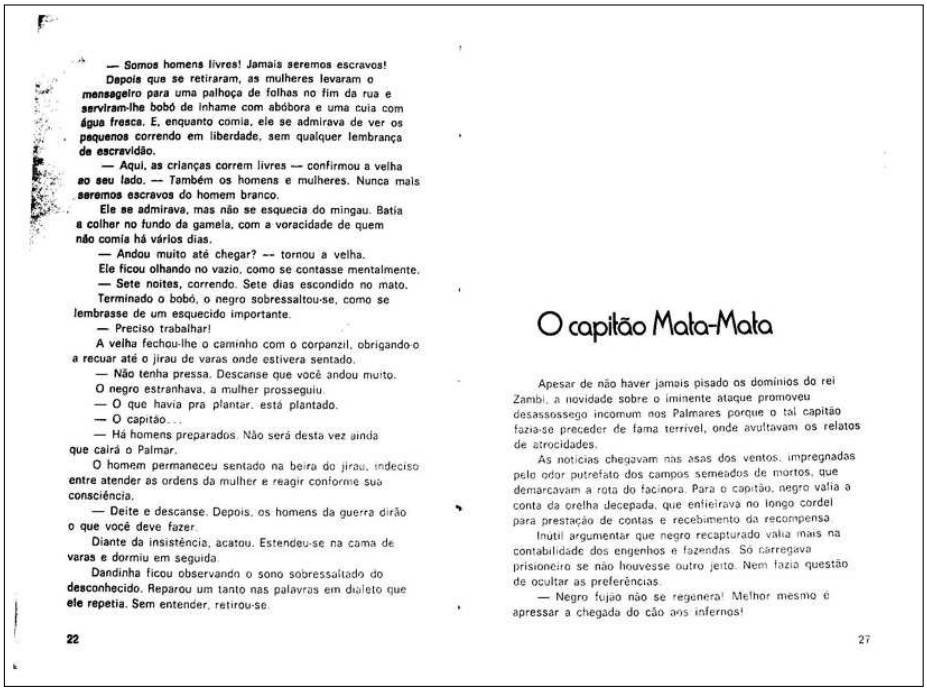


Figura 5.4: Documento da Figura 5.1 binarizado pelo algoritmo Pun globalmente.



— Somos homens livres! Jamais seremos escravos!

Depois que se retiraram, as mulheres levaram o mensageiro para uma palhoça de folhas no fim da rua e serviram-lhe bobó de inhame com abóbora e uma cuia com água fresca. E, enquanto comia, ele se admirava de ver os pequenos correndo em liberdade, sem qualquer lembrança de escravidão.

— Aqui, as crianças correm livres — confirmou a velha ao seu lado. — Também os homens e mulheres. Nunca mais seremos escravos do homem branco.

Ele se admirava, mas não se esquecia do mingau. Batia a colher no fundo da gamela, com a voracidade de quem não comia há vários dias.

— Andou muito até chegar? — tornou a velha.

Ele ficou olhando no vazio, como se contasse mentalmente.

— Sete noites, correndo. Sete dias escondido no mato.

Terminado o bobó, o negro sobressaltou-se, como se lembrasse de um esquecido importante.

— Preciso trabalhar!

A velha fechou-lhe o caminho com o corpanzil, obrigando-o a recuar até o jirau de varas onde estivera sentado.

— Não tenha pressa. Descanse que você andou muito.

O negro estranhava, a mulher prosseguiu.

— O que havia pra plantar, está plantado.

— O capitão...

— Há homens preparados. Não será desta vez ainda que cairá o Palmar.

O homem permaneceu sentado na beira do jirau, indeciso entre atender as ordens da mulher e reagir conforme sua consciência.

— Deite e descanse. Depois, os homens da guerra dirão o que você deve fazer.

Diante da insistência, acatou. Estendeu-se na cama de varas e dormiu em seguida.

Dandinha ficou observando o sono sobressaltado do desconhecido. Reparou um tanto nas palavras em dialeto que ele repetia. Sem entender, retirou-se.

22

O capitão Mata-Mata

Apesar de não haver jamais pisado os domínios do rei Zambi, a novidade sobre o iminente ataque promoveu desassossego incomum nos Palmares porque o tal capitão fazia-se preceder de fama terrível, onde avultavam os relatos de atrocidades.

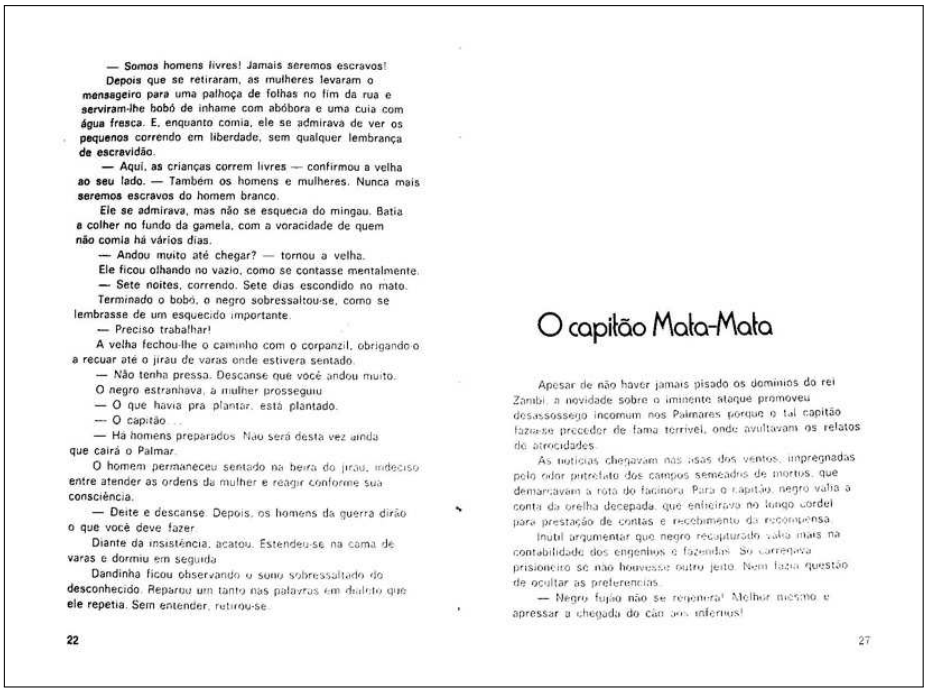
As notícias chegavam nas asas dos ventos, impregnadas pelo odor putrefato dos campos semeados de mortos, que demarcavam a rota do facínora. Para o capitão, negro valia a conta da orelha decepada, que enfileirava no longo cordel para prestação de contas e recebimento da recompensa.

Inútil argumentar que negro recapturado valia mais na contabilidade dos engenhos e fazendas. Só carregava prisioneiro se não houvesse outro jeito. Nem fazia questão de ocultar as preferências.

— Negro fujão não se regenera! Melhor mesmo é apressar a chegada do cão aos infernos!

27

Figura 5.5: Documento da Figura 5.1 binarizado pelo algoritmo Kapur-Sahoo-Wong globalmente.



— Somos homens livres! Jamais seremos escravos!

Depois que se retiraram, as mulheres levaram o mensageiro para uma palhoça de folhas no fim da rua e serviram-lhe bobó de inhame com abóbora e uma cuia com água fresca. E, enquanto comia, ele se admirava de ver os pequenos correndo em liberdade, sem qualquer lembrança de escravidão.

— Aqui, as crianças correm livres — confirmou a velha ao seu lado. — Também os homens e mulheres. Nunca mais seremos escravos do homem branco.

Ele se admirava, mas não se esquecia do mingau. Batia a colher no fundo da gamela, com a voracidade de quem não comia há vários dias.

— Andou muito até chegar? — tornou a velha.

Ele ficou olhando no vazio, como se contasse mentalmente.

— Sete noites, correndo. Sete dias escondido no mato.

Terminado o bobó, o negro sobressaltou-se, como se lembrasse de um esquecido importante.

— Preciso trabalhar!

A velha fechou-lhe o caminho com o corpanzil, obrigando-o a recuar até o jirau de varas onde estivera sentado.

— Não tenha pressa. Descanse que você andou muito.

O negro estranhava, a mulher prosseguiu.

— O que havia pra plantar, está plantado.

— O capitão...

— Há homens preparados. Não será desta vez ainda que cairá o Palmar.

O homem permaneceu sentado na beira do jirau, indeciso entre atender as ordens da mulher e reagir conforme sua consciência.

— Deite e descanse. Depois, os homens da guerra dirão o que você deve fazer.

Diante da insistência, acatou. Estendeu-se na cama de varas e dormiu em seguida.

Dandinha ficou observando o sono sobressaltado do desconhecido. Reparou um tanto nas palavras em dialeto que ele repetia. Sem entender, retirou-se.

22

O capitão Mata-Mata

Apesar de não haver jamais pisado os domínios do rei Zambi, a novidade sobre o iminente ataque promoveu desassossego incomum nos Palmares porque o tal capitão fazia-se preceder de fama terrível, onde avultavam os relatos de atrocidades.

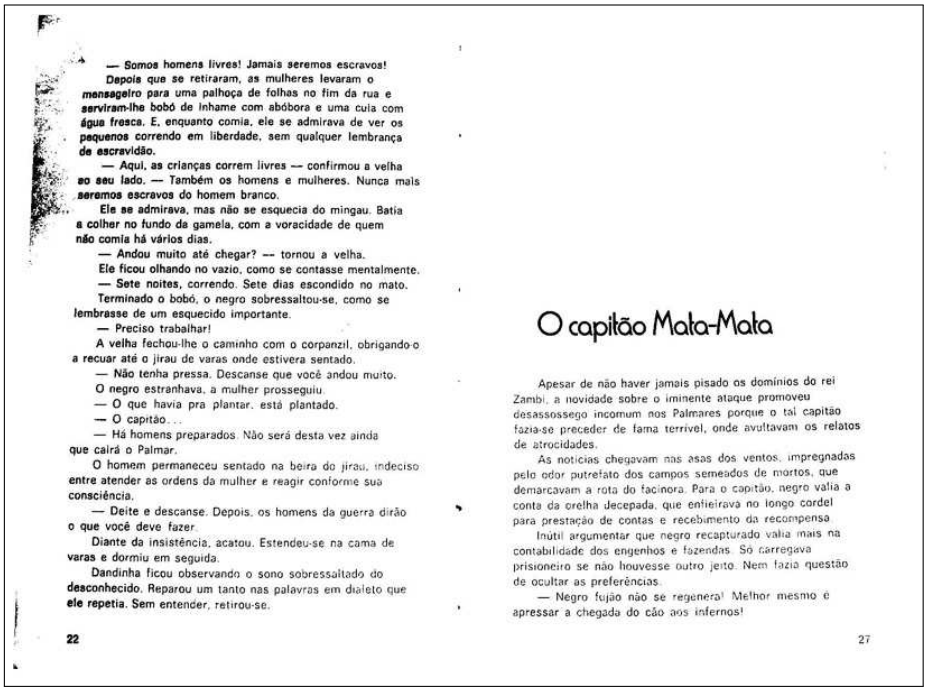
As notícias chegavam nas asas dos ventos, impregnadas pelo odor putrefato dos campos semeados de mortos, que demarcavam a rota do facínora. Para o capitão, negro valia a conta da orelha decepada, que enfileirava no longo cordel para prestação de contas e recebimento da recompensa.

Inútil argumentar que negro recapturado valia mais na contabilidade dos engenhos e fazendas. Só carregava prisioneiro se não houvesse outro jeito. Nem fazia questão de ocultar as preferências.

— Negro fujão não se regenera! Melhor mesmo é apressar a chegada do cão aos infernos!

27

Figura 5.6: Documento da Figura 5.1 binarizado pelo algoritmo Wulu globalmente.



— Somos homens livres! Jamais seremos escravos!

Depois que se retiraram, as mulheres levaram o mensageiro para uma palhoça de folhas no fim da rua e serviram-lhe bobó de inhame com abóbora e uma cuia com água fresca. E, enquanto comia, ele se admirava de ver os pequenos correndo em liberdade, sem qualquer lembrança de escravidão.

— Aqui, as crianças correm livres — confirmou a velha ao seu lado. — Também os homens e mulheres. Nunca mais seremos escravos do homem branco.

Ele se admirava, mas não se esquecia do mingau. Batia a colher no fundo da gamela, com a voracidade de quem não comia há vários dias.

— Andou muito até chegar? — tornou a velha.

Ele ficou olhando no vazio, como se contasse mentalmente.

— Sete noites, correndo. Sete dias escondido no mato.

Terminado o bobó, o negro sobressaltou-se, como se lembrasse de um esquecido importante.

— Preciso trabalhar!

A velha fechou-lhe o caminho com o corpanzil, obrigando-o a recuar até o jirau de varas onde estivera sentado.

— Não tenha pressa. Descanse que você andou muito.

O negro estranhava, a mulher prosseguiu.

— O que havia pra plantar, está plantado.

— O capitão...

— Há homens preparados. Não será desta vez ainda que cairá o Palmar.

O homem permaneceu sentado na beira do jirau, indeciso entre atender as ordens da mulher e reagir conforme sua consciência.

— Deite e descanse. Depois, os homens da guerra dirão o que você deve fazer.

Diante da insistência, acatou. Estendeu-se na cama de varas e dormiu em seguida.

Dandinha ficou observando o sono sobressaltado do desconhecido. Reparou um tanto nas palavras em dialeto que ele repetia. Sem entender, retirou-se.

22

O capitão Mata-Mata

Apesar de não haver jamais pisado os domínios do rei Zambi, a novidade sobre o iminente ataque promoveu desassossego incomum nos Palmares porque o tal capitão fazia-se preceder de fama terrível, onde avultavam os relatos de atrocidades.

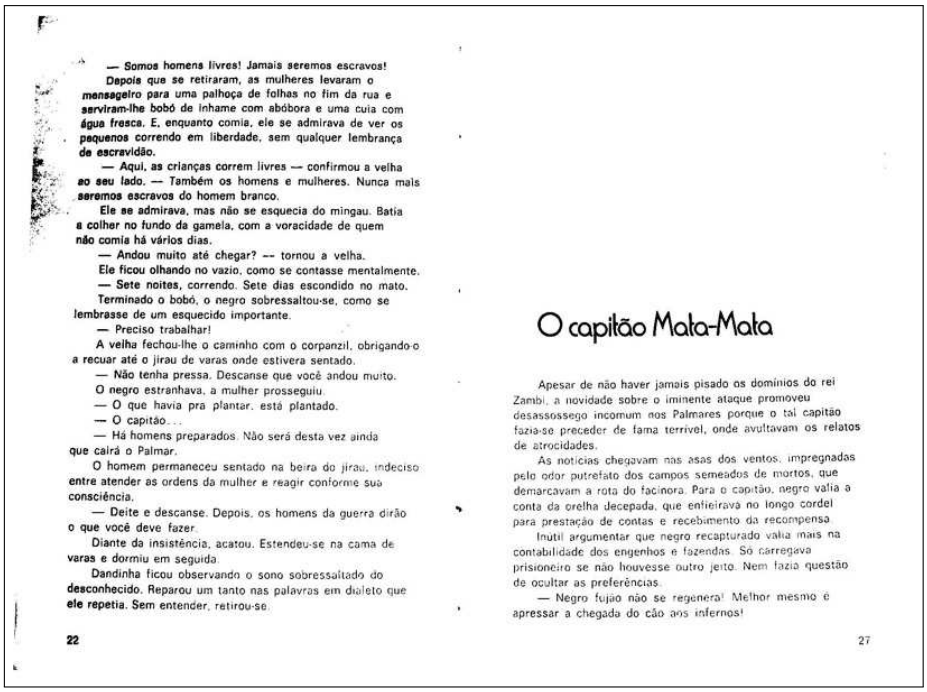
As notícias chegavam nas asas dos ventos, impregnadas pelo odor putrefato dos campos semeados de mortos, que demarcavam a rota do facínora. Para o capitão, negro valia a conta da orelha decepada, que enfieirava no longo cordel para prestação de contas e recebimento da recompensa.

Inútil argumentar que negro recapturado valia mais na contabilidade dos engenhos e fazendas. Só carregava prisioneiro se não houvesse outro jeito. Nem fazia questão de ocultar as preferências.

— Negro fujão não se regenera! Melhor mesmo é apressar a chegada do cão aos infernos!

27

Figura 5.7: Documento da Figura 5.1 binarizado pelo algoritmo Otsu globalmente.



— Somos homens livres! Jamais seremos escravos!

Depois que se retiraram, as mulheres levaram o mensageiro para uma palhoça de folhas no fim da rua e serviram-lhe bobó de inhame com abóbora e uma cuia com água fresca. E, enquanto comia, ele se admirava de ver os pequenos correndo em liberdade, sem qualquer lembrança de escravidão.

— Aqui, as crianças correm livres — confirmou a velha ao seu lado. — Também os homens e mulheres. Nunca mais seremos escravos do homem branco.

Ele se admirava, mas não se esquecia do mingau. Batia a colher no fundo da gamela, com a voracidade de quem não comia há vários dias.

— Andou muito até chegar? — tornou a velha.

Ele ficou olhando no vazio, como se contasse mentalmente.

— Sete noites, correndo. Sete dias escondido no mato.

Terminado o bobó, o negro sobressaltou-se, como se lembrasse de um esquecido importante.

— Preciso trabalhar!

A velha fechou-lhe o caminho com o corpanzil, obrigando-o a recuar até o jirau de varas onde estivera sentado.

— Não tenha pressa. Descanse que você andou muito.

O negro estranhava, a mulher prosseguiu.

— O que havia pra plantar, está plantado.

— O capitão...

— Há homens preparados. Não será desta vez ainda que cairá o Palmar.

O homem permaneceu sentado na beira do jirau, indeciso entre atender as ordens da mulher e reagir conforme sua consciência.

— Deite e descanse. Depois, os homens da guerra dirão o que você deve fazer.

Diante da insistência, acatou. Estendeu-se na cama de varas e dormiu em seguida.

Dandinha ficou observando o sono sobressaltado do desconhecido. Reparou um tanto nas palavras em dialeto que ele repetia. Sem entender, retirou-se.

22

O capitão Mata-Mata

Apesar de não haver jamais pisado os domínios do rei Zambi, a novidade sobre o iminente ataque promoveu desassossego incomum nos Palmares porque o tal capitão fazia-se preceder de fama terrível, onde avultavam os relatos de atrocidades.

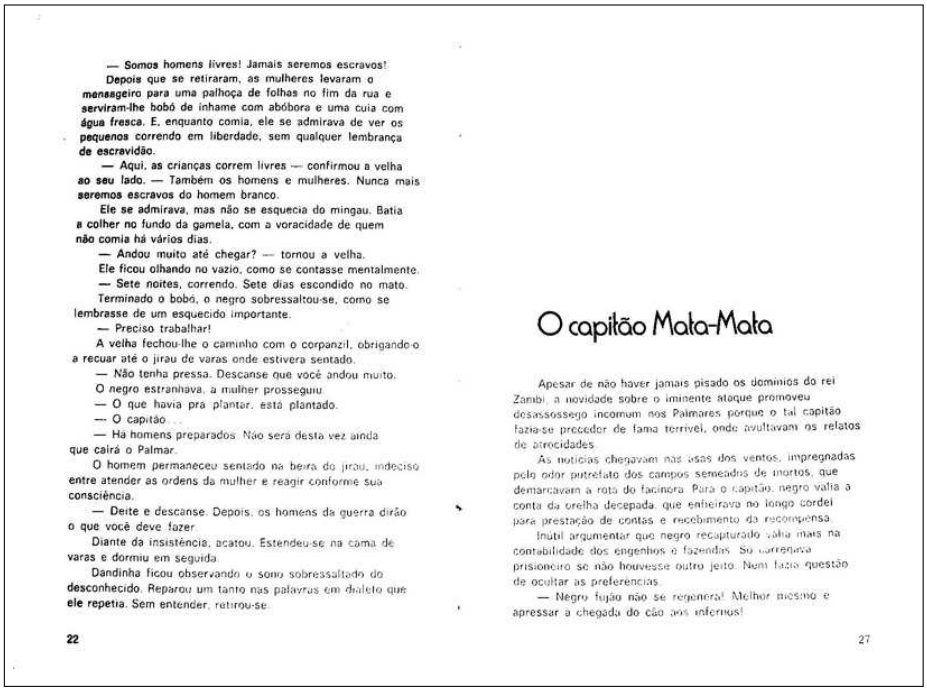
As notícias chegavam nas asas dos ventos, impregnadas pelo odor putrefato dos campos semeados de mortos, que demarcavam a rota do facínora. Para o capitão, negro valia a conta da orelha decepada, que enfieirava no longo cordel para prestação de contas e recebimento da recompensa.

Inútil argumentar que negro recapturado valia mais na contabilidade dos engenhos e fazendas. Só carregava prisioneiro se não houvesse outro jeito. Nem fazia questão de ocultar as preferências.

— Negro fujão não se regenera! Melhor mesmo é apressar a chegada do cão aos infernos!

27

Figura 5.8: Documento da Figura 5.1 binarizado pelo algoritmo Yen-Chang-Chang globalmente.



— Somos homens livres! Jamais seremos escravos!

Depois que se retiraram, as mulheres levaram o mensageiro para uma palhoça de folhas no fim da rua e serviram-lhe bobó de inhame com abóbora e uma cuia com água fresca. E, enquanto comia, ele se admirava de ver os pequenos correndo em liberdade, sem qualquer lembrança de escravidão.

— Aqui, as crianças correm livres — confirmou a velha ao seu lado. — Também os homens e mulheres. Nunca mais seremos escravos do homem branco.

Ele se admirava, mas não se esquecia do mingau. Batia a colher no fundo da gamela, com a voracidade de quem não comia há vários dias.

— Andou muito até chegar? — tornou a velha.

Ele ficou olhando no vazio, como se contasse mentalmente.

— Sete noites, correndo. Sete dias escondido no mato.

