

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

**ALGORITMOS PARA A MELHORIA DE  
IMAGENS DE DOCUMENTOS**

por

**DANIEL MARQUES OLIVEIRA**

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

**ORIENTADOR: PROF DR. RAFAEL DUEIRE LINS**

Recife, fevereiro de 2012.  
Daniel Marques Oliveira, 2012

Catálogo na fonte  
Bibliotecária: Rosineide Mesquita Gonçalves Luz / CRB4-1361 (BCTG)

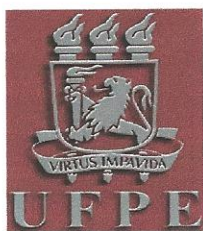
O48a Oliveira, Daniel Marques.  
Algoritmos para a melhoria de imagens de documentos / Daniel Marques  
Oliveira – Recife: O Autor, 2012.  
x, 99f. il., figs., gráfs., tabs.

Orientador: Prof. Dr. Rafael Dueire Lins.

Tese (Doutorado) – Universidade Federal de Pernambuco. CTG.  
Programa de Pós-Graduação em Engenharia Elétrica, 2012.  
Inclui Referências e Apêndice.

1. Engenharia Elétrica. 2. Documentos.. 3. Segmentação de Linhas de texto. 4. Embaçamento. 5. Iluminação irregular. I. Lins, Rafael Dueire (Orientador). II. Título.

621.3 CDD (22.ed) UFPE/BCTG-2012 / 172



# Universidade Federal de Pernambuco

## Pós-Graduação em Engenharia Elétrica

**PARECER DA COMISSÃO EXAMINADORA DE DEFESA DE  
TESE DE DOUTORADO**

# DANIEL MARQUES OLIVEIRA

**TÍTULO**

**“ALGORITMOS PARA A MELHORIA DE IMAGENS DE DOCUMENTOS”**

A comissão examinadora composta pelos professores: RAFAEL DUEIRE LINS, CIN/UFPE; RICARDO MENEZES CAMPELLO DE SOUZA, DES/UFPE; HÉLIO MAGALHÃES DE OLIVEIRA, DES/UFPE; KÁTIA SILVA GUIMARÃES, CIN/UFPE e JEAN MARC-OGIER, Laboratório de Informática/Universidade de La Rochelle, sob a presidência do primeiro, consideram o candidato **DANIEL MARQUES OLIVEIRA APROVADO**.

Recife, 17 de fevereiro de 2012.

**CECÍLIO JOSÉ LINS PIMENTEL**  
Coordenador do PPGEE

**RAFAEL DUEIRE LINS**  
Orientador e Membro Titular Interno

**KÁTIA SILVA GUIMARÃES**  
Membro Titular Externo

**RICARDO MENEZES CAMPELLO DE SOUZA**  
Titular Interno

**JEAN MARC-OGIER**  
Membro Titular Externo

**HÉLIO MAGALHÃES DE OLIVEIRA**  
Membro Titular Externo

## AGRADECIMENTOS

Desejo expressar os meus sinceros agradecimentos a todas as pessoas que contribuíram, direta ou indiretamente, no trabalho que será apresentado nesta tese. Dentre essas pessoas, sou grato em especial:

- Ao meu orientador Rafael, por me guiar durante toda a minha vida acadêmica desde a iniciação científica até o doutorado.
- À minha noiva e futura esposa, Carla pelo seu apoio, compreensão, conselhos e fonte de inspiração durante estes 11 anos juntos.
- Aos meus pais, Sérgio e Glória, e minha irmã e meu cunhado, Martha e Erik, pela suas orientações na minha vida profissional, acadêmica e pessoal.
- À minha avó, Adélia, e meus tios no incentivo ao doutorado.
- À Madalena, Neide, tia Mônica, Rebecca, Roberta e Mercês pela presença nos almoços e cafés durante o desenvolvimento da pesquisa para a tese.
- À equipe do Departamento de Engenharia Elétrica da UFPE, por prover a estrutura necessária para realização do doutorado, em especial, a Andréa (da secretaria de Pós-Graduação), Ricardo (da Telemática) e Paulo (da secretaria de Graduação).
- Aos professores Hélio e Ricardo, por suas aulas e sugestões na tese.
- Ao professor Orgier por se deslocar da França, prestigiar a minha defesa de tese e prover comentários valiosos sobre o meu trabalho.
- À professora Kátia, por seus ensinamentos na cadeira de algoritmos, que foram essenciais no desenvolvimento deste trabalho, e sugestões na tese.
- À todos meus amigos de pós-graduação do PPGEE da UFPE, em especial, a João Marcelo, Bruno, Gabriel e Ednardo.
- Às minhas tias Ana Lúcia, Adriana e Regina pelo seus conselhos e suporte no Rio de Janeiro.
- Aos meus colegas de trabalho na Chemtech/Siemens-RJ, em especial, a Cruz, Cabral, Alessandro, Campinho, Zanon, Leo Melo, Caio, Tercete por me ajudarem a concluir o doutorado.

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

# **ALGORITMOS PARA A MELHORIA DE IMAGENS DE DOCUMENTOS**

**DANIEL MARQUES OLIVEIRA**

Fevereiro/2012

Orientador: Prof. Dr. Rafael Dueire Lins.

Área de Concentração: Telecomunicações.

Palavras-chave: documentos, segmentação de linhas de texto, embaçamento, iluminação irregular.

Número de Páginas: 99.

Câmeras digitais portáteis e *scanners* são equipamentos amplamente utilizados para digitalizar documentos. A imagem obtida pela digitalização pode conter diversos problemas tais como iluminação irregular, embaçamento (*blur*) ou distorções geométricas. A iluminação irregular ocorre quando a luz refletida pelo papel não é uniforme, gerando uma variação na distribuição de iluminação da imagem digitalizada. O defeito de embaçamento tem como uma de suas causas mais frequentes a incorreta calibração focal entre a câmera e o objeto fotografado ou o distanciamento do objeto da plataforma do scanner. Nesse último caso também surge uma distorção geométrica.

Esta tese propõe novos algoritmos para minimizar tais problemas em imagens de documentos fotografados ou escaneados.

Abstract of Thesis presented to UFPE as a partial fulfillment of the requirements for the degree of Doutor in Electrical Engineering.

# **ALGORITMOS PARA A MELHORIA DE IMAGENS DE DOCUMENTOS**

**DANIEL MARQUES OLIVEIRA**

February/2012

Supervisor: Prof. Dr. Rafael Dueire Lins.

Area of Concentration: Telecommunications.

Keywords: documents, text-line segmentation, blur, irregular illumination.

Number of pages: 99.

Portable digital cameras and flatbed scanners are devices widely used to digitize documents. The image acquired by these devices may have several problems such as irregular illumination, blur or geometric distortions. The irregular illumination occurs when the light reflected by the paper is not uniform, yielding into an uneven light distribution in the digitized image. The blur is often caused by the incorrect focal calibration between the camera and the subject or the distance between the flatbed surface. In such scenario, one may also obtain an image with geometric distortion.

This thesis proposes new algorithms to minimize such problems in images acquired by digital cameras or flatbed scanners.

# SUMÁRIO

<b>LISTA DE FIGURAS.....</b>	<b>VIII</b>
<b>LISTA DE TABELAS.....</b>	<b>XI</b>
<b>1 INTRODUÇÃO .....</b>	<b>1</b>
1.1 Objetivos .....	3
1.2 Contribuições.....	5
1.3 Imagens utilizadas .....	6
1.4 Organização da Tese .....	7
<b>2 SEGMENTAÇÃO DE LINHAS DE TEXTO .....</b>	<b>8</b>
2.1 Trabalhos relacionados.....	8
2.2 Primeiro Algoritmo Proposto.....	11
2.3 Segundo Algoritmo Proposto.....	15
2.3.1 Estudo tipográfico sobre a forma dos caracteres.....	17
2.3.2 Resumo da segunda proposta .....	21
2.4 Resultados.....	23
2.4.1 Comparação das medidas de qualidade.....	28
<b>3 CORREÇÃO DE DISTORÇÃO GEOMÉTRICA.....</b>	<b>34</b>
3.1 Trabalhos relacionados.....	34
3.2 Modelo de captura de imagens por <i>scanner</i> .....	37
3.3 Primeiro Algoritmo Proposto .....	38
3.4 Segundo Algoritmo Proposto.....	43
<b>4 REMOÇÃO DE EMBAÇAMENTO (<i>BLUR</i>).....</b>	<b>54</b>