Terminado o bobó, o negro sobressaltou-se, como se lembrasse de um esquecido importante.

— Preciso trabalhar!

A velha fechou-lhe o caminho com o corpanzil, obrigando-o a recuar até o jirau de varas onde estivera sentado.

— Não tenha pressa. Descanse que você andou muito.

O negro estranhava, a mulher prosseguiu.

— O que havia pra plantar, está plantado.

— O capitão...

— Há homens preparados. Não será desta vez ainda que cairá o Palmar.

O homem permaneceu sentado na beira do jirau, indeciso entre atender as ordens da mulher e reagir conforme sua consciência.

— Deite e descanse. Depois, os homens da guerra dirão o que você deve fazer.

Diante da insistência, acatou. Estendeu-se na cama de varas e dormiu em seguida.

Dandinha ficou observando o sono sobressaltado do desconhecido. Reparou um tanto nas palavras em dialeto que ele repetia. Sem entender, retirou-se.

O capitão Mata-Mata

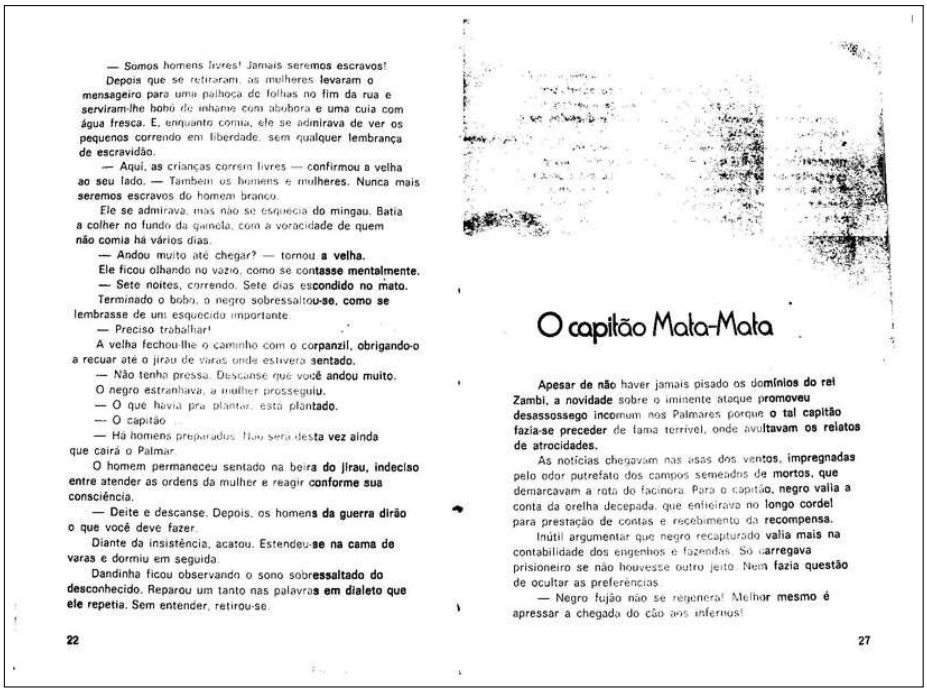
Apesar de não haver jamais pisado os domínios do rei Zambi, a novidade sobre o iminente ataque promoveu desassossego incomum nos Palmares porque o tal capitão fazia-se preceder de fama terrível, onde avultavam os relatos de atrocidades.

As notícias chegavam nas asas dos ventos, impregnadas pelo odor putrefato dos campos semeados de mortos, que demarcavam a rota do facínora. Para o capitão, negro valia a conta da orelha decepada, que enfileirava no longo cordel para prestação de contas e recebimento da recompensa.

Inútil argumentar que negro recapturado valia mais na contabilidade dos engenhos e fazendas. Só carregava prisioneiro se não houvesse outro jeito. Nem fazia questão de ocultar as preferências.

— Negro fujão não se regenera! Melhor mesmo é apressar a chegada do cão aos infernos!

Figura 5.9: Documento da Figura 5.1 binarizado pelo algoritmo Johannsen-Bille globalmente.



— Somos homens livres! Jamais seremos escravos!

Depois que se retiraram, as mulheres levaram o mensageiro para uma palhoça de folhas no fim da rua e serviram-lhe bobó de inhame com abóbora e uma cuia com água fresca. E, enquanto comia, ele se admirava de ver os pequenos correndo em liberdade, sem qualquer lembrança de escravidão.

— Aqui, as crianças correm livres — confirmou a velha ao seu lado. — Também os homens e mulheres. Nunca mais seremos escravos do homem branco.

Ele se admirava, mas não se esquecia do mingau. Batia a colher no fundo da gamela, com a voracidade de quem não comia há vários dias.

— Andou muito até chegar? — tornou a velha.

Ele ficou olhando no vazio, como se contasse mentalmente.

— Sete noites, correndo. Sete dias escondido no mato.

Terminado o bobó, o negro sobressaltou-se, como se lembrasse de um esquecido importante.

— Preciso trabalhar!

A velha fechou-lhe o caminho com o corpanzil, obrigando-o a recuar até o jirau de varas onde estivera sentado.

— Não tenha pressa. Descanse que você andou muito.

O negro estranhava, a mulher prosseguiu.

— O que havia pra plantar, está plantado.

— O capitão...

— Há homens preparados. Não será desta vez ainda que cairá o Palmar.

O homem permaneceu sentado na beira do jirau, indeciso entre atender as ordens da mulher e reagir conforme sua consciência.

— Deite e descanse. Depois, os homens da guerra dirão o que você deve fazer.

Diante da insistência, acatou. Estendeu-se na cama de varas e dormiu em seguida.

Dandinha ficou observando o sono sobressaltado do desconhecido. Reparou um tanto nas palavras em dialeto que ele repetia. Sem entender, retirou-se.

O capitão Mata-Mata

Apesar de não haver jamais pisado os domínios do rei Zambi, a novidade sobre o iminente ataque promoveu desassossego incomum nos Palmares porque o tal capitão fazia-se preceder de fama terrível, onde avultavam os relatos de atrocidades.

As notícias chegavam nas asas dos ventos, impregnadas pelo odor putrefato dos campos semeados de mortos, que demarcavam a rota do facínora. Para o capitão, negro valia a conta da orelha decepada, que enfileirava no longo cordel para prestação de contas e recebimento da recompensa.

Inútil argumentar que negro recapturado valia mais na contabilidade dos engenhos e fazendas. Só carregava prisioneiro se não houvesse outro jeito. Nem fazia questão de ocultar as preferências.

— Negro fujão não se regenera! Melhor mesmo é apressar a chegada do cão aos infernos!

Figura 5.10: Documento da Figura 5.1 binarizado pelo algoritmo daSilva-Lins-Rocha em 18 regiões.

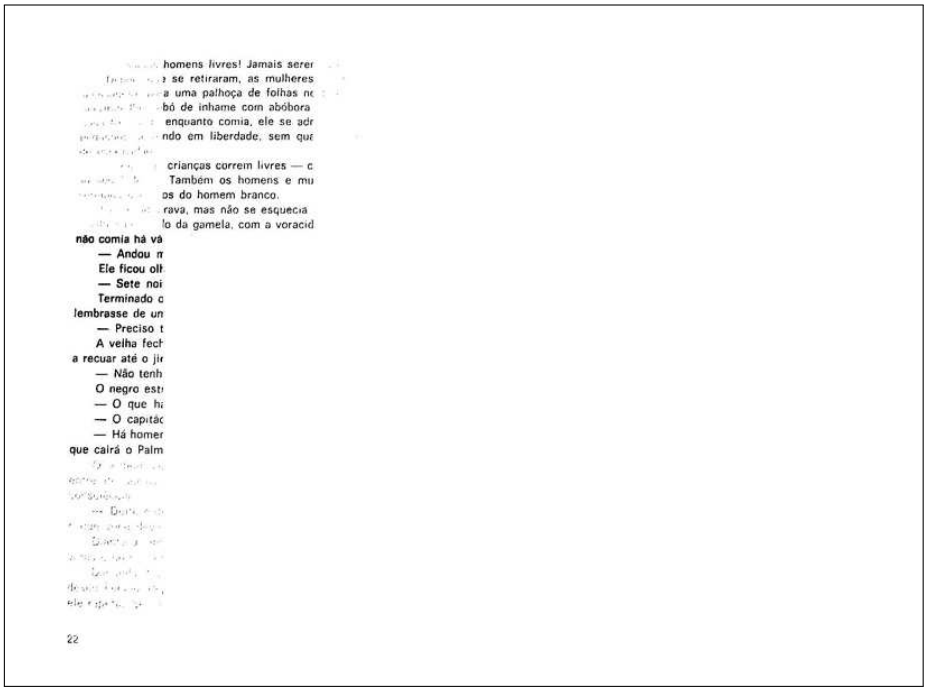


Figura 5.11: Documento da Figura 5.1 binarizado pelo algoritmo Mello-Lins em 18 regiões.

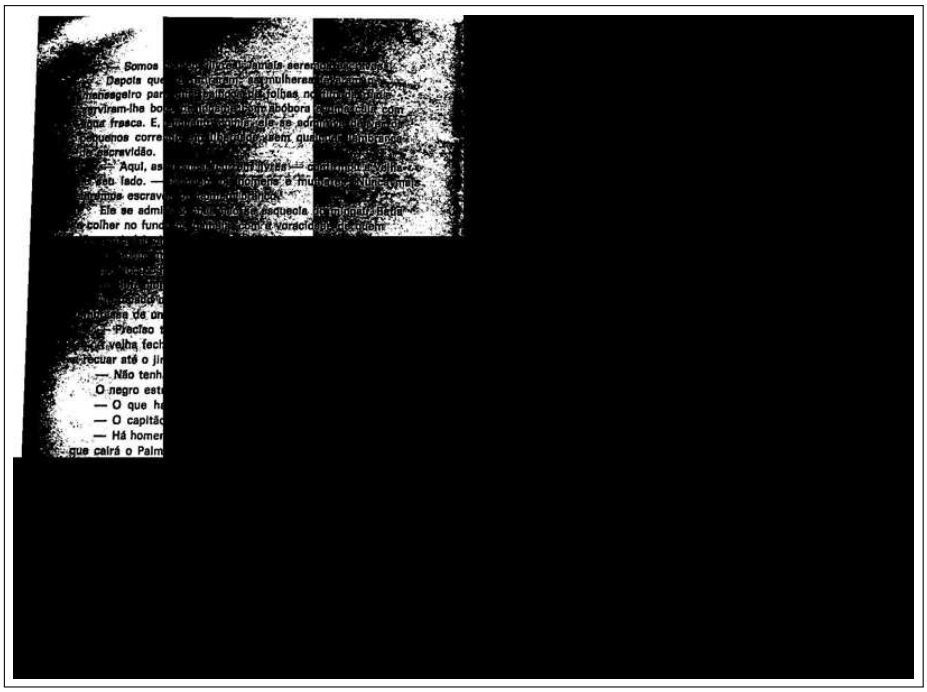
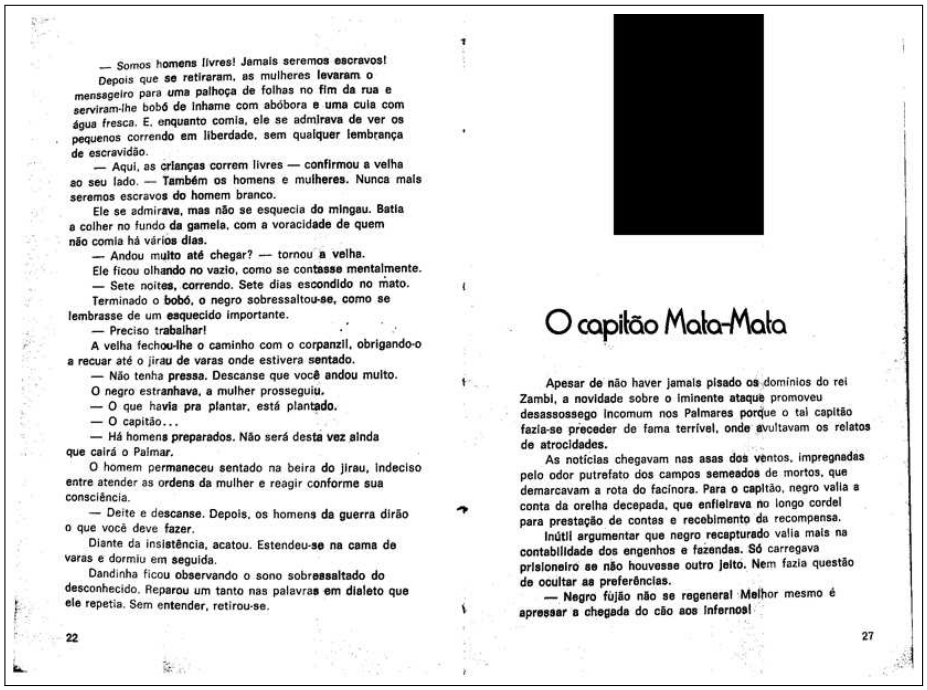


Figura 5.12: Documento da Figura 5.1 binarizado pelo algoritmo Pun em 18 regiões.



— Somos homens livres! Jamais seremos escravos!

Depois que se retiraram, as mulheres levaram o mensageiro para uma palhoça de folhas no fim da rua e serviram-lhe bobó de inhame com abóbora e uma cuia com água fresca. E, enquanto comia, ele se admirava de ver os pequenos correndo em liberdade, sem qualquer lembrança de escravidão.

— Aqui, as crianças correm livres — confirmou a velha ao seu lado. — Também os homens e mulheres. Nunca mais seremos escravos do homem branco.

Ele se admirava, mas não se esquecia do mingau. Batia a colher no fundo da gamela, com a voracidade de quem não comia há vários dias.

— Andou muito até chegar? — tornou a velha.

Ele ficou olhando no vazio, como se contasse mentalmente.

— Sete noites, correndo. Sete dias escondido no mato.

Terminado o bobó, o negro sobressaltou-se, como se lembrasse de um esquecido importante.

— Preciso trabalhar!

A velha fechou-lhe o caminho com o corpanzil, obrigando-o a recuar até o jirau de varas onde estivera sentado.

— Não tenha pressa. Descanse que você andou muito.

O negro estranhava, a mulher prosseguiu.

— O que havia pra plantar, está plantado.

— O capitão...

— Há homens preparados. Não será desta vez ainda que cairá o Palmar.

O homem permaneceu sentado na beira do jirau, indeciso entre atender as ordens da mulher e reagir conforme sua consciência.

— Deite e descanse. Depois, os homens da guerra dirão o que você deve fazer.

Diante da insistência, acatou. Estendeu-se na cama de varas e dormiu em seguida.

Dandinha ficou observando o sono sobressaltado do desconhecido. Reparou um tanto nas palavras em dialeto que ele repetia. Sem entender, retirou-se.

22

O capitão Mata-Mata

Apesar de não haver jamais pisado os domínios do rei Zambi, a novidade sobre o iminente ataque promoveu desassossego incomum nos Palmares porque o tal capitão fazia-se preceder de fama terrível, onde avultavam os relatos de atrocidades.

As notícias chegavam nas asas dos ventos, impregnadas pelo odor putrefato dos campos semeados de mortos, que demarcavam a rota do facínora. Para o capitão, negro valia a conta da orelha decepada, que enfileirava no longo cordel para prestação de contas e recebimento da recompensa.

Inútil argumentar que negro recapturado valia mais na contabilidade dos engenhos e fazendas. Só carregava prisioneiro se não houvesse outro jeito. Nem fazia questão de ocultar as preferências.

— Negro fujão não se regenera! Melhor mesmo é apressar a chegada do cão aos infernos!

27

Figura 5.13: Documento da Figura 5.1 binarizado pelo algoritmo Kapur-Sahoo-Wong em 18 regiões.

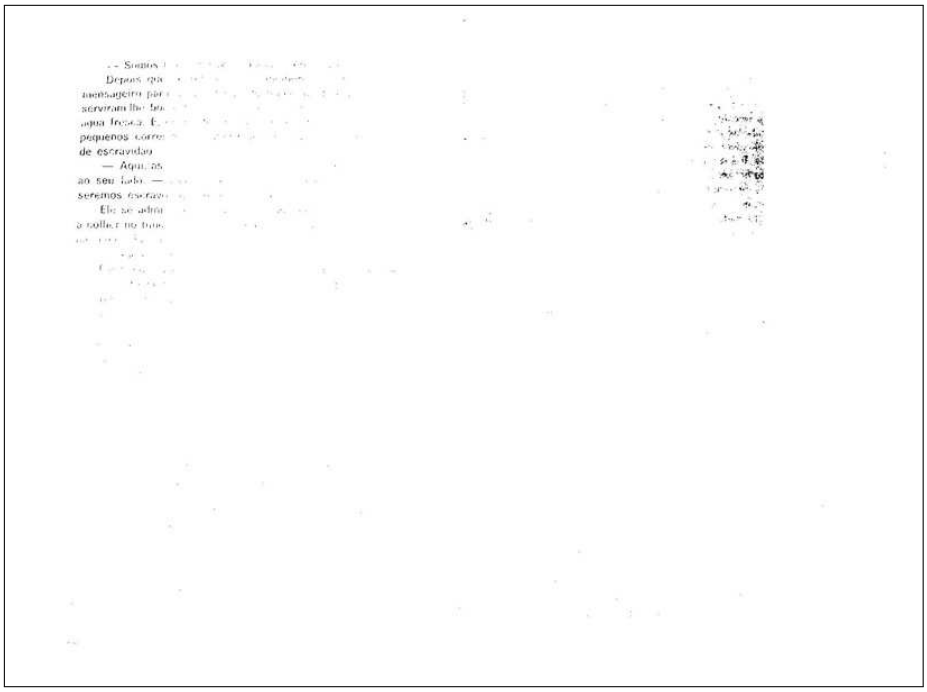


Figura 5.14: Documento da Figura 5.1 binarizado pelo algoritmo Wulu em 18 regiões.

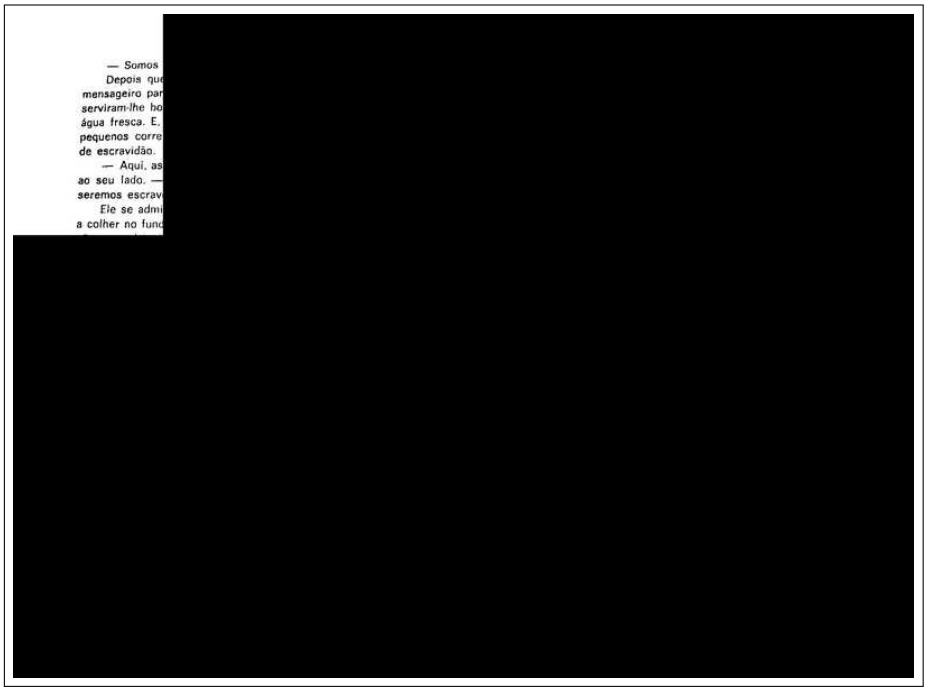


Figura 5.15: Documento da Figura 5.1 binarizado pelo algoritmo Otsu em 18 regiões.

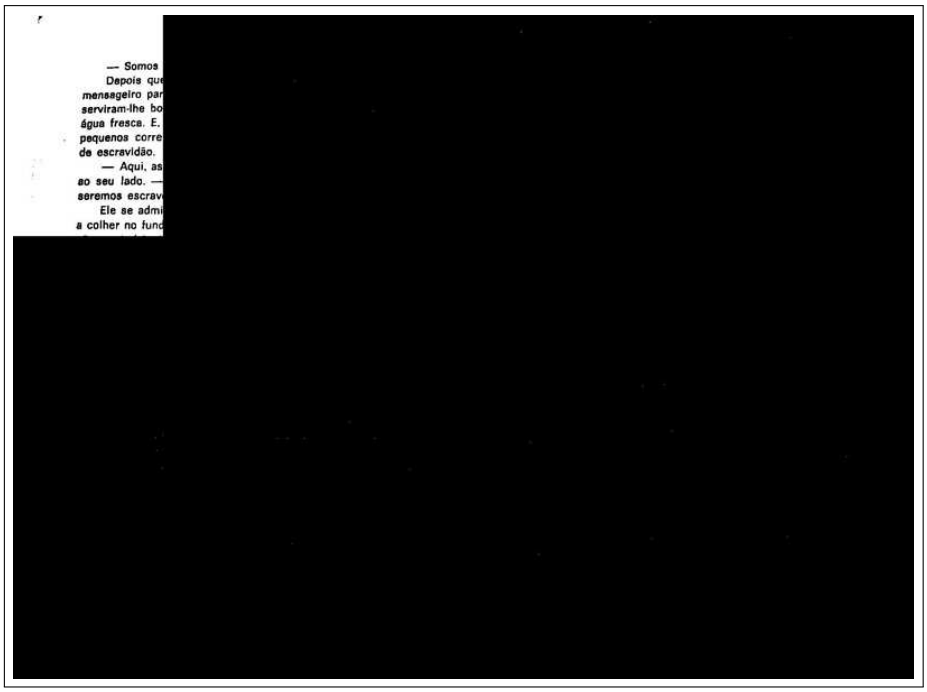


Figura 5.16: Documento da Figura 5.1 binarizado pelo algoritmo Yen-Chang-Chang em 18 regiões.

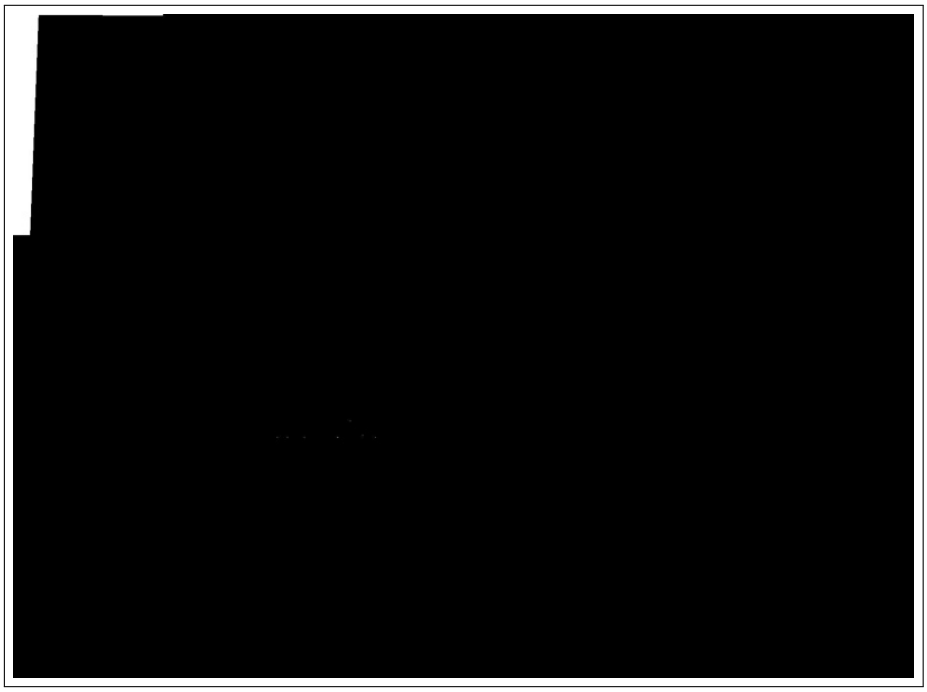


Figura 5.17: Documento da Figura 5.1 binarizado pelo algoritmo Johannsen-Bille em 18 regiões.

capítulo 6

Correção de distorções de perspectiva

Sempre que o objeto fotografado não estiver paralelo ao plano da objetiva da câmera haverá distorção de perspectiva na imagem gerada. O uso de câmeras digitais portáteis para a digitalização de documentos em geral não utiliza suporte mecânico para possibilitar um alinhamento entre o documento e a câmera, dessa maneira é freqüente a distorção de perspectiva na imagem. A distorção causada por perspectiva gera considerável grau de dificuldade na análise de documentos, incluindo a remoção de bordas, devido ao fato de que o documento deixa de ser um retângulo, e o OCR, visto que as fontes podem se apresentar distorcidas ou em tamanho inferior ao aceitável. No presente capítulo visa-se estudar e corrigir a distorção de perspectiva em documentos fotografados com câmeras digitais portáteis.

Antes da aplicação de algoritmos de correção de perspectiva, faz-se necessário o conhecimento do processo de captura de imagens, descrito na próxima seção.

6.1 A formação de imagens

A formação de imagens pode ser assumida como a projeção de três dimensões para duas dimensões. O ponto essencial é a perda de uma coordenada, o que constitui considerável perda de informação. A análise de imagens bidimensionais, assim como a reconstrução de imagens tridimensionais a partir de imagens bidimensionais para posterior análise, são tarefas de grande complexidade.

A posição de objetos pode ser descrita como ilustrado na Figura 6.1, em que as coorde-

nadas relacionadas à cena observada são chamadas de coordenadas no mundo e denotadas por $X = (X_1, X_2, X_3)$. Um segundo sistema de coordenadas, as coordenadas da câmera $X' = (X'_1, X'_2, X'_3)$, é definido como a posição da câmera que capta a cena, em que o eixo X'_3 é alinhado com o eixo óptico do sistema da câmera.

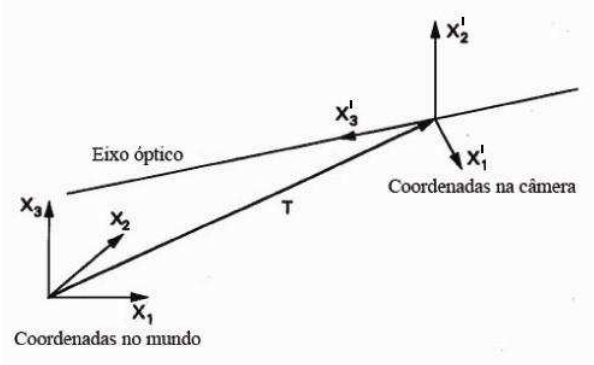


Figura 6.1: Ilustração das coordenadas do mundo e da câmera.

6.1.1 O modelo de câmera pinhole

Uma vez conhecidas as coordenadas da câmera, pode-se então estudar o sistema óptico dessa. Embora existam diferentes modelos de câmera, o modelo pinhole é popular devido à sua tratabilidade matemática, tido como o mais simples possível [17]. O elemento de formação de imagem dessa câmera é um buraco infinitesimalmente pequeno. Apenas os raios de luz vindos de um objeto em (X_1, X_2, X_3) e que passam através do buraco atingem o plano da imagem em (x_1, x_2, d_i) (Figura 6.2).

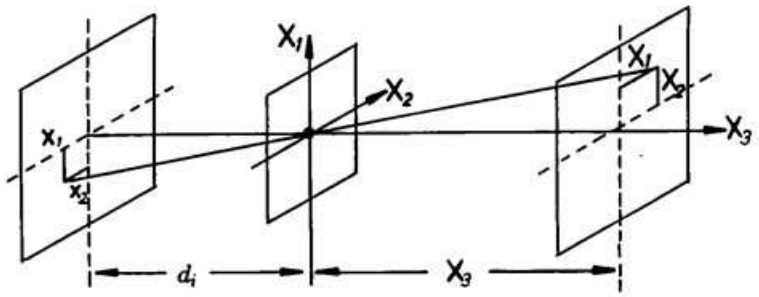


Figura 6.2: Formação de imagens com uma câmera pinhole.

A formação de imagens com uma câmera pinhole é essencialmente uma projeção de perspectiva, então esta formação de imagens é semelhante à formação de imagens com raios

penetrantes, como raios X (vide Figura 6.3), cuja equação de projeção corresponde a:

$$(X_1, X_2, X_3) \rightarrow (x_1, x_2) = (\frac{d_i X_1}{X_3}, \frac{d_i X_2}{X_3}),$$

em que $X = (X_1, X_2, X_3)$ representa um ponto no mundo, $x = (x_1, x_2)$ representa X na imagem, e d_i representa a distância entre o ponto no mundo e sua representação [17].

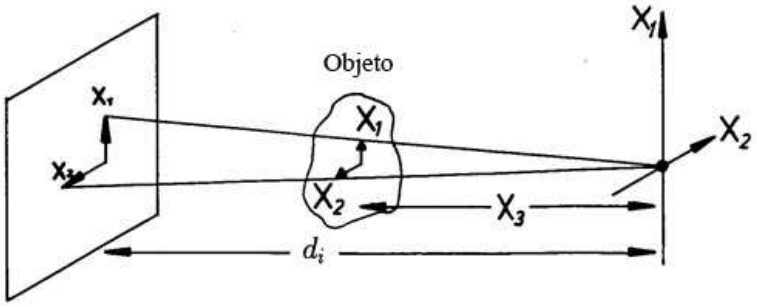


Figura 6.3: Projeção de perspectiva com raios-X.

Quando coordenadas generalizadas são utilizadas, as coordenadas da imagem são divididas pela distância d_i:

$$\frac{x_1}{d_i} \rightarrow x_1, \frac{x_2}{d_i} \rightarrow x_2.$$

A equação geral de projeção de perspectiva reduz a:

$$(X_1, X_2, X_3) \rightarrow (x_1, x_2) = (\frac{X_1}{X_3}, \frac{X_2}{X_3}).$$

6.1.2 Coordenadas homogêneas

Coordenadas homogêneas são definidas como um vetor de quatro componentes. Considere-se $X = (tX_1, tX_2, tX_3, t)$, através do qual as tradicionais coordenadas tridimensionais são obtidas dividindo-se os três primeiros componentes pelo quarto. Transformações como translação, rotação por qualquer um dos três eixos, redimensionamento e projeção de perspectiva podem ser obtidas multiplicando-se X por uma matriz M de dimensão 4x4 [17]:

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -T_1 & -T_2 & -T_3 & 1 \end{bmatrix}$$ Translação por $(-T_1, -T_2, -T_3)$

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ Rotação em torno do eixo x por θ

$$R_y = \begin{bmatrix} \cos\psi & 0 & \sin\psi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\psi & 0 & \cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ Rotação em torno do eixo y por ψ

$$R_z = \begin{bmatrix} \cos\phi & \sin\phi & 0 & 0 \\ -\sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ Rotação em torno do eixo z por ϕ

$$E = \begin{bmatrix} s_1 & 0 & 0 & 0 \\ 0 & s_2 & 0 & 0 \\ 0 & 0 & s_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ Mudança de escala

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1/d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ Projeção de perspectiva

Transformações completas de pontos no mundo para coordenadas em imagens podem ser compostas de matrizes elementares, visto que multiplicação de matrizes é associativa. A matriz M pode seguir a seguinte decomposição:

$$M = TR_xR_yR_zPEC,$$

em que as matrizes T, R_x, R_y, R_z, P, S e C correspondem a translação, rotação nos eixos x, y e z, projeção de perspectiva, redimensionamento e recorte, respectivamente.

A formação de imagem utilizando o modelo de câmera pinhole e aplicando-se coordenadas generalizadas é dada por $p = MP$, em que p é o ponto na imagem, P é o ponto no mundo real e M é a matriz da câmera composta por parâmetros internos da câmera e externos, como a posição da câmera. Os pontos p e P são representados em coordenadas homogêneas de dimensões 3x1 e 4x1 respectivamente. A matriz M é uma matriz de dimensão 3x4. O referido modelo preserva a incidência de linhas e induz transformações lineares no espaço projetivo.

6.2 Transformação de perspectiva

Quando objetos planares, como é o caso de documentos, são capturados, as imagens observadas de diferentes posições são relacionadas por uma transformação projetiva linear, referida como Homógrafo.

$$x'_i = Hx_i.$$

x'_i e x_i são vetores 3x1 e poderiam corresponder a imagens de um mesmo ponto. A matriz H é uma matriz 3x3, definida por um fator de escala e oito graus de liberdade. Dados quatro pontos correspondentes entre duas imagens, H pode ser unicamente calculado [14].

Correção de perspectiva pode ser assumida como a recuperação da visão frontal de uma imagem através da obtenção do homógrafo a partir de uma visão arbitrária. Pontos correspondentes nas duas imagens são relacionados por uma transformação linear no espaço projetivo. Se $x' = [x', y']^T$ e $x = [x, y]^T$, são os pontos correspondentes em duas imagens relacionadas por um homógrafo, então:

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}},$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}.$$

Visto que o homógrafo possui oito variáveis é necessário o número mínimo de oito equações para calculá-lo. Tais variáveis podem ser calculadas a partir de quatro pontos correspondentes entre duas imagens. Um homógrafo projetivo pode ser compreendido como o produto de três

componentes: similaridade, afim e projetiva. Ao se buscar corrigir a perspectiva da imagem, procura-se remover as componentes afim e projetiva.

6.3 Métodos de interpolação

A interpolação em imagens pode ser definida como um método de construir novos pixels a partir de um conjunto de pixels conhecidos. Operações envolvendo transformações de perspectiva ou geométricas em imagens requerem a aplicação de algum método de interpolação. Os principais métodos de interpolação aplicados a imagens são descritos nas subseções que seguem.

6.3.1 Interpolação pela vizinhança mais próxima

A interpolação pela vizinhança mais próxima é também conhecida como interpolação de ordem zero, sendo o método mais rápido de interpolação. Entretanto, a aplicação deste método de interpolação pode causar erros de sobreposição de freqüências ou distorções conhecidas como *jaggies*, que são distorções nos quais os contornos dos objetos aparecem serrilhados ou imprecisos.

Interpolação de ordem zero simplesmente determina o ponto D na imagem destino com o valor do pixel mais próximo a esta região correspondente na imagem fonte.

6.3.2 Interpolação linear

O método de interpolação linear é aplicado em uma dimensão, determinando-se um ponto baseando-se em outros dois. Conhecendo-se as coordenadas $P_0 = (x_0, y_0)$ e $P_1 = (x_1, y_1)$, ilustradas na Figura 6.4, deseja-se determinar os pontos na linha formada por P_0 e P_1 com um dado x no intervalo $[x_0, x_1]$. Fazendo:

$$\alpha = \frac{y - y_0}{y_1 - y_0} = \frac{x - x_0}{x_1 - x_0},$$

em que α é denominado coeficiente de interpolação. Como o valor de x já é conhecido, pode-se determinar α. Manipulando-se matematicamente a equação, têm-se:

$$y = (1 - \alpha)y_0 + \alpha y_1,$$

através do qual é possível o cálculo de y diretamente. Como a aplicação é em uma dimensão, determina-se que a intensidade da cor c do pixel p é representada pelo y acima descrito, enquanto x denota a posição de p na linha (ou coluna) da imagem.

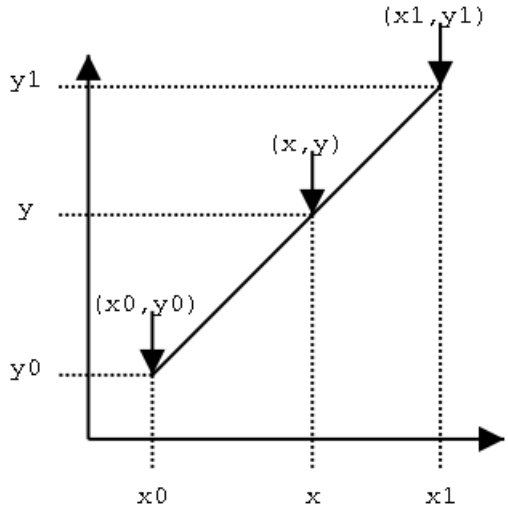


Figura 6.4: Interpolação Linear.

6.3.3 Interpolação bilinear

Interpolação bilinear é uma extensão da interpolação linear para aplicação em funções de duas variáveis. A idéia principal é executar a interpolação linear em uma direção e depois em outra ortogonal à primeira.

Supondo que deseja-se encontrar o valor de uma função desconhecida f no ponto $P = (x, y)$. Assume-se que são conhecidos quatro valores de $f, Q_{11} = (x_1, y_1), Q_{12} = (x_1, y_2), Q_{21} = (x_2, y_1), Q_{22} = (x_2, y_2)$, ilustrados na Figura 6.5.

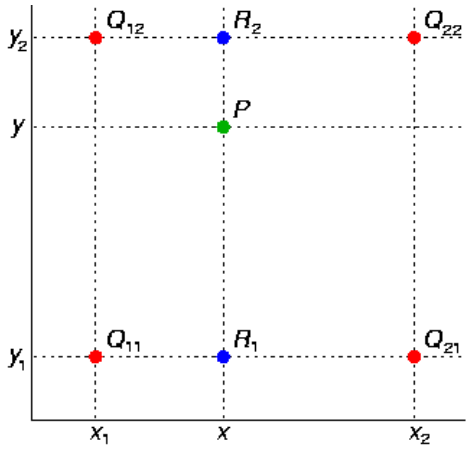


Figura 6.5: Interpolação Bilinear.

Primeiro executa-se interpolação linear na direção x:

$$f(R_1) = \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}),$$

$$f(R_2) = \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}).$$

Em que $R_1 = (x, y_1)$ e $R_2 = (x, y_2)$. Em seguida aplica-se a interpolação na direção y:

$$f(P) = \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2)$$

Ao contrário do que o nome sugere, observa-se que a interpolação bilinear não é linear.

6.3.4 Interpolação bicúbica

A interpolação bicúbica preserva detalhes presentes na imagem ao custo de tempo adicional para a execução da interpolação. Neste método, o valor $f(x, y)$ de uma função f no ponto (x, y) é calculado como uma média ponderada dos dezesseis pontos mais próximos a ele, montando uma matriz 4x4. Dois polinômios cúbicos de interpolação são utilizados, um para cada direção [22].

A interpolação bicúbica é calculada da seguinte forma:

$$f(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j$$

O procedimento utilizado para encontrar os coeficientes a_{ij} depende das propriedades dos dados de origem. Supondo que se deseja encontrar o ponto $P = (j + x, k + y)$ ilustrado na Figura 6.6, as equações são dadas por:

- Interpolação segundo o eixo horizontal

$$a_{j+x,k} = \frac{1}{6}(a_{j-1,k} R_1 + a_{j,k} R_2 + a_{j+1,k} R_3 + a_{j+2,k} R_4)$$

- Interpolação segundo o eixo vertical

$$a_{j+x,k} = \frac{1}{6}(a_{j+x,k-1} R_1 + a_{j+x,k} R_2 + a_{j+x,k+1} R_3 + a_{j+x,k+2} R_4)$$

Onde os coeficientes R_1 a R_4 são:

$$R_1 = (3+x)^3 - 4(2+x)^3 + 6(1+x)^3 - 4x^3$$

$$R_2 = (2+x)^3 - 4(2+x)^3 + 6x^3$$

$$R_3 = (1+x)^3 - 4(2+x)^3$$

$$R_4 = x^3$$

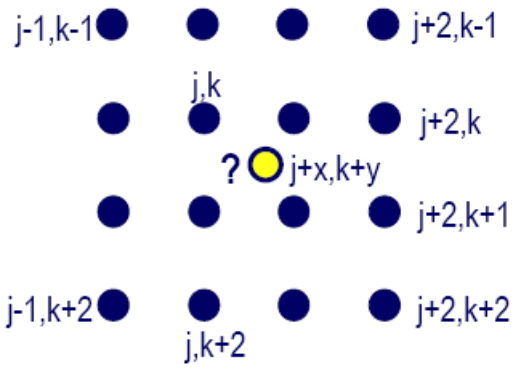


Figura 6.6: Interpolação Bicúbica.

A Figura 6.7 mostra os resultados obtidos pela rotação de uma linha vertical de cor preta em 17 graus, utilizando os algoritmos de interpolação pela vizinhança mais próxima, bilinear e bicúbica. É observado que o algoritmo de interpolação pela vizinhança mais próxima não produz níveis de cinza, gerando descontinuidades na linha. A interpolação bilinear produz linhas contínuas, enquanto a interpolação bicúbica preserva o melhor contraste da linha.

6.4 Um novo algoritmo para correção de perspectiva

Devido ao fato de que a captura dos parâmetros necessários à correção de perspectiva de imagens de documentos exige formas próprias de aquisição, pode-se encontrar pesquisas utilizando características do *layout* [40] [29] [6] [33], limites dos documentos, como contornos ou vértices [5] e características de conteúdo específico, como o tipo de texto utilizado ou utilizando-se símbolos conhecidos [16] [28].

Para a definição de um algoritmo de correção de perspectiva, optou-se pela identificação dos vértices do documento, tendo em vista sua simplicidade computacional e na aplicação deste algoritmo a documentos burocráticos, os quais tiveram suas características observadas

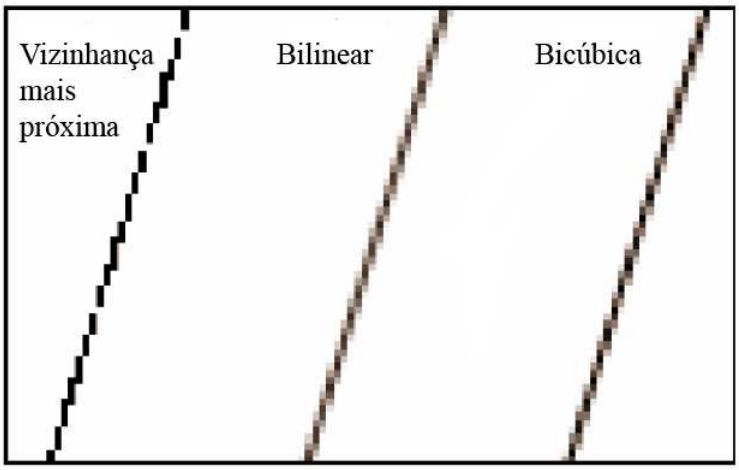


Figura 6.7: Comparação dos métodos de interpolação.

ao longo desta dissertação. Criou-se então um banco de imagens contendo 50 documentos fotografados a 3.1 Mpixels e 4.2 Mpixels (100 documentos digitalizados).

Neste algoritmo se busca identificar pontos nos contornos da imagem, para então se calcular quatro equações de retas. Com a intersecção dessas retas estimam-se os vértices dos documentos. Para se encontrar esses pontos nos contornos, há a necessidade da utilização de características dos documentos, para assim diferenciá-lo da borda. Optou-se pela identificação das cores mais freqüentes nas laterais da imagem como informação para decisão da localização do contorno.

Conforme se observa na seção 5.1.3, são necessários quatro pontos correspondentes em cada imagem, origem e destino, para que seja possível o cálculo do homógrafo. Com os vértices estimados na imagem origem, em sentido horário e partindo do canto superior esquerdo pode-se nomeá-los C_0, C_1, C_2, C_3. Calcula-se então a razão de aspecto, dada por

$$R = \frac{a+c}{b+d}$$

onde $a = |C_0 - C_1|, b = |C_1 - C_2|, c = |C_2 - C_3|, d = |C_3 - C_0|$. A imagem destino de tamanho W x H é construída de forma que $W/H = R$. Para preservação da região de resolução máxima da fonte na imagem, optou-se por preservar o valor de c na imagem destino. Assim sendo, $c' = a' = c$ e $b' = d'$, onde a',b',c' e d' são comprimentos interligando dois vértices da mesma forma que na imagem origem, definindo o retângulo com o auxílio da razão de aspecto.