4.1	Trabalhos relacionados.....	54
4.2	Algoritmo proposto .....	55
4.2.1	Resultados .....	56
<b>5</b>	<b>REMOÇÃO DE SOMBREAMENTO .....</b>	<b>65</b>
5.1	Trabalhos relacionados.....	65
5.2	Primeiro Algoritmo Proposto .....	66
5.3	Segundo Algoritmo Proposto .....	68
5.4	Terceiro Algoritmo Proposto .....	74
<b>6</b>	<b>CONCLUSÕES.....</b>	<b>88</b>
6.1	Trabalhos futuros .....	89
	<b>REFERÊNCIAS .....</b>	<b>90</b>
	<b>APÊNDICE A .....</b>	<b>98</b>
A.1	Publicações sobre segmentação de linhas de texto .....	98
A.2	Publicação sobre estudo estatístico sobre elementos tipográficos de texto .....	98
A.3	Publicações sobre correção geométrica .....	98
A.4	Publicação sobre remoção de embaçamento.....	98
A.5	Publicações sobre remoção de sombreamento.....	99



## LISTA DE FIGURAS

Figura 1.1 Documentos digitalizados por <i>scanners</i> de mesa em posições distintas .....	2
Figura 1.2 Documentos digitalizados por câmeras .....	3
Figura 1.3 Exemplo de uma proposta de remoção de distorção geométrica .....	4
Figura 1.4 Exemplo de uma proposta de remoção de sombreamento .....	4
Figura 1.5 Exemplo de uma proposta de remoção de embaçamento .....	5
Figura 2.1 Expansão da linha utilizando serpentes acopladas .....	10
Figura 2.2 Detecção de linhas de texto (BUKHARI <i>et al.</i> , 2011) .....	11
Figura 2.3. Formação de linha de texto feita em (ÁVILA e LINS, 2005) .....	12
Figura 2.4 Letra maiúscula seguida de minúscula.....	13
Figura 2.5 Imagem binária disponível na base CBDAR 2007 .....	14
Figura 2.6 Linhas de texto detectadas da Figura 2.5 por (OLIVEIRA <i>et al.</i> , 2010a) .....	15
Figura 2.7 Informação obtida a partir dos componentes conectados .....	16
Figura 2.8 Letras “b” e “y” possuem medidas semelhantes, porém não são um bom início para a formação de linha.....	17
Figura 2.9. Linhas tipográficas descritas em (NAYLOR, 2010) .....	17
Figura 2.10 Ilustração da identificação de linhas pela segmentação (OLIVEIRA <i>et al.</i> , 2011) .....	22
Figura 2.11 Obtenção das medidas de qualidade para segmentações de linhas de texto .....	24
Figura 2.12 Ilustração das medidas usadas na comparação da segmentação: ruídos em preto puro; linhas detectadas em cores.....	25
Figura 2.13 Gráfico das medidas de Cobertura vs Precisão das segmentações .....	28
Figura 2.14 Exemplos de linhas perdidas.....	29
Figura 2.15 Imagem de referência obtida da base CBDAR 2007 .....	30
Figura 2.16 Exemplo de alarme falso.....	31
Figura 2.17 Exemplo de processamento feito com sucesso. ....	33
Figura 3.1 Correção geométrica proposta por (TAN <i>et al.</i> , 2006) que requer vários parâmetros de scanners .....	35
Figura 3.2 Resultado de algoritmos de correção geométrica para câmeras digitais.....	36
Figura 3.3 Comprovação sobre o modelo de projeção de <i>scanner</i> .....	37
Figura 3.4 Livro com papel quadriculado posto sobre o meio do livro.....	38
Figura 3.5 Imagem usada para calcular a relação entre intensidade e altura .....	39
Figura 3.6 Imagem escaneada com distorção geométrica de dimensões iguais a 1810x3208 <i>pixels</i> . .....	39
Figura 3.7 Aplicação da primeira proposta de correção geométrica na Figura 3.6, que alterou as dimensões para 1939x3208 <i>pixels</i> .....	40
Figura 3.8 Imagem escaneada com distorção geométrica de dimensões iguais a 1472x2744 <i>pixels</i> . .....	41
Figura 3.9 Aplicação da primeira proposta de correção geométrica na Figura 3.8, que alterou as dimensões para 1562x2744 <i>pixels</i> .....	42

Figura 3.10 Cenários de digitalização por <i>scanners</i> .....	43
Figura 3.11 Identificação de método de extração da altura do livro .....	44
Figura 3.12 Coordenadas $x$ , $y$ e $z$ relativas à página do livro digitalizado .....	44
Figura 3.13 Validação da aproximação geométrica pela função exponencial .....	45
Figura 3.14 Imagem escaneada no cenário horizontal da Figura 3.11 com .....	46
Figura 3.15 Aplicação da segunda proposta de correção geométrica na Figura 3.14, que alterou as dimensões para 1653x2744 <i>pixels</i> .....	47
Figura 3.16 Imagem escaneada no cenário horizontal da Figura 3.11 com dimensões iguais a 1544x2728 <i>pixels</i> . .....	48
Figura 3.17 Aplicação da segunda proposta de correção geométrica na Figura 3.16, que alterou as dimensões para 1544x2728 <i>pixels</i> .....	49
Figura 3.18 Imagem escaneada no cenário vertical da Figura 3.11 com dimensões iguais a 2551x1768 <i>pixels</i> . .....	50
Figura 3.19 Aplicação da segunda proposta de correção geométrica na Figura 3.18, que alterou as dimensões para 2858x1927 <i>pixels</i> .....	51
Figura 3.20 Imagem escaneada no cenário vertical da Figura 3.11 com dimensões iguais a 2551x1680 <i>pixels</i> . .....	52
Figura 3.21 Aplicação da segunda proposta de correção geométrica na Figura 3.20 que alterou as dimensões para 2771x1773 <i>pixels</i> .....	53
Figura 4.1 Oscilações após a aplicação da remoção de embaçamento .....	55
Figura 4.2 Função $St$ para $p$ igual a 1,00 e 0,06 .....	56
Figura 4.3 Obtenção das medidas de qualidade para a remoção de embaçamento .....	57
Figura 4.4 Letra “M” gerada por computador com vários níveis de embaçamento .....	57
Figura 4.5 Regiões periféricas da letra “M” .....	60
Figura 4.6 Resultado do filtro de remoção de embaçamento com janela 3x3 .....	60
Figura 4.7 Resultado do filtro de remoção de embaçamento com $p = 0,25$ .....	62
Figura 5.1 Remoção de sombreamento aplicada em um quadro branco .....	67
Figura 5.2 Remoção de sombreamento aplicada num quadro verde.....	68
Figura 5.3 Dilatação iterativa para estimar cor do papel .....	69
Figura 5.4 Documento com iluminação gerada por computador .....	70
Figura 5.5 Processamento da segunda proposta de remoção de sombreamento sobre a Figura 5.4 .....	71
Figura 5.6 Fotografia de página do livro “Sherlock Homes: Vale do Terror” .....	72
Figura 5.7 Processamento da segunda proposta de remoção de sombreamento sobre a Figura 5.6 .....	73
Figura 5.8 Estimativa de sombreamento da segunda proposta .....	74
Figura 5.9 Identificação da região pertencente ao papel.....	75
Figura 5.10 Entrelaçamento de <i>bits</i> na obtenção do código .....	75
Figura 5.11 Iteração na ordem- $z$ sobre o espaço bidimensional (EPPSTEIN, 2011) .....	76