Foram utilizados dois métodos de interpolação, bilinear e bicúbica, buscando prover aná-

lise do melhor desempenho para aplicação em documentos digitalizados, motivados pelas utilizações em [6] e [39].

As etapas do algoritmo para correção de perspectiva de documentos fotografados são descritas a seguir:

Etapa 1 Estimativa dos limites laterais

A estimativa dos limites laterais torna-se necessária para o cálculo das cores nas regiões próximas aos contornos. Tal estimativa ocorre da mesma forma descrita pelas etapas 1 e 2 do algoritmo apresentado no capítulo 3 desta dissertação. De forma que os pontos encontrados são (a_0=$\{x_0$,$y_0\}$, a_1=$\{x_1$,$y_1\}$, a_2=$\{x_2$,$y_2\}$ e a_3=$\{x_3$,$y_3\}$).

Etapa 2 Região de estimativa dos contornos

Para melhor precisão da estimativa dos vértices é necessário o cálculo da moda nas regiões mais distantes do centro do documento. Estas novas modas são calculadas apenas nas regiões próximas aos pontos a_1 e a_2, visto que as regiões laterais contêm os vértices e possuem maior variação de brilho. Para o cálculo desta moda utilizou-se o valor de 15 pixels em direção à região externa ao documento e 120 pixels em direção à área interna. O valor de 120 pixels é o dobro do valor estabelecido no algoritmo descrito no capítulo 3, devido à necessidade de maior precisão para a estimativa dos pontos no contorno do documento. Esses valores puderam ser aplicados tanto nas imagens com 3.2 quanto 4.1 Mpixels com bons resultados, entretanto, conforme descrito no capítulo 3, para resoluções superiores deve ser realizado cálculo para obtenção de novos valores, baseando-se na quantidade de pixels da imagem.

Etapa 3 Cálculo das equações das retas

Utilizando-se os valores adquiridos pelos cálculos das modas nas regiões laterais do documento, executa-se varredura partindo dos limites inferiores e superiores do documento - altura ou largura - em direção aos limites superiores e inferiores, respectivamente, classificando os pixels como documento ou borda. Esta varredura tem início em uma das coordenadas centrais do documento, vertical ou horizontal, e a outra coordenada no ponto máximo ou mínimo, em direção ao documento.

Por exemplo, para se encontrar os dois pontos utilizados para o cálculo da reta no contorno inferior do documento, inicia-se a varredura a partir do ponto $P_0 = (L/2, 0)$, em direção ao ponto $(L/2, A)$, onde L representa a largura e A representa a altura do documento. Ao encontrar um pixel classificado como documento, desloca-se a varredura

para a esquerda e para a direita, sendo os dois últimos pixels, um da esquerda e um da direita, classificados como documento para posterior utilização no cálculo da equação da reta (Figura 6.8). A cada pixel classificado como papel, retrocede-se cinco pixels em direção à área externa ao documento, para se evitar as possíveis irregularidades nos contornos.

Como critério de classificação, as cores constituintes de um pixel precisam estar dentro de uma tolerância de 32 níveis em relação ao valor das modas nas laterais dos documentos para a classificação como parte do documento. Este valor foi encontrado através das análises realizadas durante o desenvolvimento do algoritmo descrito no capítulo 3.

Após encontrados dois pontos pertencentes a cada lado do documento, procede-se então com a definição das equações das retas, e em seguida, o cálculo das intersecções entre elas, resultando nos vértices estimados do documento (Figura 6.9).

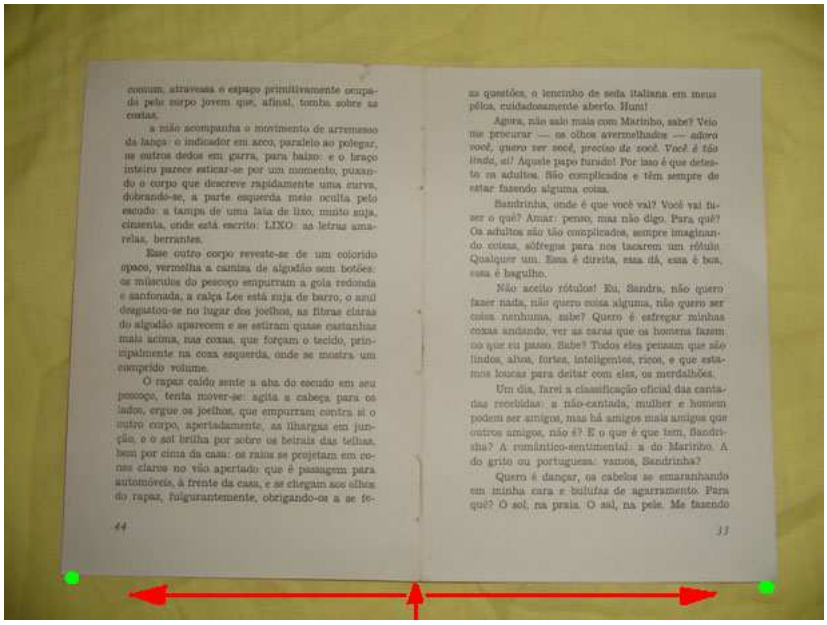


Figura 6.8: Localização dos pontos no contorno.

Etapa 4 Transformação da imagem

De posse dos vértices estimados do documento procede-se então com o cálculo da razão de aspecto para definir o tamanho da nova imagem, e com isso calcular também os vértices do documento na imagem destino.

Uma vez estimados os vértices do documento nas imagens origem e destino, é possível o cálculo do homógrafo, e posteriormente, a multiplicação deste homógrafo pela imagem

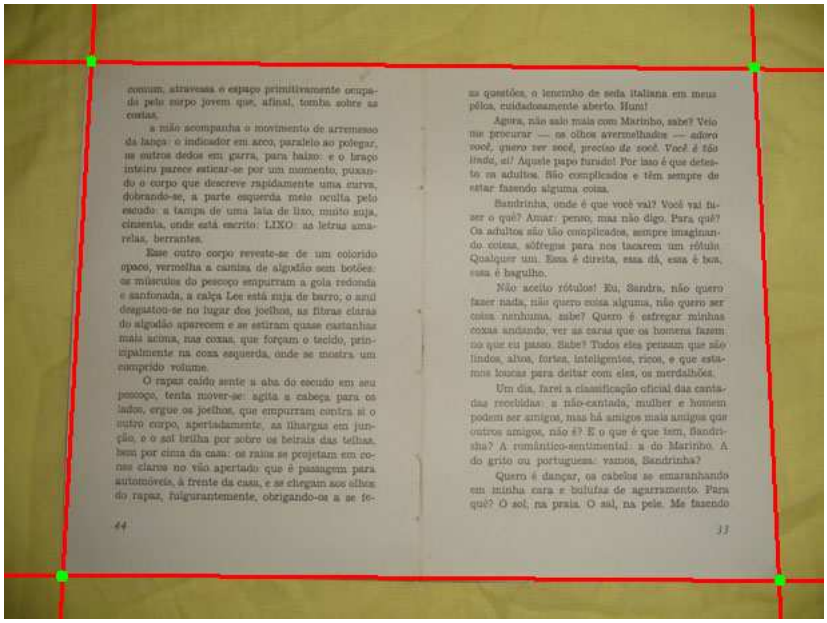


Figura 6.9: Pontos coincidentes das retas.

original, resultando na correção da perspectiva (Figura 6.10).

6.5 Os efeitos das distorções de perspectiva no OCR

Sabe-se que as distorções de perspectiva, comuns em documentos fotografados, reduzem a variação da resolução das fontes que constituem o documento, constituindo perda de qualidade. Como ferramentas de OCR constituem uma das principais aplicações dos documentos digitalizados, além de servir de referência para comparações entre documentos digitalizados por *scanners*, faz-se necessário a análise dos resultados obtidos segundo a capacidade de transcrição por uma ferramenta de OCR. Após a correção das dirtorções de perspectiva, foi realizado corte das áreas inerentes às bordas das imagens, para então ser realizada uma análise da quantidade de erros de transcrição dos documentos.

Os resultados demonstram a importância do método de interpolação utilizado na correção de perspectiva. Embora aparentemente a qualidade dos documentos digitalizados tenha melhorado, observa-se pequeno aumento na quantidade de erros, principalmente os erros de pontuação, devido à grande quantidade de substituição de vírgulas por pontos, conforme pode ser observados nas Tabelas 6.1 e 6.4. Utilizando as tabelas 6.7 à 6.14 observa-se que apesar do aumento na quantidade de erros de caracteres por substituição, sua distribuição tornou-se mais homogênea. Conforme ilustrado nas tabelas 6.2 e 6.6, nota-se que houve redução na

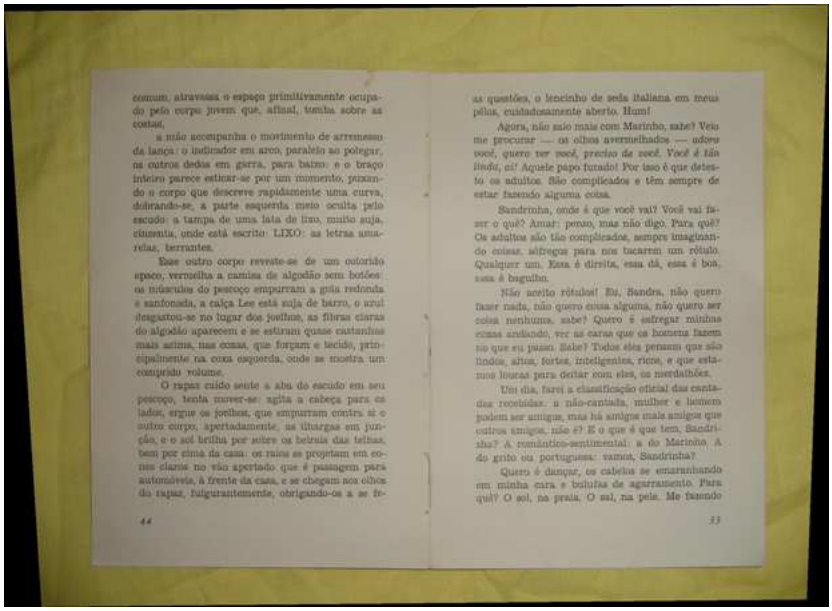


Figura 6.10: Correção de perspectiva - etapa 4.

Tabela 6.1: Erros de caracteres encontrados nas imagens utilizando interpolação bicúbica (valor absoluto).

	Corrigidos		**Sem correção**	
Tipo do erro	**3.2 Mpixels**	**4.1 Mpixels**	**3.2 Mpixels**	**4.1 Mpixels**
Substituição	147	93	104	84
Pontuação	121	42	40	12
Acentuação	76	49	83	22
Ausência	63	19	17	16
Inserção	18	14	25	19

quantidade de ausências de palavras. Ao se analisar as transcrições realizadas antes e após as correções de perspectiva, foi observado que nas transcrições dos documentos fotografados a 3.2 Mpixels as coincidências dos erros de caracteres foram equivalentes a aproximadamente 18% e 13% dos erros após correção, para os métodos de interpolação bicúbica e bilinear, respectivamente, enquanto para as imagens fotografadas a 4.1 foram aproximadamente de 19% para ambos métodos de interpolação.

Tabela 6.2: Erros de palavras encontrados nas imagens utilizando interpolação bicúbica (valor absoluto).

	Corrigidos		**Sem correção**	
Tipo do erro	**3.2 Mpixels**	**4.1 Mpixels**	**3.2 Mpixels**	**4.1 Mpixels**
Erros	146	84	106	67
Ausência	2	0	5	1

Tabela 6.3: Localização dos erros de caracteres utilizando interpolação bicúbica (em percentual).

	Corrigidos		**Sem correção**	
Regiões	**3.2 Mpixels**	**4.1 Mpixels**	**3.2 Mpixels**	**4.1 Mpixels**
Superior	46,70%	53,40%	55,66%	60,58%
Inferior	53,30%	46,60%	44,34%	39,42%
Direita	52,12%	51,16%	55,66%	50%
Esquerda	47,88%	48,84%	44,34%	50%

Tabela 6.4: Erros de caracteres encontrados nas imagens utilizando interpolação bilinear (valor absoluto).

	Corrigidos		**Sem correção**	
Tipo do erro	**3.2 Mpixels**	**4.1 Mpixels**	**3.2 Mpixels**	**4.1 Mpixels**
Substituição	151	87	104	84
Pontuação	147	49	40	12
Acentuação	86	53	83	22
Ausência	82	19	17	16
Inserção	14	13	25	19

Tabela 6.5: Erros de palavras encontrados nas imagens utilizando interpolação bilinear (valor absoluto).

	Corrigidos		**Sem correção**	
Tipo do erro	**3.2 Mpixels**	**4.1 Mpixels**	**3.2 Mpixels**	**4.1 Mpixels**
Erros de palavras	156	86	106	67
Ausência	2	0	5	1

Tabela 6.6: Localização dos erros de caracteres utilizando interpolação bilinear (em percentual).

	Corrigidos		**Sem correção**	
Regiões	**3.2 Mpixels**	**4.1 Mpixels**	**3.2 Mpixels**	**4.1 Mpixels**
Superior	42,53%	53,39%	55,66%	60,58%
Inferior	57,47%	46,61%	44,34%	39,42%
Direita	50,83%	51,16%	55,66%	50%
Esquerda	49,17%	48,84%	44,34%	50%

Tabela 6.7: Variância dos erros de caracteres utilizando interpolação bicúbica.

	Corrigidos		**Sem correção**	
Tipo do erro	**3.2 Mpixels**	**4.1 Mpixels**	**3.2 Mpixels**	**4.1 Mpixels**
Substituição	10,958	3,193	31,367	3,324
Pontuação	13,514	2,816	2,16	0,227
Acentuação	3,677	1,551	3,535	1,231
Ausência	9,347	0,604	18,557	0,467
Inserção	0,4648	0,245	0,867	0,689

Tabela 6.8: Moda dos erros de caracteres utilizando interpolação bicúbica.

	Corrigidos		**Sem correção**	
Tipo do erro	**3.2 Mpixels**	**4.1 Mpixels**	**3.2 Mpixels**	**4.1 Mpixels**
Substituição	2	0 e 1 (bimodal)	1, 2 e 3 (trimodal)	1
Pontuação	0	0	0	0
Acentuação	0	0	1 e 0 (bimodal)	0
Ausência	0	0	0	0
Inserção	0	0	0	0

Tabela 6.9: Variância dos erros de palavras utilizando interpolação bicúbica.

	Corrigidos		**Sem correção**	
Tipo do erro	**3.2 Mpixels**	**4.1 Mpixels**	**3.2 Mpixels**	**4.1 Mpixels**
Erros de palavras	9,976	3,029	5,209	2,147
Ausência	0,04	0	12,02	0,02

Tabela 6.10: Moda dos erros de palavras utilizando interpolação bicúbica.

	Corrigidos		**Sem correção**	
Tipo do erro	**3.2 Mpixels**	**4.1 Mpixels**	**3.2 Mpixels**	**4.1 Mpixels**
Erros de palavras	0	0	amodal	1
Ausência	0	0	0	0

Tabela 6.11: Variância dos erros de caracteres utilizando interpolação bilinear.

	Corrigidos		**Sem correção**	
Tipo do erro	**3.2 Mpixels**	**4.1 Mpixels**	**3.2 Mpixels**	**4.1 Mpixels**
Substituição	8,197	2,704	31,367	3,324
Pontuação	23,428	3,918	2,16	0,227
Acentuação	2,634	1,544	3,535	1,231
Ausência	14,75	0,645	18,557	0,467
Inserção	0,571	0,358	0,867	0,689

Tabela 6.12: Moda dos erros de caracteres utilizando interpolação bilinear.

	Corrigidos		**Sem correção**	
Tipo do erro	**3.2 Mpixels**	**4.1 Mpixels**	**3.2 Mpixels**	**4.1 Mpixels**
Substituição	0 e 2(bimodal)	0	1, 2 e 3 (trimodal)	1
Pontuação	0	0	0	0
Acentuação	0	0	1 e 0 (bimodal)	0
Ausência	0	0	0	0
Inserção	0	0	0	0

Tabela 6.13: Variância dos erros de palavras utilizando interpolação bilinear.

	Corrigidos		**Sem correção**	
Tipo do erro	**3.2 Mpixels**	**4.1 Mpixels**	**3.2 Mpixels**	**4.1 Mpixels**
Erros de palavras	3,777	2,756	5,209	2,147
Ausência	0,04	0	12,02	0,02

Tabela 6.14: Moda dos erros de palavras utilizando interpolação bilinear.

	Corrigidos		**Sem correção**	
Tipo do erro	**3.2 Mpixels**	**4.1 Mpixels**	**3.2 Mpixels**	**4.1 Mpixels**
Erros de palavras	amodal	0	amodal	1
Ausência	0	0	0	0

capítulo 7

Conclusões e trabalhos futuros

A praticidade e o avanço das tecnologias de armazenamento e processamento, além do barateamento das câmeras digitais, tornaram grande a disseminação do uso de câmeras fotográficas digitais portáteis.

Esta pesquisa surgiu da utilização de câmeras fotográficas digitais portáteis para a digitalização de documentos, uso esse distinto do originalmente previsto, como fotos de famílias e paisagens. Esse trabalho apresenta um estudo para identificação dos principais fatores influentes na qualidade de imagens de documentos fotografados, buscando melhoria de alguns destes fatores para facilitar a utilização de documentos fotografados. A grande variação de ambientes nos quais documentos são digitalizados impede que as técnicas tradicionais de digitalização possam ser diretamente aplicadas. O desenvolvimento de novos algoritmos para melhoria da qualidade se torna então essencial para que essa forma de aquisição torne-se possível nas mesmas aplicações que as imagens adquiridas por *scanner* ou em um número ainda maior de aplicações.

Apesar das adversidades encontradas na digitalização de documentos por câmeras fotográficas digitais, foi possível constatar que esta digitalização é uma forma viável de aquisição e disseminação de informação. Os efeitos indesejados causados no processo de digitalização de documentos burocráticos mostraram que sua influência aumenta o erro na transcrição por ferramentas de OCR, entretanto este aumento da quantidade de erros mostrou não inviabilizar essa transcrição, visto que a média de erros em caracteres nos documentos fotografados a

4.1 Mpixels, sem correção de perspectiva, foi menor do que os documentos digitalizados por *scanner* a 150 dpi.

Além da análise dos principais fatores influentes na qualidade de documentos digitalizados por câmeras fotográficas digitais, foram implementados dois algoritmos, sendo um para remoção de bordas em fotografias coloridas e o outro para correção de perspectiva.

Ainda há muito a ser desenvolvido para que se possa considerar a análise de documentos fotografados um problema resolvido. Alguns dos principais trabalhos necessários para melhoria desse ambiente incluem o desenvolvimento eficiente de algoritmos para binarização e correção de distorções geométricas causadas pela lente da câmera ou pela superfície sobre a qual o documento foi posto. Os trabalhos futuros incluem também a aplicação de filtros para ganho nas freqüências altas, que possivelmente melhoraria a transcrição por ferramentas de OCR, e possibilidade da extração automática de tolerâncias para remoção de bordas e correção de perspectiva, possivelmente baseando-se em características como entropia e distribuição do histograma, tal modificação além de facilitar a aplicação, tornaria esses algoritmos mais robustos.

Referências

[1] N. F. Alves, "Estratégias para melhoria do desempenho de ferramentas comerciais de reconhecimento óptico de caracteres," Dissertação de Mestradoem Engenharia Elétrica, Universidade Federal de Pernambuco, Recife, Brasil, 2003.

[2] H. S. Baird, H. Bunke, and K. Y. (Eds.), "Document image defect models," New York, pp. 546–556, 1992.

[3] H. Baird, "Document image defect models and their uses," in *International Conference on Document Analysis and Recognition ICDAR93*, Tsukuba, Japan, pp. 62–67, 2003.

[4] M. Cannon, J. Hochberg, and P. Kelly, "Quality assessment and restoration of typewritten document images." *IJDAR*, vol. 2, no. 2-3, pp. 80–89, 1999.

[5] P. Clark and M. Mirmehdi, "Location and recovery of text on oriented surfaces," in *SPIE conference on Document Recognition and Retrieval VII*. The International Society for Optical Engineering, January 2000, pp. 267–277. [Online]. Available: http://www.cs.bris.ac.uk/Publications/Papers/1000439.pdf

[6] ——, "On the recovery of oriented documents from single images," in *Proceedings of the 4th IEEE Advanced Concepts for Intelligent Vision Systems*, W. Philips, Ed. Ghent University, September 2002, pp. 190–197. [Online]. Available: http://www.cs.bris.ac.uk/Publications/Papers/1000664.pdf

[7] J. M. M. da Silva, R. D. Lins, and V. C. da Rocha, "Binarizing and filtering historical documents with back-to-front interference," in *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*, New York, NY, USA, pp. 853–858, ACM Press, 2006.

[8] J. M. M. da Silva, "Um novo algoritmo baseado em entropia para filtragem de interferência frente-verso," Dissertação de Mestrado em Engenharia Elétrica, Universidade Federal de Pernambuco, Recife, Brasil, 2004.

[9] C. Dance and L. Fan, "Color reconstruction in digital cameras: optimization for document images," *International Journal on Document Analysis and Recognition*, vol. 7, pp. Issue 2–3, Pages 138–146, Jul 2005.

[10] J. G. e L. Velho, *Computação Gráfica: Imagem.* Sociedade Brasileira de Matemática, 1994.

[11] A. R. G. e Silva and R. D. Lins, "Background removal of document images acquired using portable digital cameras," in *Proceedings of ICIAR 2005, Lecture Notes in Computer Science*, vol. 3656, pp. 278–285, Springer Verlag, 2005.