Figura 5.12 Criação da base de dados com iluminação gerada por computador .....	77
Figura 5.13 Obtenção de medidas de similaridades .....	77
Figura 5.14 Relação entre o tamanho da imagem e o tempo de processamento .....	79
Figura 5.15 Imagem de referência a partir de uma página da revista Negócios PE .....	80
Figura 5.16 Imagem com distorção relativa a Figura 5.15.....	81
Figura 5.17 Processamento pela terceira proposta na Figura 5.16 com similaridades SSIM relativas a imagem de referência da Figura 5.15 iguais a 96,90% (Verm.), 97,57% (Verde), 95,68% (Azul) e 97,96% (Cinza) .....	82
Figura 5.18 Processamento da terceira proposta de remoção de sombreamento sobre a Figura 5.9 .....	83
Figura 5.19 Página de uma edição da revista Negócios PE.....	84
Figura 5.20 Processamento da terceira proposta de remoção de sombreamento sobre a Figura 5.19 .....	85
Figura 5.21 Duas páginas de uma edição da revista Negócios PE .....	86
Figura 5.22 Processamento da terceira proposta de remoção de sombreamento sobre a Figura 5.21 .....	87

## LISTA DE TABELAS

Tabela 2.1 Medidas tipográficas relativas a <i>x-height</i> _____	18
Tabela 2.2 Percentuais de pares de caracteres conforme sua altura tipográfica _____	19
Tabela 2.3 Classificação de caracteres em relação às linhas da Figura 2.9 _____	20
Tabela 2.4 Comparação entre as frequências das formas das letras obtidas _____	20
Tabela 2.5 Medidas de qualidade dos algoritmos de segmentação de linhas de texto _____	27
Tabela 2.6 Outras medidas sugeridas de qualidade dos algoritmos _____	27
Tabela 4.1 Média SSIM entre as imagens sem embaçamento e corrigidas com $p = 0,06$ _____	59
Tabela 5.1 Qualidade (SSIM) do algoritmo proposto _____	78
Tabela 5.2 Tempo (em segundos) do processamento do algoritmo _____	79

# 1 Introdução

Há séculos a humanidade vem utilizando o papel como meio de armazenamento e transmissão de informação. No entanto, além do papel sofrer desgaste natural pelo envelhecimento, ele pode ser danificado pelo manuseio incorreto. Acrescente-se a isso, o transporte físico de documentos pode demandar custo e tempo elevados. Essas desvantagens podem ser contornadas através da digitalização.

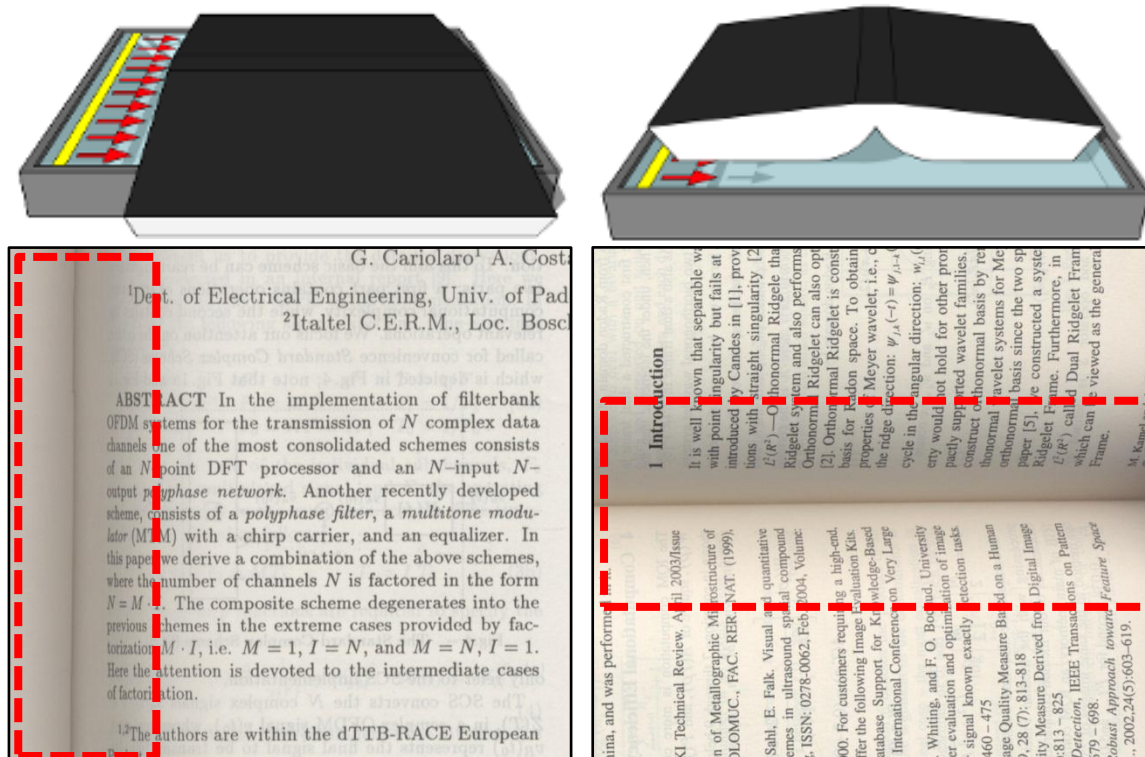
A digitalização apresenta inúmeras vantagens. Ela possibilita a cópia de documentos de maneira fácil. O documento no formato digital, aliado à infraestrutura de rede de computadores, tal como a Internet, possibilita uma maior divulgação a custo e tempo menores, comparado ao transporte físico. Caso o formato digital possua texto, é possível a pesquisa por palavras-chave do mesmo. Vale ressaltar que essa diminuição de custo e facilidade de divulgação tornam o acesso à informação mais democrático.