[12] K.-C. Fan, T.-R. Lay, and Y.-K. Wang, "Marginal noise removal of document images." in *6th International Conference on Document Analysis and Recognition (ICDAR 2001)*, Seattle, WA, USA, september 2001, pp. 317–321.

[13] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Prentice Hall, 2001.

[14] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2000.

[15] C. Hass, JPEG Quality Comparison, url: http://www.impulseadventure.com/photo/jpeg-quality.html. Visitado em 26/10/2006.

[16] L. Jagannathan and C. V. Jawahar, "Perspective correction methods for camera based document analysis," *First International Workshop on Camera-based Document Analysis and Recognition, CBDAR, in conjunction with ICDAR*, pp. 148–154, 2005.

[17] B. Jähne, *Digital Image Processing*, 3rd ed. Springer, 1995.

[18] G. Johannsen and J. Bille, "A threshold selection method using information measures," *ICPR'82: Proc. 6th Intl. Conf. Patt. Recog.*, pp. 140–143, 1982.

[19] T. Kanungo, H. Baird, and R. Haralick, "Validation and estimation of document degradation models," *Proc. Fourth Annual Symp. Document Analysis and Information Retrieval*, April 1995.

[20] T. Kanungo, R. Haralick, and I. Phillips, "Global and local document degradation models," 1993. [Online]. Available: citeseer.ist.psu.edu/kanungo93global.html

[21] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer Vision, Graphics and Image Processing*, vol. 29(3), pp. 273–285, 1985.

[22] D. Kidner, M. Dorey, and D. Smith, "What's the point? interpolation and extrapolation with a regular grid dem," in *Geocomputation 99: Proceedings of the 4th International Conference on GeoComputation*, 1999. [Online]. Available: http://www.geovista.psu.edu/sites/geocomp99/Gc99/082/gc_082.htm

[23] J. Kim, Y. Byun, and J. Choi, "Background removal of document images acquired using portable digital cameras," in *Advances in Multimedia Information Processing - PCM 2004, Lecture Notes in Computer Science*, vol. 3333. Springer, 2004, pp. 331–339.

[24] K. Larson, "The science of word recognition or how I learned to stop worrying and love the bouma," Microsoft Corporation, 2004, url: http://www.microsoft.com/typography/ctfonts/WordRecognition.aspx. Visitado em 07/06/2006.

[25] A. Leykin and F. Cutzu, "Differences of edge properties in photographs and paintings." in *International Conference on Image Processing, ICIP (3)*, pp. 541–544, 2003.

[26] J. Liang, D. Doermann, and H. Li, "Camera-based analysis of text and documents: A survey," *International Journal on Document Analysis and Recognition*, vol. 7, no. 2-3, pp. 83 – 104, July 2005.

[27] R. D. Lins and N. F. Alves, "A new technique for acessing the performance of ocrs," in *IADIS - International Conference on Computer Applications*, vol. 1, pp. 51–56, Algarve, 2005.

[28] S. Lu, B. M. Chen, and C. C. Ko, "Perspective rectification of document images using fuzzy set and morphological operations." *Image Vision Comput.*, vol. 23, no. 5, pp. 541–553, 2005.

[29] S. Lu and C. L. Tan, "The restoration of camera documents through image segmentation," in *Document Analysis Systems*, 2006, pp. 484–495.

[30] C. A. B. Mello and R. D. Lins, "Image segmentation of historical documents," *Proceedings of Visual 2000*, 2000.

[31] C. A. B. Mello and R.D.Lins, "A comparative study on OCR tools," in *Vision Interface 99*, 1999, pp. 700–704.

[32] C. A. Mello and R. D. Lins, "Generation of images of historical documents by composition," in *DocEng '02: Proceedings of the 2002 ACM symposium on Document engineering*, pp. 127–133, ACM Press, 2002.

[33] G. K. Myers, R. C. Bolles, Q.-T. Luong, J. A. Herson, and H. Aradhye, "Rectification and recognition of text in 3-d scenes," *IJDAR*, vol. 7, no. 2-3, pp. 147–158, 2005.

[34] A. C. Naiman, "The use of grayscale for improved character presentation," Ph.D. dissertation, University of Toronto, Toronto, Canada, 1991.

[35] N. Otsu, "A threshold selection method from gray level histograms," *IEEE Trans. Syst. Man Cybern. - SMC*, vol. 9(1), pp. 62–66, 1979.

[36] K. Popat, "Decoding of text lines in grayscale document images," in *Proceedings of the 2001 International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*. Salt Lake City, Utah: IEEE, May 2001.

[37] T. Pun, "Entropic thresholding, a new approach," *C. Graphics and Image Processing*, vol. 16(3), 1981.

[38] J. Reilly and F. Frey, "Recommendations for the evaluation of digital images produced from photographic, microphotographic, and various paper formats," Library of Congress National Digital Library Project, Washington, DC, 1996, url: http://www.microsoft.com/typography/ctfonts/WordRecognition.aspx. Visitado em 07/06/2006.

[39] P. L. Rosin and A. D. Marshall, "A light-weight text image processing method for handheld embedded cameras," in *Proceedings of the British Machine Vision Conference 2002, BMVC 2002, Cardiff, UK, 2-5 September 2002.* British Machine Vision Association, 2002.

[40] L. S. and C. L. Tan, "Camera document restoration for OCR," *First International Workshop on Camera-based Document Analysis and Recognition, CBDAR, in conjunction with ICDAR*, 2005.

[41] M. Seeger and C. Dance, "Binarising camera images for OCR," in *ICDAR '01: Proceedings of the Sixth International Conference on Document Analysis and Recognition*, pp. 54–59, IEEE Computer Society, 2001.

[42] C. Shannon, "A mathematical theory of communication," in *Bell System Technology Journal*, vol. 27, pp. 370–423 and 623–656, 1948.

[43] C. Thillou and B. Gosselin, "Color binarization for complex camera-based images," in *Videometrics VIII. Proceedings of the SPIE*, vol. 5667, pp. 301–308, 2004.

[44] O. D. Trier and T. Taxt, "Evaluation of binarization methods for document images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 3, pp. 312–315, 1995.

[45] B. T. Ávila and R. D. Lins, "A new algorithm for removing noisy borders from monochromatic documents," in *SAC '04: Proceedings of the 2004 ACM Symposium on Applied Computing*, pp. 1219–1225, ACM Press, 2004.

[46] L. U. Wu, M. A. Songde, and L. U. Hanqing, "An effective entropic thresholding for ultrasonic imaging," *ICPR'98: Intl. Conf. Patt. Recog.*, pp. 1522–1524, 1998.

[47] H. S. Yam and E. H. B. Smith, "Estimating degradation model parameters from character images," *Proceedings of ICDAR 2003*, vol. 02, p. 710, 2003.

[48] J.-C. Yen, F.-J. Chang, and S. Chang, "A new criterion for automatic multilevel thresholding." *IEEE Transactions on Image Processing*, vol. 4, no. 3, pp. 370–378, 1995.

[49] ACD Systems, url: http://www.acdsee.com/index.aspx. Visitado em 07/06/2006.

[50] ACM Symposium on Document Engineering 2002, url: http://www.sdml.info/doceng2002/cfp.html. Visitado em 07/06/2006.

[51] Nuance Corporate, url: http://www.nuance.com/omnipage/professional. Visitado em 07/06/2006.

APÊNDICE A

CÓDIGOS-FONTE DOS ALGORITMOS

A.1 Remoção de bordas

```
//** Programa desenvolvido em linguagem C **//

/* Este programa lê arquivos BMP sem compressão, coloca os valores
do mapa de cores em uma matriz, verifica as bordas encontradas e
copia os bytes que estão localizados entre estas bordas, referentes
ao interior do documeento.O programa verifica se os pixels adjacente
mudam repentinamente e com isso define a borda. */

#include <stdio.h>
#include <stdlib.h>

#define FUNDO 32
#define DOCUMENTO           32

// Refere-se à tolerância entre verificações de pixels do DOCUMENTO */
#define TOLERANCIA          16
// Refere-se à tolerância entre variações de mesma cor

// Inicia programa
int main(int argc, char* argv[]) {

// Declara as variáveis a serem utilizadas
char nome1[30],nome2[30]; unsigned char **BMP,R,G,B,R1,G1,B1;
int
t,i,c,j,c1,FinaldeArquivo,Altura,Largura,A,L,BMPSize,
     BordaLateralE,BordaLateralD,BordaSuperior,BordaInferior,temp,
     AlturaSize,LarguraSize,i1,i2,i3,i4,*Moda;
long int MatrixSize,M; unsigned long int biWidth,biHeight,bfSize;
```

```
FILE *arq1, *arq2;

//Iniciar as Variaveis
t=0; i=0; c=0; j=0; c1=0; FinaldeArquivo=0; Altura=0,Largura=0; A=0;
L=0; BMPSize=0; BordaLateralE=0; BordaLateralD=0; BordaSuperior=0;
BordaInferior=0; temp=0; AlturaSize=0; LarguraSize=0; i1=0; i2=0;
i3=0; i4=0;

 // Prepara o arquivo a ser analisado e o arquivo a ser gerado
    printf("Escreva o nome do arquivo origem: ");
    scanf("%s",&nome1);
    printf("Escreva o nome do arquivo destino: ");
    scanf("%s",&nome2);

    if ((arq1=fopen(nome1,"rb"))==NULL)
    {
        printf("Arquivo %s nao pode ser aberto\n",nome1);
        return 0;
    }
    if ((arq2=fopen(nome2,"wb"))==NULL)
    {
        printf("Arquivo %s nao pode ser aberto ou criado\n",nome2);
        return 0;
    }

    //Ignora os 18 primeiros bytes, pois é o cabecalho a ser repetido no arquido destino

    while (temp<18) {
        c=fgetc(arq1);
        temp++;
    };

    //Ignora os 12 primeiros bytes, pois é o cabecalho a ser repetido no arquido destino

    while (temp<12) {
        c=fgetc(arq1);
        temp++;
    };

    // Verifica os próximos 8 bytes, quatro referentes a largura e os quatro
    // seguintes referentes à altura

    c=fgetc(arq1);
    c1=c;
    c=fgetc(arq1)*16*16;
    c1=c1+c;
    c=fgetc(arq1)*16*16*16*16;
    c1=c1+c;
    c=fgetc(arq1)*16*16*16*16*16*16*16*16;
    Largura=c1;

    c=fgetc(arq1);
    c1=c;
```

```
    c=fgetc(arq1)*16*16;
    c1=c1+c;
    c=fgetc(arq1)*16*16*16*16;
    c1=c1+c;
    c=fgetc(arq1)*16*16*16*16*16*16*16*16;
    Altura=c1;

    //Salta os próximos 28 bytes, pois são cabeçalho a serem repetidos no arquivo destino;

    temp=0;
    while (temp<28) {
        c=fgetc(arq1);
        temp++;
    }

    //Carrega a matriz BMP com o mapa de cores da imagem, onde cada elemento
    //será um byte

    LarguraSize = Largura*3;
    AlturaSize = Altura;

    BMP=calloc(AlturaSize,sizeof(char*));
if(!BMP){
    printf("erro na alocacao de memoria");
    exit(1);;
}

for(t=0; t<AlturaSize; ++t){
    BMP[t] = calloc(LarguraSize,sizeof(char));
  if(!BMP[t]){
    printf("erro na alocacao de memoria");
    exit(1);;
  };
};

for(t=0; t<Altura; ++t){
  for (i=0;i<LarguraSize; ++i) {
    BMP[t][i] = getc(arq1);
  };
};

t=Altura/3; i=LarguraSize/3;

//Verifica o pixel que mais se repete na area central da imagem,
//correpondendo a 1/9 da área total da imagem

MatrixSize=16777216; /* 2^24 elementos, referentes a 24 bits de
cores */

    Moda=calloc(MatrixSize,sizeof(int));
if(!Moda){
    printf("erro na alocacao de memoria");
    exit(1);
```

```
};

i=(LarguraSize/3)+1;
 t=Altura/3;

   for(t=Altura/3; t<2*Altura/3; ++t){
    i=(LarguraSize/3)+1;
       do{
        c=BMP[t][i];
        c1=c*65536;
        c=BMP[t][i+1];
        c=c*256;
        c1=c1+c;
        c=BMP[t][i+2];
        c1=c1+c;
        Moda[c1]=Moda[c1]+1;
        i=i+3;
      }while(i<((2*LarguraSize)/3));
  };

t=0; temp=0; while(t<MatrixSize){
c=Moda[t];
    if(temp<c){
    temp=c;
    M=t;
    };
    t++;
};

  B=0xFF&M;
  M=M>>8;
  G=0xFF&M;
  M=M>>8;
  R=0xFF&M;

free(Moda);

    // Varre o DOCUMENTO de dentro para fora e identifica um ponto como borda
    // lateral esquerda média.

 A=Altura/2;
 temp=LarguraSize/2;
 t=0;
 i=0;
        if((BMP[A][temp]>=R-TOLERANCIA)&&(BMP[A][temp]<=R+TOLERANCIA)
        &&(BMP[A][temp+1]>=G-TOLERANCIA)&&(BMP[A][temp+1]<=G+TOLERANCIA)
        &&(BMP[A][temp+2]>=B-TOLERANCIA)&&(BMP[A][temp+2]<=B+TOLERANCIA)){
        goto teste;
        }else{
        goto testez;
        };

  do{
```

```
teste:
   if((BMP[A][temp]<=R-DOCUMENTO)||(BMP[A][temp]>=R+DOCUMENTO)
     ||(BMP[A][temp+1]<=G-DOCUMENTO)||(BMP[A][temp+1]>=G+DOCUMENTO)
     ||(BMP[A][temp+2]<=B-DOCUMENTO)||(BMP[A][temp+2]>=B+DOCUMENTO)){
        t=1;
        break;
   }else if((BMP[A][temp]>=BMP[A][temp-3]-TOLERANCIA)&&(BMP[A][temp]<=BMP[A][temp-3]+TOLERANCIA)
        &&(BMP[A][temp+1]>=BMP[A][temp-2]-TOLERANCIA)&&(BMP[A][temp+1]<=BMP[A][temp-2]+TOLERANCIA)){
        temp=temp-3;
        goto teste;
        }else if((BMP[A][temp]>=BMP[A][temp-3]-TOLERANCIA)&&(BMP[A][temp]<=BMP[A][temp-3]+TOLERANCIA)
        &&(BMP[A][temp+2]>=BMP[A][temp-1]-TOLERANCIA)&&(BMP[A][temp+2]<=BMP[A][temp-1]+TOLERANCIA)){
        temp=temp-3;
        goto teste;
        }else if((BMP[A][temp+1]>=BMP[A][temp-2]-TOLERANCIA)&&(BMP[A][temp+1]<=BMP[A][temp-2]+TOLERANCIA)
        &&(BMP[A][temp+2]>=BMP[A][temp-1]-TOLERANCIA)&&(BMP[A][temp+2]<=BMP[A][temp-1]+TOLERANCIA)){
        temp=temp-3;
        goto teste;
        }else{
        t=1;
        break;
        };
    }while(t=0);
BordaLateralE=temp/3; i=temp+3;

if(temp>0){
    do{
testez:
   if(temp<=0){
        break;
   };
   if(i>0){
   if((BMP[A][temp]>=BMP[A][i+6]-8)&&(BMP[A][temp]<=BMP[A][i+6]+8)
     &&(BMP[A][temp+1]>=BMP[A][i+7]-8)&&(BMP[A][temp+1]<=BMP[A][i+7]+8)
     &&(BMP[A][temp+2]>=BMP[A][i+8]-8)&&(BMP[A][temp+2]<=BMP[A][i+8]+8)){
        temp=temp-3;
        goto teste;
   };
  };
        if((BMP[A][temp]>=BMP[A][temp-3]-TOLERANCIA)&&(BMP[A][temp]<=BMP[A][temp-3]+TOLERANCIA)
        &&(BMP[A][temp+1]>=BMP[A][temp-2]-TOLERANCIA)&&(BMP[A][temp+1]<=BMP[A][temp-2]+TOLERANCIA)){
        temp=temp-3;
        goto testez;
        }else if((BMP[A][temp]>=BMP[A][temp-3]-TOLERANCIA)&&(BMP[A][temp]<=BMP[A][temp-3]+TOLERANCIA)
        &&(BMP[A][temp+2]>=BMP[A][temp-1]-TOLERANCIA)&&(BMP[A][temp+2]<=BMP[A][temp-1]+TOLERANCIA)){
        temp=temp-3;
        goto testez;
        }else if((BMP[A][temp+1]>=BMP[A][temp-2]-TOLERANCIA)&&(BMP[A][temp+1]<=BMP[A][temp-2]+TOLERANCIA)
        &&(BMP[A][temp+2]>=BMP[A][temp-1]-TOLERANCIA)&&(BMP[A][temp+2]<=BMP[A][temp-1]+TOLERANCIA)){
        temp=temp-3;
        goto testez;
        }else if((BMP[A][temp-3]>=R-DOCUMENTO)&&(BMP[A][temp-3]<=R+DOCUMENTO)
        &&(BMP[A][temp-2]>=G-DOCUMENTO)&&(BMP[A][temp-2]<=G+DOCUMENTO)
```

```
        &&(BMP[A][temp-1]>=B-DOCUMENTO)&&(BMP[A][temp-1]<=B+DOCUMENTO)){
        temp=temp-3;
        goto teste;
        }else if((BMP[A][temp-3]>=BMP[A][i]-TOLERANCIA)&&(BMP[A][temp-3]<=BMP[A][i]+TOLERANCIA)
        &&(BMP[A][temp-2]>=BMP[A][i+1]-TOLERANCIA)&&(BMP[A][temp-2]<=BMP[A][i+1]+TOLERANCIA)
        &&(BMP[A][temp-1]>=BMP[A][i+2]-TOLERANCIA)&&(BMP[A][temp-1]<=BMP[A][i+2]+TOLERANCIA)){
        temp=temp-3;
        goto teste;
        }else{
        temp=temp-3;
        goto testez;
        };
    }while(temp>0);
    };
i1=i;

    //   Varre o documento de dentro para fora e identifica um ponto como borda
    // lateral direita aproximada.

 A=Altura/2;
 temp=3+LarguraSize/2;
 t=0;
 i=0;
        if((BMP[A][temp]>=R-TOLERANCIA)&&(BMP[A][temp]<=R+TOLERANCIA)
        &&(BMP[A][temp+1]>=G-TOLERANCIA)&&(BMP[A][temp+1]<=G+TOLERANCIA)
        &&(BMP[A][temp+2]>=B-TOLERANCIA)&&(BMP[A][temp+2]<=B+TOLERANCIA)){
        goto teste2;
        }else{
        goto testez2;
        };

  do{
teste2:
   if((BMP[A][temp]<=R-DOCUMENTO)||(BMP[A][temp]>=R+DOCUMENTO)
     ||(BMP[A][temp+1]<=G-DOCUMENTO)||(BMP[A][temp+1]>=G+DOCUMENTO)
     ||(BMP[A][temp+2]<=B-DOCUMENTO)||(BMP[A][temp+2]>=B+DOCUMENTO)){
     t=1;
     break;
     }else if((BMP[A][temp]>=BMP[A][temp+3]-TOLERANCIA)&&(BMP[A][temp]<=BMP[A][temp+3]+TOLERANCIA)
        &&(BMP[A][temp+1]>=BMP[A][temp+4]-TOLERANCIA)&&(BMP[A][temp+1]<=BMP[A][temp+4]+TOLERANCIA)){
        temp=temp+3;
        goto teste2;
        }else if((BMP[A][temp]>=BMP[A][temp+3]-TOLERANCIA)&&(BMP[A][temp]<=BMP[A][temp+3]+TOLERANCIA)
        &&(BMP[A][temp+2]>=BMP[A][temp+5]-TOLERANCIA)&&(BMP[A][temp+2]<=BMP[A][temp+5]+TOLERANCIA)){
        temp=temp+3;
        goto teste2;
        }else if((BMP[A][temp+1]>=BMP[A][temp+4]-TOLERANCIA)&&(BMP[A][temp+1]<=BMP[A][temp+4]+TOLERANCIA)
        &&(BMP[A][temp+2]>=BMP[A][temp+5]-TOLERANCIA)&&(BMP[A][temp+2]<=BMP[A][temp+5]+TOLERANCIA)){
        temp=temp+3;
        goto teste2;
        }else{
        t=1;
        break;
```

```
        };
    }while(t=0);
BordaLateralD=temp/3; i=temp-3;

if(temp<LarguraSize){
    do{
testez2:
     if(temp>=LarguraSize){
        break;
     };
    if(i>0){
     if((BMP[A][temp]>=BMP[A][i-9]-8)&&(BMP[A][temp]<=BMP[A][i-9]+8)
     &&(BMP[A][temp+1]>=BMP[A][i-8]-8)&&(BMP[A][temp+1]<=BMP[A][i-8]+8)
     &&(BMP[A][temp+2]>=BMP[A][i-7]-8)&&(BMP[A][temp+2]<=BMP[A][i-7]+8)){
        temp=temp+3;
        goto teste2;
  };
    };
        if((BMP[A][temp]>=BMP[A][temp+3]-TOLERANCIA)&&(BMP[A][temp]<=BMP[A][temp+3]+TOLERANCIA)
        &&(BMP[A][temp+1]>=BMP[A][temp+4]-TOLERANCIA)&&(BMP[A][temp+1]<=BMP[A][temp+4]+TOLERANCIA)){
        temp=temp+3;
        goto testez2;
        }else if((BMP[A][temp]>=BMP[A][temp+3]-TOLERANCIA)&&(BMP[A][temp]<=BMP[A][temp+3]+TOLERANCIA)
        &&(BMP[A][temp+2]>=BMP[A][temp+5]-TOLERANCIA)&&(BMP[A][temp+2]<=BMP[A][temp+5]+TOLERANCIA)){
        temp=temp+3;
        goto testez2;
        }else if((BMP[A][temp+1]>=BMP[A][temp+4]-TOLERANCIA)&&(BMP[A][temp+1]<=BMP[A][temp+4]+TOLERANCIA)
        &&(BMP[A][temp+2]>=BMP[A][temp+5]-TOLERANCIA)&&(BMP[A][temp+2]<=BMP[A][temp+5]+TOLERANCIA)){
        temp=temp+3;
        goto testez2;
        }else if((BMP[A][temp+3]>=R-DOCUMENTO)&&(BMP[A][temp+3]<=R+DOCUMENTO)
        &&(BMP[A][temp+4]>=G-DOCUMENTO)&&(BMP[A][temp+4]<=G+DOCUMENTO)
        &&(BMP[A][temp+5]>=B-DOCUMENTO)&&(BMP[A][temp+5]<=B+DOCUMENTO)){
        temp=temp+3;
        goto teste2;
        }else if((BMP[A][temp+3]>=BMP[A][i]-TOLERANCIA)&&(BMP[A][temp+3]<=BMP[A][i]+TOLERANCIA)
        &&(BMP[A][temp+4]>=BMP[A][i+1]-TOLERANCIA)&&(BMP[A][temp+4]<=BMP[A][i+1]+TOLERANCIA)
        &&(BMP[A][temp+5]>=BMP[A][i+2]-TOLERANCIA)&&(BMP[A][temp+5]<=BMP[A][i+2]+TOLERANCIA)){
        temp=temp+3;
        goto teste2;
        }else{
        temp=temp+3;
        goto testez2;
        };
    }while(temp<LarguraSize);
    };

i2=i;

    //   Varre o documento de dentro para fora e identifica um ponto como borda
    // superior aproximada.

 A=Altura/2;
```

```
 temp=LarguraSize/2;
 t=0;
 i=0;
        if((BMP[A][temp]>=R-TOLERANCIA)&&(BMP[A][temp]<=R+TOLERANCIA)
        &&(BMP[A][temp+1]>=G-TOLERANCIA)&&(BMP[A][temp+1]<=G+TOLERANCIA)
        &&(BMP[A][temp+2]>=B-TOLERANCIA)&&(BMP[A][temp+2]<=B+TOLERANCIA)){
        goto teste3;
        }else{
        goto testez3;
        };
  do{
teste3:
if(A<Altura-1){
   if((BMP[A][temp]<=R-DOCUMENTO)||(BMP[A][temp]>=R+DOCUMENTO)
     ||(BMP[A][temp+1]<=G-DOCUMENTO)||(BMP[A][temp+1]>=G+DOCUMENTO)
     ||(BMP[A][temp+2]<=B-DOCUMENTO)||(BMP[A][temp+2]>=B+DOCUMENTO)){
        t=1;
        break;
   }else if((BMP[A][temp]>=BMP[A+1][temp]-TOLERANCIA)&&(BMP[A][temp]<=BMP[A+1][temp]+TOLERANCIA)
        &&(BMP[A][temp+1]>=BMP[A+1][temp+1]-TOLERANCIA)&&(BMP[A][temp+1]<=BMP[A+1][temp+1]+TOLERANCIA)){
        A=A+1;
        goto teste3;
        }else if((BMP[A][temp]>=BMP[A+1][temp]-TOLERANCIA)&&(BMP[A][temp]<=BMP[A+1][temp]+TOLERANCIA)
        &&(BMP[A][temp+2]>=BMP[A+1][temp+2]-TOLERANCIA)&&(BMP[A][temp+2]<=BMP[A+1][temp+2]+TOLERANCIA)){
        A=A+1;
        goto teste3;
        }else if((BMP[A][temp+1]>=BMP[A+1][temp+1]-TOLERANCIA)&&(BMP[A][temp+1]<=BMP[A+1][temp+1]+TOLERANCIA)
        &&(BMP[A][temp+2]>=BMP[A+1][temp+2]-TOLERANCIA)&&(BMP[A][temp+2]<=BMP[A+1][temp+2]+TOLERANCIA)){
        A=A+1;
        goto teste3;
        }else{
        t=1;
        break;
        };
}else{ t=1; break; };
    }while(t=0);
BordaSuperior=A; i=A-1;

if(A<Altura-1){
    do{
testez3:
   if(A>=Altura-1){
        A=Altura-1;
        break;
   };
   if(i>0){
     if((BMP[A][temp]>=BMP[i-2][temp]-8)&&(BMP[A][temp]<=BMP[i-2][temp]+8)
     &&(BMP[A][temp+1]>=BMP[i-2][temp+1]-8)&&(BMP[A][temp+1]<=BMP[i-2][temp+1]+8)
     &&(BMP[A][temp+2]>=BMP[i-2][temp+2]-8)&&(BMP[A][temp+2]<=BMP[i-2][temp+2]+8)){
        A=A+1;
        goto teste3;
     };
  };
```

```
  if((BMP[A][temp]>=BMP[A+1][temp]-TOLERANCIA)&&(BMP[A][temp]<=BMP[A+1][temp]+TOLERANCIA)
        &&(BMP[A][temp+1]>=BMP[A+1][temp+1]-TOLERANCIA)&&(BMP[A][temp+1]<=BMP[A+1][temp+1]+TOLERANCIA)){
        A=A+1;
        goto testez3;
 }else if((BMP[A][temp]>=BMP[A+1][temp]-TOLERANCIA)&&(BMP[A][temp]<=BMP[A+1][temp]+TOLERANCIA)
        &&(BMP[A][temp+2]>=BMP[A+1][temp+2]-TOLERANCIA)&&(BMP[A][temp+2]<=BMP[A+1][temp+2]+TOLERANCIA)){
        A=A+1;
        goto testez3;
 }else if((BMP[A][temp+1]>=BMP[A+1][temp+1]-TOLERANCIA)&&(BMP[A][temp+1]<=BMP[A+1][temp+1]+TOLERANCIA)
        &&(BMP[A][temp+2]>=BMP[A+1][temp+2]-TOLERANCIA)&&(BMP[A][temp+2]<=BMP[A+1][temp+2]+TOLERANCIA)){
        A=A+1;
        goto testez3;
  }else if((BMP[A+1][temp]>=R-DOCUMENTO)&&(BMP[A+1][temp]<=R+DOCUMENTO)
        &&(BMP[A+1][temp+1]>=G-DOCUMENTO)&&(BMP[A+1][temp+1]<=G+DOCUMENTO)
        &&(BMP[A+1][temp+2]>=B-DOCUMENTO)&&(BMP[A+1][temp+2]<=B+DOCUMENTO)){
        A=A+1;
        goto teste3;
  }else if((BMP[A+1][temp]>=BMP[i][temp]-TOLERANCIA)&&(BMP[A+1][temp]<=BMP[i][temp]+TOLERANCIA)
        &&(BMP[A+1][temp+1]>=BMP[i][temp+1]-TOLERANCIA)&&(BMP[A+1][temp+1]<=BMP[i][temp+1]+TOLERANCIA)
        &&(BMP[A+1][temp+2]>=BMP[i][temp+2]-TOLERANCIA)&&(BMP[A+1][temp+2]<=BMP[i][temp+2]+TOLERANCIA)){
        A=A+1;
        goto teste3;
 }else{
        A=A+1;
        goto testez3;
        };
    }while(A<Altura-1);
    };
i3=i;

    //   Varre o documento de dentro para fora e identifica um ponto como borda
    // inferior aproximada.

 A=Altura/2;
 temp=LarguraSize/2;
 t=0;
 i=0;
        if((BMP[A][temp]>=R-TOLERANCIA)&&(BMP[A][temp]<=R+TOLERANCIA)
        &&(BMP[A][temp+1]>=G-TOLERANCIA)&&(BMP[A][temp+1]<=G+TOLERANCIA)
        &&(BMP[A][temp+2]>=B-TOLERANCIA)&&(BMP[A][temp+2]<=B+TOLERANCIA)){
        goto teste4;
        }else{
        goto testez4;
        };
  do{
teste4:
if(A>0){
   if((BMP[A][temp]<=R-DOCUMENTO)||(BMP[A][temp]>=R+DOCUMENTO)
     ||(BMP[A][temp+1]<=G-DOCUMENTO)||(BMP[A][temp+1]>=G+DOCUMENTO)
     ||(BMP[A][temp+2]<=B-DOCUMENTO)||(BMP[A][temp+2]>=B+DOCUMENTO)){
        t=1;
        break;
   }else if((BMP[A][temp]>=BMP[A-1][temp]-TOLERANCIA)&&(BMP[A][temp]<=BMP[A-1][temp]+TOLERANCIA)
```

```
        &&(BMP[A][temp+1]>=BMP[A-1][temp+1]-TOLERANCIA)&&(BMP[A][temp+1]<=BMP[A-1][temp+1]+TOLERANCIA)){
        A=A-1;
        goto teste4;
        }else if((BMP[A][temp]>=BMP[A-1][temp]-TOLERANCIA)&&(BMP[A][temp]<=BMP[A-1][temp]+TOLERANCIA)
        &&(BMP[A][temp+2]>=BMP[A-1][temp+2]-TOLERANCIA)&&(BMP[A][temp+2]<=BMP[A-1][temp+2]+TOLERANCIA)){
        A=A-1;
        goto teste4;
        }else if((BMP[A][temp+1]>=BMP[A-1][temp+1]-TOLERANCIA)&&(BMP[A][temp+1]<=BMP[A-1][temp+1]+TOLERANCIA)
        &&(BMP[A][temp+2]>=BMP[A-1][temp+2]-TOLERANCIA)&&(BMP[A][temp+2]<=BMP[A-1][temp+2]+TOLERANCIA)){
        A=A-1;
        goto teste4;
        }else{
        t=1;
        break;
        };
}else{ t=1; break; };
    }while(t=0);
BordaInferior=A; i=A+1;

if(A>0){
    do{
testez4:
      if(A<=0){
        A=0;
        break;
      };
   if(i>0){
     if((BMP[A][temp]>=BMP[i+2][temp]-8)&&(BMP[A][temp]<=BMP[i+2][temp]+8)
     &&(BMP[A][temp+1]>=BMP[i+2][temp+1]-8)&&(BMP[A][temp+1]<=BMP[i+2][temp+1]+8)
     &&(BMP[A][temp+2]>=BMP[i+2][temp+2]-8)&&(BMP[A][temp+2]<=BMP[i+2][temp+2]+8)){
        A=A-1;
        goto teste4;
  };
    };
        if((BMP[A][temp]>=BMP[A-1][temp]-TOLERANCIA)&&(BMP[A][temp]<=BMP[A-1][temp]+TOLERANCIA)
        &&(BMP[A][temp+1]>=BMP[A-1][temp+1]-TOLERANCIA)&&(BMP[A][temp+1]<=BMP[A-1][temp+1]+TOLERANCIA)){
        A=A-1;
        goto testez4;
        }else if((BMP[A][temp]>=BMP[A-1][temp]-TOLERANCIA)&&(BMP[A][temp]<=BMP[A-1][temp]+TOLERANCIA)
        &&(BMP[A][temp+2]>=BMP[A-1][temp+2]-TOLERANCIA)&&(BMP[A][temp+2]<=BMP[A-1][temp+2]+TOLERANCIA)){
        A=A-1;
        goto testez4;
        }else if((BMP[A][temp+1]>=BMP[A-1][temp+1]-TOLERANCIA)&&(BMP[A][temp+1]<=BMP[A-1][temp+1]+TOLERANCIA)
        &&(BMP[A][temp+2]>=BMP[A-1][temp+2]-TOLERANCIA)&&(BMP[A][temp+2]<=BMP[A-1][temp+2]+TOLERANCIA)){
        A=A-1;
        goto testez4;
        }else if((BMP[A-1][temp]>=R-DOCUMENTO)&&(BMP[A-1][temp]<=R+DOCUMENTO)
        &&(BMP[A-1][temp+1]>=G-DOCUMENTO)&&(BMP[A-1][temp+1]<=G+DOCUMENTO)
        &&(BMP[A-1][temp+2]>=B-DOCUMENTO)&&(BMP[A-1][temp+2]<=B+DOCUMENTO)){
        A=A-1;
        goto teste4;
        }else if((BMP[A-1][temp]>=BMP[i][temp]-TOLERANCIA)&&(BMP[A-1][temp]<=BMP[i][temp]+TOLERANCIA)
        &&(BMP[A-1][temp+1]>=BMP[i][temp+1]-TOLERANCIA)&&(BMP[A-1][temp+1]<=BMP[i][temp+1]+TOLERANCIA)
```

```
        &&(BMP[A-1][temp+2]>=BMP[i][temp+2]-TOLERANCIA)&&(BMP[A-1][temp+2]<=BMP[i][temp+2]+TOLERANCIA)){
        A=A-1;
        goto teste4;
        }else{
        A=A-1;
        goto testez4;
        };
    }while(A>0);
    };
i4=i;

    //Esta parte do programa verifica a moda da região próxima à borda esquerda

    Moda=calloc(MatrixSize,sizeof(int));
if(!Moda){
    printf("erro na alocacao de memoria");
    exit(1);
};
j=i1;
     if(j<15){
     j=51;
     };
 t=0;
   for(t=BordaInferior; t<BordaSuperior; ++t){
    i=j-15;
       do{
        c=BMP[t][i];
        c1=c*65536;
        c=BMP[t][i+1];
        c=c*256;
        c1=c1+c;
        c=BMP[t][i+2];
        c1=c1+c;
        Moda[c1]=Moda[c1]+1;
        i=i+3;
      }while(i<(j+60));
  };
t=0; temp=0;

while(t<MatrixSize){
c=Moda[t];
    if(temp<c){
    temp=c;
    M=t;
    };
    t++;
};

  B1=0xFF&M;
  M=M>>8;
  G1=0xFF&M;
  M=M>>8;
  R1=0xFF&M;
```

```
free(Moda);

   if((R1<=R-DOCUMENTO)||(R1>=R+DOCUMENTO)
     ||(G1<=G-DOCUMENTO)||(G1>=G+DOCUMENTO)
     ||(B1<=B-DOCUMENTO)||(B1>=B+DOCUMENTO)){
     R1=R;
     G1=G;
     B1=B;
     };


          /* Esta parte do algoritmo identifica em que ponto o DOCUMENTO inicia
           identificando o início do DOCUMENTO de fora para dentro do DOCUMENTO
           varrendo verticalmente a imagem 40 pixels anteriores à borda esquerda
           média encontrada anteriormente. */

t=0; temp=j; A=0;
do{
next:
     if(temp<=0){
     t=1;
     break;
     };
      if((BMP[A][temp]>=R1-FUNDO)&&(BMP[A][temp]<=R1+FUNDO)
      &&(BMP[A][temp+1]>=G1-FUNDO)&&(BMP[A][temp+1]<=G1+FUNDO)
      &&(BMP[A][temp+2]>=B1-FUNDO)&&(BMP[A][temp+2]<=B1+FUNDO)
      &&(BMP[A+1][temp]>=R1-FUNDO)&&(BMP[A+1][temp]<=R1+FUNDO)
      &&(BMP[A+1][temp+1]>=G1-FUNDO)&&(BMP[A+1][temp+1]<=G1+FUNDO)
      &&(BMP[A+1][temp+2]>=B1-FUNDO)&&(BMP[A+1][temp+2]<=B1+FUNDO)){
         if(A>=5){
         A=A-5;
         };
      temp=temp-3;
      goto next;
      }else if(A<Altura-1){
      A=A+1;
      goto next;
      }else{
      t=1;
      break;
      };
}while(t=0);

i1=temp;

    //Esta parte do programa verifica a moda da região próxima à borda direita

    Moda=calloc(MatrixSize,sizeof(int));
if(!Moda){
    printf("erro na alocacao de memoria");
    exit(1);
};
```

```
j=i2;
     if(j>LarguraSize-15){
     j=LarguraSize-15;
     };
 t=0;
   for(t=BordaInferior; t<BordaSuperior; ++t){
    i=j-60;
       do{
        c=BMP[t][i];
        c1=c*65536;
        c=BMP[t][i+1];
        c=c*256;
        c1=c1+c;
        c=BMP[t][i+2];
        c1=c1+c;
        Moda[c1]=Moda[c1]+1;
        i=i+3;
      }while(i<(j+15));
  };
t=0; temp=0;
while(t<MatrixSize){
c=Moda[t];
    if(temp<c){
    temp=c;
    M=t;
    };
    t++;
};

  B1=0xFF&M;
  M=M>>8;
  G1=0xFF&M;
  M=M>>8;
  R1=0xFF&M;
free(Moda);

   if((R1<=R-DOCUMENTO)||(R1>=R+DOCUMENTO)
     ||(G1<=G-DOCUMENTO)||(G1>=G+DOCUMENTO)
     ||(B1<=B-DOCUMENTO)||(B1>=B+DOCUMENTO)){
     R1=R;
     G1=G;
     B1=B;
     };


           /* Esta parte do programa identifica em que ponto o DOCUMENTO inicia
              identificando o início do DOCUMENTO de fora para dentro do DOCUMENTO
              varrendo verticalmente a imagem 40 pixels anteriores à borda direita
              média encontrada anteriormente.*/

t=0; temp=j; A=0;
do{
next1:
```

```
      if(temp>=LarguraSize-1){
      t=1;
      break;
      };
      if((BMP[A][temp]>=R1-FUNDO)&&(BMP[A][temp]<=R1+FUNDO)
      &&(BMP[A][temp+1]>=G1-FUNDO)&&(BMP[A][temp+1]<=G1+FUNDO)
      &&(BMP[A][temp+2]>=B1-FUNDO)&&(BMP[A][temp+2]<=B1+FUNDO)
      &&(BMP[A+1][temp]>=R1-FUNDO)&&(BMP[A+1][temp]<=R1+FUNDO)
      &&(BMP[A+1][temp+1]>=G1-FUNDO)&&(BMP[A+1][temp+1]<=G1+FUNDO)
      &&(BMP[A+1][temp+2]>=B1-FUNDO)&&(BMP[A+1][temp+2]<=B1+FUNDO)){
         if(A>=5){
         A=A-5;
         };
      temp=temp+3;
      goto next1;
      }else if(A<Altura-1){
      A=A+1;
      goto next1;
      }else{
      t=1;
      break;
      };
}while(t=0);
i2=temp;

    //Esta parte do programa verifica a moda da região próxima à borda superior

    Moda=calloc(MatrixSize,sizeof(int));
if(!Moda){
    printf("erro na alocacao de memoria");
    exit(1);
};
j=i3;
     if(j>Altura-5){
     j=Altura-5;
     };
 t=0;
 i=0;
   for(i=(BordaLateralE*3); i<(BordaLateralD*3); i=i+3){
    t=j-20;
       do{
        c=BMP[t][i];
        c1=c*65536;
        c=BMP[t][i+1];
        c=c*256;
        c1=c1+c;
        c=BMP[t][i+2];
        c1=c1+c;
        Moda[c1]=Moda[c1]+1;
        t=t+1;
      }while(t<(j+5));
  };
t=0; temp=0;
```

```
while(t<MatrixSize){
c=Moda[t];
    if(temp<c){
    temp=c;
    M=t;
    };
    t++;
};

  B1=0xFF&M;
  M=M>>8;
  G1=0xFF&M;
  M=M>>8;
  R1=0xFF&M;
free(Moda);

   if((R1<=R-DOCUMENTO)||(R1>=R+DOCUMENTO)
     ||(G1<=G-DOCUMENTO)||(G1>=G+DOCUMENTO)
     ||(B1<=B-DOCUMENTO)||(B1>=B+DOCUMENTO)){
     R1=R;
     G1=G;
     B1=B;
     };

        /* Esta parte do programa identifica em que ponto o DOCUMENTO inicia
           identificando o início do DOCUMENTO de fora para dentro do DOCUMENTO
           varrendo verticalmente a imagem 40 pixels anteriores à borda superior
           média encontrada anteriormente.*/


t=0; A=j; temp=0;
do{
next2:
      if(A>=Altura-1){
      t=1;
      break;
      };
      if((BMP[A][temp]>=R1-FUNDO)&&(BMP[A][temp]<=R1+FUNDO)
      &&(BMP[A][temp+1]>=G1-FUNDO)&&(BMP[A][temp+1]<=G1+FUNDO)
      &&(BMP[A][temp+2]>=B1-FUNDO)&&(BMP[A][temp+2]<=B1+FUNDO)
      &&(BMP[A][temp+3]>=R1-FUNDO)&&(BMP[A][temp+3]<=R1+FUNDO)
      &&(BMP[A][temp+4]>=G1-FUNDO)&&(BMP[A][temp+4]<=G1+FUNDO)
      &&(BMP[A][temp+5]>=B1-FUNDO)&&(BMP[A][temp+5]<=B1+FUNDO)){
         if(temp>=15){
         temp=temp-15;
         };
      A=A+1;
      goto next2;
      }else if(temp<LarguraSize-8){
      temp=temp+3;
      goto next2;
      }else{
```

```
      t=1;
      break;
      };
}while(t=0);

i3=A;

    //Esta parte do programa verifica a moda da região próxima à borda inferior

Moda=calloc(MatrixSize,sizeof(int));
if(!Moda){
    printf("erro na alocacao de memoria");
    exit(1);
};

j=i4;
     if(j<5){
     j=5;
     };
 t=0;
 i=0;
   for(i=(BordaLateralE*3); i<(BordaLateralD*3); i=i+3){
    t=j+20;
       do{
        c=BMP[t][i];
        c1=c*65536;
        c=BMP[t][i+1];
        c=c*256;
        c1=c1+c;
        c=BMP[t][i+2];
        c1=c1+c;
        Moda[c1]=Moda[c1]+1;
        t=t-1;
      }while(t>(j-5));
  };

t=0; temp=0;
while(t<MatrixSize){
c=Moda[t];
    if(temp<c){
    temp=c;
    M=t;
    };
    t++;
};

  B1=0xFF&M;
  M=M>>8;
  G1=0xFF&M;
  M=M>>8;
  R1=0xFF&M;
free(Moda);
```

```
   if((R1<=R-DOCUMENTO)||(R1>=R+DOCUMENTO)
     ||(G1<=G-DOCUMENTO)||(G1>=G+DOCUMENTO)
     ||(B1<=B-DOCUMENTO)||(B1>=B+DOCUMENTO)){
     R1=R;
     G1=G;
     B1=B;
     };

        /* Esta parte do programa identifica em que ponto o DOCUMENTO inicia
           identificando o início do DOCUMENTO de fora para dentro do DOCUMENTO
           varrendo verticalmente a imagem 40 pixels anteriores à borda inferior
           média encontrada anteriormente.*/

t=0; A=j; temp=0;
do{
next3:
      if(A<=0){
      t=1;
      break;
      };
      if((BMP[A][temp]>=R1-FUNDO)&&(BMP[A][temp]<=R1+FUNDO)
      &&(BMP[A][temp+1]>=G1-FUNDO)&&(BMP[A][temp+1]<=G1+FUNDO)
      &&(BMP[A][temp+2]>=B1-FUNDO)&&(BMP[A][temp+2]<=B1+FUNDO)
      &&(BMP[A][temp+3]>=R1-FUNDO)&&(BMP[A][temp+3]<=R1+FUNDO)
      &&(BMP[A][temp+4]>=G1-FUNDO)&&(BMP[A][temp+4]<=G1+FUNDO)
      &&(BMP[A][temp+5]>=B1-FUNDO)&&(BMP[A][temp+5]<=B1+FUNDO)){
         if(temp>=15){
         temp=temp-15;
         };
      A=A-1;
      goto next3;
      }else if(temp<LarguraSize-8){
      temp=temp+3;
      goto next3;
      }else{
      t=1;
      break;
      };
}while(t=0);

i4=A;

//Reinicia o ponteiro do arquivo para copiar parte do cabeçalho do arquivo
//origem e definir o valor da largura/altura da nova imagem. Após essa definição
//o arquivo origem é fechado e o arquivo destino(com corte) é criado.

biWidth=((i2-i1)/3)+1; temp=0;
header:
  while(temp<biWidth){
  temp=temp+4;
  goto header;
  };
biWidth=temp;
```

```
temp=0; biHeight=(i3-i4)+1;
bfSize=(biWidth*biHeight*3)+54; i2=i1+((bfSize-54)/biHeight);

    rewind(arq1);
    while (temp<2) {
        c=fgetc(arq1);
        fputc(c,arq2);
        temp++;
    };
   M=0xFF;
   temp=0;
   while(temp<4){
     c=fgetc(arq1);
     c=M&bfSize;
     fputc(c,arq2);
     bfSize=bfSize>>8;
     temp++;
   };

   temp=0;
    while (temp<12) {
        c=fgetc(arq1);
        fputc(c,arq2);
        temp++;
    };

  temp=0;
   while(temp<4){
   c=M&biWidth;
   fputc(c,arq2);
   biWidth=biWidth>>8;
   temp++;
   };

   temp=0;
   while(temp<4){
   c=M&biHeight;
   fputc(c,arq2);
   biHeight=biHeight>>8;
   temp++;
   };

   temp=0;
    while (temp<8) {
        c=fgetc(arq1);
        temp++;
    };

   temp=0;
    while (temp<28) {
        c=fgetc(arq1);
        fputc(c,arq2);
        temp++;
```

```
    };

    t=0;
    i=0;
temp=0;
    for(t=i4; t<i3+1; ++t){
        for (i=i1; i<i2; ++i){
            c=BMP[t][i];
            fputc(c,arq2);
   temp++;
   }}

    for(t=0; t<AlturaSize; ++t){
    free(BMP[t]);
    }
    free(BMP);
   fclose(arq1);
   fclose(arq2);
   return 1;
}
```

A.2 Detecção dos vértices do documento (origem e destino)

```
//** Programa desenvolvido em linguagem C **//

//Este programa lê documentos digitalizados em formato BMP com 3.2 e 4.1 Mpixels
// sem compressão e estima os vértices do documento. Com a utilização desses
// vértices estima-se a nova localização dos vértices para cálculo do
// homógrafo e posterior correção de perspectiva.

#include <stdio.h> #include <stdlib.h> #include <math.h>

// Inicia programa
int main(int argc, char* argv[]) {

// Declara as variáveis a serem utilizadas


unsigned char **BMP,R,G,B,R1,G1,B1;

unsigned long int biWidth,biHeight,bfSize;

int t=0, i=0, c=0, j=0, c1=0, FinaldeArquivo=0, Altura=0, Largura=0,
A=0, L=0, BMPSize=0, BordaLateralE=0, BordaLateralD=0,
BordaSuperior=0, BordaInferior=0, temp=0, LarguraSize=0, i1=0, i2=0,
i3=0, i4=0, *Moda, x0=0, x1=0, x2=0, x3=0, y0=0, y1=0, y2=0, y3=0,
A1=0, temp1=0, x, y;

double a;
float d1, d2, d3, d4, xd0, xd1, xd2, xd3, yd0, yd1, yd2,
yd3, aspect, b, ya, yb, xa, xb;
```

```
long int MatrixSize,M;

/* Refere-se à tolerância entre variações de mesma cor */ int
tolerancia=16;

/* Refere-se à tolerância entre verificações de pixels do documento
*/ int documento;

 // Prepara o arquivo a ser analisado e o arquivo a ser gerado
    char fundo;
    fundo=32; // Refere-se à tolerância na identificação dos pontos das retas,
              //normalmente igual à variável fundo
    documento=32;

    int g = 0;
    FILE *arq3, *output;
    arq3 = fopen("teste.txt","rb"); // Esse arquivo contém os nomes dos arquivos
                                    // a serem analisados
    output = fopen("saida.txt","wb"); // Esse arquivo contém os valores dos
                                   //pontos estimados como vértices do documento

// Loop para processamento dos 50 documentos

    for(g=0;g<50;g++){
    FILE *arq1, *arq2;
    char nome1[30]="0",nome2[30]="0",nome3[22] ="C:\\teste\\";
    fscanf(arq3,"%s",nome1);

 strcat(nome3,nome1);
    if ((arq1=fopen(nome1,"rb"))==NULL)
    {
        printf("Arquivo %s nao pode ser aberto\n",nome1);
        return 0;
    }

    if ((arq2=fopen(nome3,"wb"))==NULL)
    {
        printf("Arquivo %s nao pode ser aberto ou criado\n",nome3);
        return 0;
    }

  //Ignora os 18 primeiros bytes, pois é o cabecalho a ser repetido no arquido
  // destino

    while (temp<18) {
        c=fgetc(arq1);
        temp++;
    };

    // Verifica os próximos 8 bytes, quatro referentes a largura e os quatro
    //seguintes referentes à altura

    c=fgetc(arq1);
```

```
    c1=c;
    c=fgetc(arq1)*16*16;
    c1=c1+c;
    c=fgetc(arq1)*16*16*16*16;
    c1=c1+c;
    c=fgetc(arq1)*16*16*16*16*16*16*16*16;
    Largura=c1;

    c=fgetc(arq1);
    c1=c;
    c=fgetc(arq1)*16*16;
    c1=c1+c;
    c=fgetc(arq1)*16*16*16*16;
    c1=c1+c;
    c=fgetc(arq1)*16*16*16*16*16*16*16*16;
    Altura=c1;

    //Salta os próximos 28 bytes, pois são cabeçalho a serem repetidos no
    // arquivo destino;

    temp=0;

    while (temp<28) {
        c=fgetc(arq1);
        temp++;
    }

    //Carrega a matriz BMP com o mapa de cores da imagem, onde cada elemento
    //será um byte

    LarguraSize = Largura*3;
    BMP=calloc(Altura,sizeof(char*));
if(!BMP){
    printf("erro na alocacao de memoria");
    exit(1);;
}

for(t=0; t<Altura; ++t){
    BMP[t] = calloc(LarguraSize,sizeof(char));
  if(!BMP[t]){
    printf("erro na alocacao de memoria");
    exit(1);;
  };
};

for(t=0; t<Altura; ++t){
  for (i=0;i<LarguraSize; ++i) {
    BMP[t][i] = getc(arq1);
  };
};

    //Verifica o pixel que mais se repete na area central da imagem,
    // correpondendo a 1/9 da área total da imagem
```

```
MatrixSize=16777216; /* 2^24 elementos, referentes a 24 bits de
cores */
    Moda=calloc(MatrixSize,sizeof(int));

if(!Moda){
    printf("erro na alocacao de memoria");
    exit(1);
};

i=(LarguraSize/3)+1;
 t=Altura/3;
   for(t=Altura/3; t<2*Altura/3; ++t){
    i=(LarguraSize/3)+1; //Para 4.1Mpixels o valor +1 deve ser retirado
       do{
        c=BMP[t][i];
        c1=c*65536;
        c=BMP[t][i+1];
        c=c*256;
        c1=c1+c;
        c=BMP[t][i+2];
        c1=c1+c;
        Moda[c1]=Moda[c1]+1;
        i=i+3;
      }while(i<((2*LarguraSize)/3));
  };
t=0; temp=0;

while(t<MatrixSize){
c=Moda[t];
    if(temp<c){
    temp=c;
    M=t;
    };
    t++;
};

  B=0xFF&M;
  M=M>>8;
  G=0xFF&M;
  M=M>>8;
  R=0xFF&M;
free(Moda);

    //   Varre o documento de dentro para fora e identifica um ponto como borda
    // lateral esquerda aproximada.

 A=Altura/2;
 temp=LarguraSize/2;
 t=0;
 i=0;
        if((BMP[A][temp]>=R-tolerancia)&&(BMP[A][temp]<=R+tolerancia)
        &&(BMP[A][temp+1]>=G-tolerancia)&&(BMP[A][temp+1]<=G+tolerancia)
```

```
        &&(BMP[A][temp+2]>=B-tolerancia)&&(BMP[A][temp+2]<=B+tolerancia)){
        goto teste;
        }else{
        goto testez;
        };

  do{
teste:
   if((BMP[A][temp]<=R-documento)||(BMP[A][temp]>=R+documento)
     ||(BMP[A][temp+1]<=G-documento)||(BMP[A][temp+1]>=G+documento)
     ||(BMP[A][temp+2]<=B-documento)||(BMP[A][temp+2]>=B+documento)){
        t=1;
        break;
   }else if((BMP[A][temp]>=BMP[A][temp-3]-tolerancia)&&(BMP[A][temp]<=BMP[A][temp-3]+tolerancia)
        &&(BMP[A][temp+1]>=BMP[A][temp-2]-tolerancia)&&(BMP[A][temp+1]<=BMP[A][temp-2]+tolerancia)){
        temp=temp-3;
        goto teste;
        }else if((BMP[A][temp]>=BMP[A][temp-3]-tolerancia)&&(BMP[A][temp]<=BMP[A][temp-3]+tolerancia)
        &&(BMP[A][temp+2]>=BMP[A][temp-1]-tolerancia)&&(BMP[A][temp+2]<=BMP[A][temp-1]+tolerancia)){
        temp=temp-3;
        goto teste;
        }else if((BMP[A][temp+1]>=BMP[A][temp-2]-tolerancia)&&(BMP[A][temp+1]<=BMP[A][temp-2]+tolerancia)
        &&(BMP[A][temp+2]>=BMP[A][temp-1]-tolerancia)&&(BMP[A][temp+2]<=BMP[A][temp-1]+tolerancia)){
        temp=temp-3;
        goto teste;
        }else{
        t=1;
        break;
        };
    }while(t=0);
BordaLateralE=temp/3; i=temp+3;

if(temp>0){
    do{
testez:
   if(temp<=0){
        break;
   };
   if(i>0){
   if((BMP[A][temp]>=BMP[A][i+6]-8)&&(BMP[A][temp]<=BMP[A][i+6]+8)
     &&(BMP[A][temp+1]>=BMP[A][i+7]-8)&&(BMP[A][temp+1]<=BMP[A][i+7]+8)
     &&(BMP[A][temp+2]>=BMP[A][i+8]-8)&&(BMP[A][temp+2]<=BMP[A][i+8]+8)){
        temp=temp-3;
        goto teste;
   };
  };
        if((BMP[A][temp]>=BMP[A][temp-3]-tolerancia)&&(BMP[A][temp]<=BMP[A][temp-3]+tolerancia)
        &&(BMP[A][temp+1]>=BMP[A][temp-2]-tolerancia)&&(BMP[A][temp+1]<=BMP[A][temp-2]+tolerancia)){
        temp=temp-3;
        goto testez;
        }else if((BMP[A][temp]>=BMP[A][temp-3]-tolerancia)&&(BMP[A][temp]<=BMP[A][temp-3]+tolerancia)
        &&(BMP[A][temp+2]>=BMP[A][temp-1]-tolerancia)&&(BMP[A][temp+2]<=BMP[A][temp-1]+tolerancia)){
        temp=temp-3;
```

```
        goto testez;
        }else if((BMP[A][temp+1]>=BMP[A][temp-2]-tolerancia)&&(BMP[A][temp+1]<=BMP[A][temp-2]+tolerancia)
        &&(BMP[A][temp+2]>=BMP[A][temp-1]-tolerancia)&&(BMP[A][temp+2]<=BMP[A][temp-1]+tolerancia)){
        temp=temp-3;
        goto testez;
        }else if((BMP[A][temp-3]>=R-documento)&&(BMP[A][temp-3]<=R+documento)
        &&(BMP[A][temp-2]>=G-documento)&&(BMP[A][temp-2]<=G+documento)
        &&(BMP[A][temp-1]>=B-documento)&&(BMP[A][temp-1]<=B+documento)){
        temp=temp-3;
        goto teste;
        }else if((BMP[A][temp-3]>=BMP[A][i]-tolerancia)&&(BMP[A][temp-3]<=BMP[A][i]+tolerancia)
        &&(BMP[A][temp-2]>=BMP[A][i+1]-tolerancia)&&(BMP[A][temp-2]<=BMP[A][i+1]+tolerancia)
        &&(BMP[A][temp-1]>=BMP[A][i+2]-tolerancia)&&(BMP[A][temp-1]<=BMP[A][i+2]+tolerancia)){
        temp=temp-3;
        goto teste;
        }else{
        temp=temp-3;
        goto testez;
        };
    }while(temp>0);
    };

i1=i;

    //   Varre o documento de dentro para fora e identifica um ponto como borda
    // lateral direita aproximada.

 A=Altura/2;
 temp=3+LarguraSize/2;
 t=0;
 i=0;
        if((BMP[A][temp]>=R-tolerancia)&&(BMP[A][temp]<=R+tolerancia)
        &&(BMP[A][temp+1]>=G-tolerancia)&&(BMP[A][temp+1]<=G+tolerancia)
        &&(BMP[A][temp+2]>=B-tolerancia)&&(BMP[A][temp+2]<=B+tolerancia)){
        goto teste2;
        }else{
        goto testez2;
        };

  do{
teste2:
   if((BMP[A][temp]<=R-documento)||(BMP[A][temp]>=R+documento)
     ||(BMP[A][temp+1]<=G-documento)||(BMP[A][temp+1]>=G+documento)
     ||(BMP[A][temp+2]<=B-documento)||(BMP[A][temp+2]>=B+documento)){
     t=1;
     break;
     }else if((BMP[A][temp]>=BMP[A][temp+3]-tolerancia)&&(BMP[A][temp]<=BMP[A][temp+3]+tolerancia)
        &&(BMP[A][temp+1]>=BMP[A][temp+4]-tolerancia)&&(BMP[A][temp+1]<=BMP[A][temp+4]+tolerancia)){
        temp=temp+3;
        goto teste2;
        }else if((BMP[A][temp]>=BMP[A][temp+3]-tolerancia)&&(BMP[A][temp]<=BMP[A][temp+3]+tolerancia)
        &&(BMP[A][temp+2]>=BMP[A][temp+5]-tolerancia)&&(BMP[A][temp+2]<=BMP[A][temp+5]+tolerancia)){
        temp=temp+3;
```

```
        goto teste2;
        }else if((BMP[A][temp+1]>=BMP[A][temp+4]-tolerancia)&&(BMP[A][temp+1]<=BMP[A][temp+4]+tolerancia)
        &&(BMP[A][temp+2]>=BMP[A][temp+5]-tolerancia)&&(BMP[A][temp+2]<=BMP[A][temp+5]+tolerancia)){
        temp=temp+3;
        goto teste2;
        }else{
        t=1;
        break;
        };
    }while(t=0);
BordaLateralD=temp/3; i=temp-3;

if(temp<LarguraSize){
    do{
testez2:
     if(temp>=LarguraSize){
        break;
     };
    if(i>0){
     if((BMP[A][temp]>=BMP[A][i-9]-8)&&(BMP[A][temp]<=BMP[A][i-9]+8)
     &&(BMP[A][temp+1]>=BMP[A][i-8]-8)&&(BMP[A][temp+1]<=BMP[A][i-8]+8)
     &&(BMP[A][temp+2]>=BMP[A][i-7]-8)&&(BMP[A][temp+2]<=BMP[A][i-7]+8)){
        temp=temp+3;
        goto teste2;
  };
    };
        if((BMP[A][temp]>=BMP[A][temp+3]-tolerancia)&&(BMP[A][temp]<=BMP[A][temp+3]+tolerancia)
        &&(BMP[A][temp+1]>=BMP[A][temp+4]-tolerancia)&&(BMP[A][temp+1]<=BMP[A][temp+4]+tolerancia)){
        temp=temp+3;
        goto testez2;
        }else if((BMP[A][temp]>=BMP[A][temp+3]-tolerancia)&&(BMP[A][temp]<=BMP[A][temp+3]+tolerancia)
        &&(BMP[A][temp+2]>=BMP[A][temp+5]-tolerancia)&&(BMP[A][temp+2]<=BMP[A][temp+5]+tolerancia)){
        temp=temp+3;
        goto testez2;
        }else if((BMP[A][temp+1]>=BMP[A][temp+4]-tolerancia)&&(BMP[A][temp+1]<=BMP[A][temp+4]+tolerancia)
        &&(BMP[A][temp+2]>=BMP[A][temp+5]-tolerancia)&&(BMP[A][temp+2]<=BMP[A][temp+5]+tolerancia)){
        temp=temp+3;
        goto testez2;
        }else if((BMP[A][temp+3]>=R-documento)&&(BMP[A][temp+3]<=R+documento)
        &&(BMP[A][temp+4]>=G-documento)&&(BMP[A][temp+4]<=G+documento)
        &&(BMP[A][temp+5]>=B-documento)&&(BMP[A][temp+5]<=B+documento)){
        temp=temp+3;
        goto teste2;
        }else if((BMP[A][temp+3]>=BMP[A][i]-tolerancia)&&(BMP[A][temp+3]<=BMP[A][i]+tolerancia)
        &&(BMP[A][temp+4]>=BMP[A][i+1]-tolerancia)&&(BMP[A][temp+4]<=BMP[A][i+1]+tolerancia)
        &&(BMP[A][temp+5]>=BMP[A][i+2]-tolerancia)&&(BMP[A][temp+5]<=BMP[A][i+2]+tolerancia)){
        temp=temp+3;
        goto teste2;
        }else{
        temp=temp+3;
        goto testez2;
        };
    }while(temp<LarguraSize);
```

```
    };

i2=i;

    //   Varre o documento de dentro para fora e identifica um ponto como borda
    // superior aproximada.

 A=Altura/2;
 temp=LarguraSize/2;
 t=0;
 i=0;
        if((BMP[A][temp]>=R-tolerancia)&&(BMP[A][temp]<=R+tolerancia)
        &&(BMP[A][temp+1]>=G-tolerancia)&&(BMP[A][temp+1]<=G+tolerancia)
        &&(BMP[A][temp+2]>=B-tolerancia)&&(BMP[A][temp+2]<=B+tolerancia)){
        goto teste3;
        }else{
        goto testez3;
        };

  do{
teste3:
 if(A<Altura-1){
   if((BMP[A][temp]<=R-documento)||(BMP[A][temp]>=R+documento)
     ||(BMP[A][temp+1]<=G-documento)||(BMP[A][temp+1]>=G+documento)
     ||(BMP[A][temp+2]<=B-documento)||(BMP[A][temp+2]>=B+documento)){
        t=1;
        break;
   }else if((BMP[A][temp]>=BMP[A+1][temp]-tolerancia)&&(BMP[A][temp]<=BMP[A+1][temp]+tolerancia)
        &&(BMP[A][temp+1]>=BMP[A+1][temp+1]-tolerancia)&&(BMP[A][temp+1]<=BMP[A+1][temp+1]+tolerancia)){
        A=A+1;
        goto teste3;
        }else if((BMP[A][temp]>=BMP[A+1][temp]-tolerancia)&&(BMP[A][temp]<=BMP[A+1][temp]+tolerancia)
        &&(BMP[A][temp+2]>=BMP[A+1][temp+2]-tolerancia)&&(BMP[A][temp+2]<=BMP[A+1][temp+2]+tolerancia)){
        A=A+1;
        goto teste3;
        }else if((BMP[A][temp+1]>=BMP[A+1][temp+1]-tolerancia)&&(BMP[A][temp+1]<=BMP[A+1][temp+1]+tolerancia)
        &&(BMP[A][temp+2]>=BMP[A+1][temp+2]-tolerancia)&&(BMP[A][temp+2]<=BMP[A+1][temp+2]+tolerancia)){
        A=A+1;
        goto teste3;
        }else{
        t=1;
        break;
        };
}else{ t=1; break; };
    }while(t=0);

BordaSuperior=A; i=A-1;
 if(A<Altura-1){
    do{
testez3:
   if(A>=Altura-1){
        A=Altura-1;
        break;
```

```
   };
   if(i>0){
     if((BMP[A][temp]>=BMP[i-2][temp]-8)&&(BMP[A][temp]<=BMP[i-2][temp]+8)
     &&(BMP[A][temp+1]>=BMP[i-2][temp+1]-8)&&(BMP[A][temp+1]<=BMP[i-2][temp+1]+8)
     &&(BMP[A][temp+2]>=BMP[i-2][temp+2]-8)&&(BMP[A][temp+2]<=BMP[i-2][temp+2]+8)){
        A=A+1;
        goto teste3;
     };
  };
  if((BMP[A][temp]>=BMP[A+1][temp]-tolerancia)&&(BMP[A][temp]<=BMP[A+1][temp]+tolerancia)
        &&(BMP[A][temp+1]>=BMP[A+1][temp+1]-tolerancia)&&(BMP[A][temp+1]<=BMP[A+1][temp+1]+tolerancia)){
        A=A+1;
        goto testez3;
 }else if((BMP[A][temp]>=BMP[A+1][temp]-tolerancia)&&(BMP[A][temp]<=BMP[A+1][temp]+tolerancia)
        &&(BMP[A][temp+2]>=BMP[A+1][temp+2]-tolerancia)&&(BMP[A][temp+2]<=BMP[A+1][temp+2]+tolerancia)){
        A=A+1;
        goto testez3;
 }else if((BMP[A][temp+1]>=BMP[A+1][temp+1]-tolerancia)&&(BMP[A][temp+1]<=BMP[A+1][temp+1]+tolerancia)
        &&(BMP[A][temp+2]>=BMP[A+1][temp+2]-tolerancia)&&(BMP[A][temp+2]<=BMP[A+1][temp+2]+tolerancia)){
        A=A+1;
        goto testez3;
  }else if((BMP[A+1][temp]>=R-documento)&&(BMP[A+1][temp]<=R+documento)
        &&(BMP[A+1][temp+1]>=G-documento)&&(BMP[A+1][temp+1]<=G+documento)
        &&(BMP[A+1][temp+2]>=B-documento)&&(BMP[A+1][temp+2]<=B+documento)){
        A=A+1;
        goto teste3;
  }else if((BMP[A+1][temp]>=BMP[i][temp]-tolerancia)&&(BMP[A+1][temp]<=BMP[i][temp]+tolerancia)
        &&(BMP[A+1][temp+1]>=BMP[i][temp+1]-tolerancia)&&(BMP[A+1][temp+1]<=BMP[i][temp+1]+tolerancia)
        &&(BMP[A+1][temp+2]>=BMP[i][temp+2]-tolerancia)&&(BMP[A+1][temp+2]<=BMP[i][temp+2]+tolerancia)){
        A=A+1;
        goto teste3;
 }else{
        A=A+1;
        goto testez3;
        };
    }while(A<Altura-1);
   };
i3=i;

    //   Varre o documento de dentro para fora e identifica um ponto como borda
    // inferior aproximada.


 A=Altura/2;
 temp=LarguraSize/2;
 t=0;
 i=0;
        if((BMP[A][temp]>=R-tolerancia)&&(BMP[A][temp]<=R+tolerancia)
        &&(BMP[A][temp+1]>=G-tolerancia)&&(BMP[A][temp+1]<=G+tolerancia)
        &&(BMP[A][temp+2]>=B-tolerancia)&&(BMP[A][temp+2]<=B+tolerancia)){
        goto teste4;
        }else{
        goto testez4;
```

```
        };

  do{
teste4:
 if(A>0){
   if((BMP[A][temp]<=R-documento)||(BMP[A][temp]>=R+documento)
     ||(BMP[A][temp+1]<=G-documento)||(BMP[A][temp+1]>=G+documento)
     ||(BMP[A][temp+2]<=B-documento)||(BMP[A][temp+2]>=B+documento)){
        t=1;
        break;
   }else if((BMP[A][temp]>=BMP[A-1][temp]-tolerancia)&&(BMP[A][temp]<=BMP[A-1][temp]+tolerancia)
        &&(BMP[A][temp+1]>=BMP[A-1][temp+1]-tolerancia)&&(BMP[A][temp+1]<=BMP[A-1][temp+1]+tolerancia)){
        A=A-1;
        goto teste4;
        }else if((BMP[A][temp]>=BMP[A-1][temp]-tolerancia)&&(BMP[A][temp]<=BMP[A-1][temp]+tolerancia)
        &&(BMP[A][temp+2]>=BMP[A-1][temp+2]-tolerancia)&&(BMP[A][temp+2]<=BMP[A-1][temp+2]+tolerancia)){
        A=A-1;
        goto teste4;
        }else if((BMP[A][temp+1]>=BMP[A-1][temp+1]-tolerancia)&&(BMP[A][temp+1]<=BMP[A-1][temp+1]+tolerancia)
        &&(BMP[A][temp+2]>=BMP[A-1][temp+2]-tolerancia)&&(BMP[A][temp+2]<=BMP[A-1][temp+2]+tolerancia)){
        A=A-1;
        goto teste4;
        }else{
        t=1;
        break;
        };
 }else{ t=1; break; };
    }while(t=0);
BordaInferior=A; i=A+1;

if(A>0){
    do{
testez4:
      if(A<=0){
        A=0;
        break;
      };
   if(i>0){
     if((BMP[A][temp]>=BMP[i+2][temp]-8)&&(BMP[A][temp]<=BMP[i+2][temp]+8)
     &&(BMP[A][temp+1]>=BMP[i+2][temp+1]-8)&&(BMP[A][temp+1]<=BMP[i+2][temp+1]+8)
     &&(BMP[A][temp+2]>=BMP[i+2][temp+2]-8)&&(BMP[A][temp+2]<=BMP[i+2][temp+2]+8)){
        A=A-1;
        goto teste4;
  };
    };
        if((BMP[A][temp]>=BMP[A-1][temp]-tolerancia)&&(BMP[A][temp]<=BMP[A-1][temp]+tolerancia)
        &&(BMP[A][temp+1]>=BMP[A-1][temp+1]-tolerancia)&&(BMP[A][temp+1]<=BMP[A-1][temp+1]+tolerancia)){
        A=A-1;
        goto testez4;
        }else if((BMP[A][temp]>=BMP[A-1][temp]-tolerancia)&&(BMP[A][temp]<=BMP[A-1][temp]+tolerancia)
        &&(BMP[A][temp+2]>=BMP[A-1][temp+2]-tolerancia)&&(BMP[A][temp+2]<=BMP[A-1][temp+2]+tolerancia)){
        A=A-1;
        goto testez4;
```

```
        }else if((BMP[A][temp+1]>=BMP[A-1][temp+1]-tolerancia)&&(BMP[A][temp+1]<=BMP[A-1][temp+1]+tolerancia)
        &&(BMP[A][temp+2]>=BMP[A-1][temp+2]-tolerancia)&&(BMP[A][temp+2]<=BMP[A-1][temp+2]+tolerancia)){
        A=A-1;
        goto testez4;
        }else if((BMP[A-1][temp]>=R-documento)&&(BMP[A-1][temp]<=R+documento)
        &&(BMP[A-1][temp+1]>=G-documento)&&(BMP[A-1][temp+1]<=G+documento)
        &&(BMP[A-1][temp+2]>=B-documento)&&(BMP[A-1][temp+2]<=B+documento)){
        A=A-1;
        goto teste4;
        }else if((BMP[A-1][temp]>=BMP[i][temp]-tolerancia)&&(BMP[A-1][temp]<=BMP[i][temp]+tolerancia)
        &&(BMP[A-1][temp+1]>=BMP[i][temp+1]-tolerancia)&&(BMP[A-1][temp+1]<=BMP[i][temp+1]+tolerancia)
        &&(BMP[A-1][temp+2]>=BMP[i][temp+2]-tolerancia)&&(BMP[A-1][temp+2]<=BMP[i][temp+2]+tolerancia)){
        A=A-1;
        goto teste4;
        }else{
        A=A-1;
        goto testez4;
        };
    }while(A>0);
    };
i4=i;

    //Esta parte do programa verifica a moda da região próxima à borda esquerda

    Moda=calloc(MatrixSize,sizeof(int));
if(!Moda){
    printf("erro na alocacao de memoria");
    exit(1);
};

j=i1;
     if(j<15){
     j=51;
     };
 t=0;

   for(t=BordaInferior; t<BordaSuperior; ++t){
    i=j-15;
       do{
        c=BMP[t][i];
        c1=c*65536;
        c=BMP[t][i+1];
        c=c*256;
        c1=c1+c;
        c=BMP[t][i+2];
        c1=c1+c;
        Moda[c1]=Moda[c1]+1;
        i=i+3;
      }while(i<(j+60));
  };
t=0; temp=0;
while(t<MatrixSize){
c=Moda[t];
```

```
    if(temp<c){
    temp=c;
    M=t;
    };
    t++;
};

  B1=0xFF&M;
  M=M>>8;
  G1=0xFF&M;
  M=M>>8;
  R1=0xFF&M;
free(Moda);

// Esta parte do programa utiliza a moda na região próxima à borda esquerda
// para identificar dois pontos estimados como vértices nas regiões inferior
// esquerda.

t=0; temp=LarguraSize/2; A=0;

do{
 next:
     if(temp<=0){
     t=1;
     break;
     };
      if(((BMP[A][temp]>=R1-fundo)&&(BMP[A][temp]<=R1+fundo)
      &&(BMP[A][temp+1]>=G1-fundo)&&(BMP[A][temp+1]<=G1+fundo)
      &&(BMP[A][temp+2]>=B1-fundo)&&(BMP[A][temp+2]<=B1+fundo)
      &&(BMP[A+1][temp]>=R1-fundo)&&(BMP[A+1][temp]<=R1+fundo)
      &&(BMP[A+1][temp+1]>=G1-fundo)&&(BMP[A+1][temp+1]<=G1+fundo)
      &&(BMP[A+1][temp+2]>=B1-fundo)&&(BMP[A+1][temp+2]<=B1+fundo))
){
     A1=A+1;
     temp1=temp;
            if(y0==0){
             x0=temp;
             y0=A;
             }else if((y0-5<A)&&(y0+5>A)){
                   x0=temp;
                   y0=A;
                                         };
         if(A>=5){
         A=A-5;
         };
     temp=temp-3;
      goto next;
      }else if(A<Altura-1){
      A=A+1;
      goto next;
      }else{
      t=1;
      break;
```

```
      };
}while(t=0);
        BMP[y0][x0]=0;
        BMP[y0][x0+1]=255;
        BMP[y0][x0+2]=0;
fprintf(output,"%d ",x0/3);
fprintf(output,"%d ",y0);

// Esta parte do programa utiliza a moda na região próxima à borda esquerda
// para identificar dois pontos estimados como vértices nas regiões superior
// esquerda.

t=0; temp=LarguraSize/2; A=Altura-1;

do{
 next1:
      if(temp<=0){
      t=1;
      break;
      };
      if(((BMP[A][temp]>=R1-fundo)&&(BMP[A][temp]<=R1+fundo)
      &&(BMP[A][temp+1]>=G1-fundo)&&(BMP[A][temp+1]<=G1+fundo)
      &&(BMP[A][temp+2]>=B1-fundo)&&(BMP[A][temp+2]<=B1+fundo)
      &&(BMP[A-1][temp]>=R1-fundo)&&(BMP[A-1][temp]<=R1+fundo)
      &&(BMP[A-1][temp+1]>=G1-fundo)&&(BMP[A-1][temp+1]<=G1+fundo)
      &&(BMP[A-1][temp+2]>=B1-fundo)&&(BMP[A-1][temp+2]<=B1+fundo))
){
     A1=A-2;
     temp1=temp;
        if(y1==0){
             x1=temp;
             y1=A;
        }else if((y1-5<A)&&(y1+5>A)){
                   x1=temp;
                   y1=A;
                                     };
         if(A-5<=Altura){
         A=A+5;
         };
      temp=temp-3;
      goto next1;
      }else if(A>0){
      A=A-1;
      goto next1;
      }else{
      t=1;
      break;
      };
}while(t=0);
        BMP[y1][x1]=0;
        BMP[y1][x1+1]=255;
        BMP[y1][x1+2]=0;
fprintf(output,"%d ",x1/3);
```

```
fprintf(output,"%d ",y1);

    //Esta parte do programa verifica a moda da região próxima à borda esquerda

    Moda=calloc(MatrixSize,sizeof(int));
if(!Moda){
    printf("erro na alocacao de memoria");
    exit(1);
};

j=i2;
     if(j>LarguraSize-15){
     j=LarguraSize-15;
     };
 t=0;
   for(t=BordaInferior; t<BordaSuperior; ++t){
    i=j-60;
       do{
        c=BMP[t][i];
        c1=c*65536;
        c=BMP[t][i+1];
        c=c*256;
        c1=c1+c;
        c=BMP[t][i+2];
        c1=c1+c;
        Moda[c1]=Moda[c1]+1;
        i=i+3;
      }while(i<(j+15));
  };
t=0; temp=0;

while(t<MatrixSize){
c=Moda[t];
    if(temp<c){
    temp=c;
    M=t;
    };
    t++;
};

  B1=0xFF&M;
  M=M>>8;
  G1=0xFF&M;
  M=M>>8;
  R1=0xFF&M;
free(Moda);

// Esta parte do programa utiliza a moda na região próxima à borda esquerda
// para identificar dois pontos estimados como vértices nas regiões inferior
// direita

t=0; temp=LarguraSize/2+3; A=0;
 do{
```

```
   nextt:
     if(temp>=LarguraSize+3){
     t=1;
     break;
     };
      if((BMP[A][temp]>=R1-fundo)&&(BMP[A][temp]<=R1+fundo)
      &&(BMP[A][temp+1]>=G1-fundo)&&(BMP[A][temp+1]<=G1+fundo)
      &&(BMP[A][temp+2]>=B1-fundo)&&(BMP[A][temp+2]<=B1+fundo)
      &&(BMP[A+1][temp]>=R1-fundo)&&(BMP[A+1][temp]<=R1+fundo)
      &&(BMP[A+1][temp+1]>=G1-fundo)&&(BMP[A+1][temp+1]<=G1+fundo)
      &&(BMP[A+1][temp+2]>=B1-fundo)&&(BMP[A+1][temp+2]<=B1+fundo)){
        if(y2==0){
             x2=temp;
             y2=A;
        }else if((y2-5<A)&&(y2+5>A)){
                   x2=temp;
                   y2=A;
                                    };
          if(A>=5){
         A=A-5;
         };
     temp=temp+3;
      goto nextt;
      }else if(A<Altura-1){
      A=A+1;
      goto nextt;
      }else{
      t=1;
      break;
      };
}while(t=0);
        BMP[y2][x2]=0;
        BMP[y2][x2+1]=255;
        BMP[y2][x2+2]=0;
fprintf(output,"%d ",x2/3);
fprintf(output,"%d ",y2);

// Esta parte do programa utiliza a moda na região próxima à borda esquerda
// para identificar dois pontos estimados como vértices nas regiões superior
// direita

t=0; temp=LarguraSize/2+3; A=Altura-1;
 do{
nextt1:
      if(temp>=LarguraSize){
      t=1;
      break;
      };
      if((BMP[A][temp]>=R1-fundo)&&(BMP[A][temp]<=R1+fundo)
      &&(BMP[A][temp+1]>=G1-fundo)&&(BMP[A][temp+1]<=G1+fundo)
      &&(BMP[A][temp+2]>=B1-fundo)&&(BMP[A][temp+2]<=B1+fundo)
      &&(BMP[A-1][temp]>=R1-fundo)&&(BMP[A-1][temp]<=R1+fundo)
      &&(BMP[A-1][temp+1]>=G1-fundo)&&(BMP[A-1][temp+1]<=G1+fundo)
```

```
      &&(BMP[A-1][temp+2]>=B1-fundo)&&(BMP[A-1][temp+2]<=B1+fundo)){
        if(y3==0){
             x3=temp;
             y3=A;
        }else if((y3-5<A)&&(y3+5>A)){
                   x3=temp;
                   y3=A;
                                    };
         if(A-5<=Altura){
         A=A+5;
         };
      temp=temp+3;
      goto nextt1;
      }else if(A>0){
      A=A-1;
      goto nextt1;
      }else{
      t=1;
      break;
      };
}while(t=0);
        BMP[y3][x3]=0;
        BMP[y3][x3+1]=255;
        BMP[y3][x3+2]=0;
fprintf(output,"%d ",x3/3);
fprintf(output,"%d ",y3);

x0=x0/3; x1=x1/3; x2=x2/3; x3=x3/3;

//Calculo das novas localização dos vértices para correção de perspectiva

d1 = sqrt(pow(x0-x2,2)+pow(y0-y2,2)); d2 =
sqrt(pow(x1-x3,2)+pow(y1-y3,2)); d3 =
sqrt(pow(x0-x1,2)+pow(y0-y1,2)); d4 =
sqrt(pow(x2-x3,2)+pow(y2-y3,2)); aspect = (d1+d2)/(d3+d4);
xd0=xd1=x0; xd2=xd3=xd0+d1; yd0=yd2=y0; yd1=yd3=yd0+(d1/aspect);

fprintf(output,"%f ",xd0);
fprintf(output,"%f ",yd0);
fprintf(output,"%f ",xd1);
fprintf(output,"%f ",yd1);
fprintf(output,"%f ",xd2);
fprintf(output,"%f ",yd2);
fprintf(output,"%f ",xd3);
fprintf(output,"%f \n",yd3);

//Esta parte do programa traça as retas em cor vermelha em um arquivo destino
// para confirmação visual do quadrilátero correspondente ao docuemnto

        //traçado lateral esquerdo
          a=(y1-y0)/(x1-x0);
          b=y0-((y1-y0)/(x1-x0))*x0;
          for(t=y0;t<y1;t++){
```

```
                        x=(t-b)/a;
                        x=x*3;
                        BMP[t][x]=0;
                        BMP[t][x+1]=255;
                        BMP[t][x+2]=0;
                        };

        //traçado lateral direito
          a=(y3-y2)/(x3-x2);
          b=y2-((y3-y2)/(x3-x2))*x2;
          for(t=y2;t<y3;t++){
                        x=(t-b)/a;
                        x=x*3;
                        BMP[t][x]=0;
                        BMP[t][x+1]=255;
                        BMP[t][x+2]=0;
                        };

        //traçado superior
          ya=y1;
          yb=y3;
          xa=x1;
          xb=x3;
          a=(yb-ya)/(xb-xa);
          b=ya-((yb-ya)/(xb-xa))*xa;
          for(t=x1;t<x3;t++){
                        y=a*t+b;
                        x=t*3;
                        BMP[y][x]=0;
                        BMP[y][x+1]=255;
                        BMP[y][x+2]=0;
                        };

        //traçado infeior
          ya=y0;
          yb=y2;
          xa=x0;
          xb=x2;
          a=(yb-ya)/(xb-xa);
          b=ya-((yb-ya)/(xb-xa))*xa;
          for(t=x0;t<x2;t++){
                        y=a*t+b;
                        x=t*3;
                        BMP[y][x]=0;
                        BMP[y][x+1]=255;
                        BMP[y][x+2]=0;
                        };

// Esta parte do programa cria o arquivo destino e fecha o atual documento

    rewind(arq1);
temp=0;
    while (temp<54) {
```

```
        c=fgetc(arq1);
        fputc(c,arq2);
        temp++;
    };
temp=0;
    for(t=0; t<Altura; ++t){
        for (i=0; i<LarguraSize; ++i){
            c=BMP[t][i];
            fputc(c,arq2);
   temp++;
   }}
    for(t=0; t<Altura; ++t){
    free(BMP[t]);
    }
    free(BMP);
   fclose(arq1);
   fclose(arq2);
}
   return 1;
   }
```

A.3 Aplicação da correção de perspectiva

```
//** Programa desenvolvido em linguagem Java **//

public void transformacaoPerspectiva(){

//Lê coordenadas de correção
      Arquivo io = new
Arquivo("G:\\Multimidia\\ImagensIC\\4mp\\42mp\\saida_4MP.txt","null.txt");
      while(!io.isEndOfFile()){
      StringTokenizer st = new StringTokenizer(io.readLine());
      String nome = st.nextToken();

// Método utilizando API java JAI que faz transfomação da pespectiva
//sendo 16 entradas correspondendo ao x y de origem e o x' y' de
//correção
      PerspectiveTransform pers = PerspectiveTransform.getQuadToQuad(
                java.lang.Float.parseFloat(st.nextToken()),
java.lang.Float.parseFloat(st.nextToken()),
                java.lang.Float.parseFloat(st.nextToken()),
java.lang.Float.parseFloat(st.nextToken()),
        java.lang.Float.parseFloat(st.nextToken()),
java.lang.Float.parseFloat(st.nextToken()),
                java.lang.Float.parseFloat(st.nextToken()),
java.lang.Float.parseFloat(st.nextToken()),
        java.lang.Float.parseFloat(st.nextToken()),
java.lang.Float.parseFloat(st.nextToken()),
                java.lang.Float.parseFloat(st.nextToken()),
java.lang.Float.parseFloat(st.nextToken()),
```

```
        java.lang.Float.parseFloat(st.nextToken()),
java.lang.Float.parseFloat(st.nextToken()),
                java.lang.Float.parseFloat(st.nextToken()),
java.lang.Float.parseFloat(st.nextToken()));
        WarpPerspective warp = new WarpPerspective(pers);

//carrega Imagens a serem transformadas
        ParameterBlock pb = new
ParameterBlock().add("G:\\Multimidia\\ImagensIC\\4mp\\42mp\\"+nome);//.substring(0,nome.length()-3)+"jpg");
        RenderedImage im = JAI.create("fileload", pb, null);
        ParameterBlock params = new ParameterBlock();
        params.addSource(im);
        params.add(warp);

//define-se qual método de interpolação
        params.add(new InterpolationNearest());
        RenderedOp image2 = JAI.create("warp", params);
        try {
            // Salva Imagem BMP
            File file = new
File("G:\\Multimidia\\ImagensIC\\4mp\\Nearest\\"+nome);
            ImageIO.write(image2, "BMP", file);
        } catch (IOException e)
            {

        }
        }
}
```

apêndice B

Cópia digital dos códigos-fonte e imagens

CD anexo contendo os códigos-fonte apresentados no apêndice A e imagens de teste.