Câmeras digitais portáteis e *scanners* são os equipamentos mais amplamente utilizados na digitalização de documentos. Atualmente, a disponibilidade desses dispositivos tem contribuído para o aumento significativo da digitalização de acervos das mais diversas naturezas.

Dentre os tipos de *scanners* existentes no mercado, o de mesa é o mais usual. Nele, o documento é posto sobre uma superfície de vidro e é pressionado por uma tampa opaca para aumentar a aderência ao vidro. Caso o objeto digitalizado seja um livro, a superfície da página perto da brochura pode não estar em contato com o vidro da mesa digitalizadora. Nesse caso, a imagem resultante apresenta:

- deformação geométrica;
- iluminação variável conforme a distância objeto-superfície;
- embaçamento (*blur*) da região não aderida ao vidro que não está no foco do sensor.

A Figura 1.1 exemplifica alguns dos problemas descritos acima, onde dois livros foram digitalizados em posições distintas. Observa-se que as linhas de texto tornam-se curvas perto da lombada na ilustração da direita, o que não acontece na ilustração da esquerda. Isto ocorre devido à projeção ser ortográfica na direção em que o sensor se move e projetiva na direção ortogonal (SCHUBERT, 2000; TAN *et al.*, 2006). Em relação à iluminação, esta é variável na região de junções de páginas.



a. Dobra ortogonal ao movimento do sensor    b. Dobra paralela ao movimento do sensor  
**Figura 1.1 Documentos digitalizados por *scanners* de mesa em posições distintas**

Outro meio de digitalização é através de câmeras digitais portáteis. Este processo é menos “intrusivo” que o do *scanner* de mesa, pois não existe o contato físico direto entre o objeto e a câmera. A câmera, por ser altamente portátil, permite que a digitalização ocorra no próprio local onde se encontra o documento. Assim como o *scanner*, o documento digitalizado por câmeras pode apresentar os seguintes defeitos:

- distorção geométrica quando a superfície do documento não é plana ou quando o documento não está perpendicular ao eixo da lente da câmera;
- iluminação irregular;
- embaçamento da região fora de foco;
- embaçamento devido ao ato de “tremar” a câmera ao capturar a foto.

A Figura 1.2 apresenta dois exemplos de documentos digitalizados por câmeras digitais onde é possível observar todos os problemas descritos acima. Observa-se que o tratamento deste tipo de imagem é bem mais complexo em relação ao tratamento de imagens digitalizadas por *scanners*, apesar de ser um dispositivo portátil e com uma captura mais rápida.

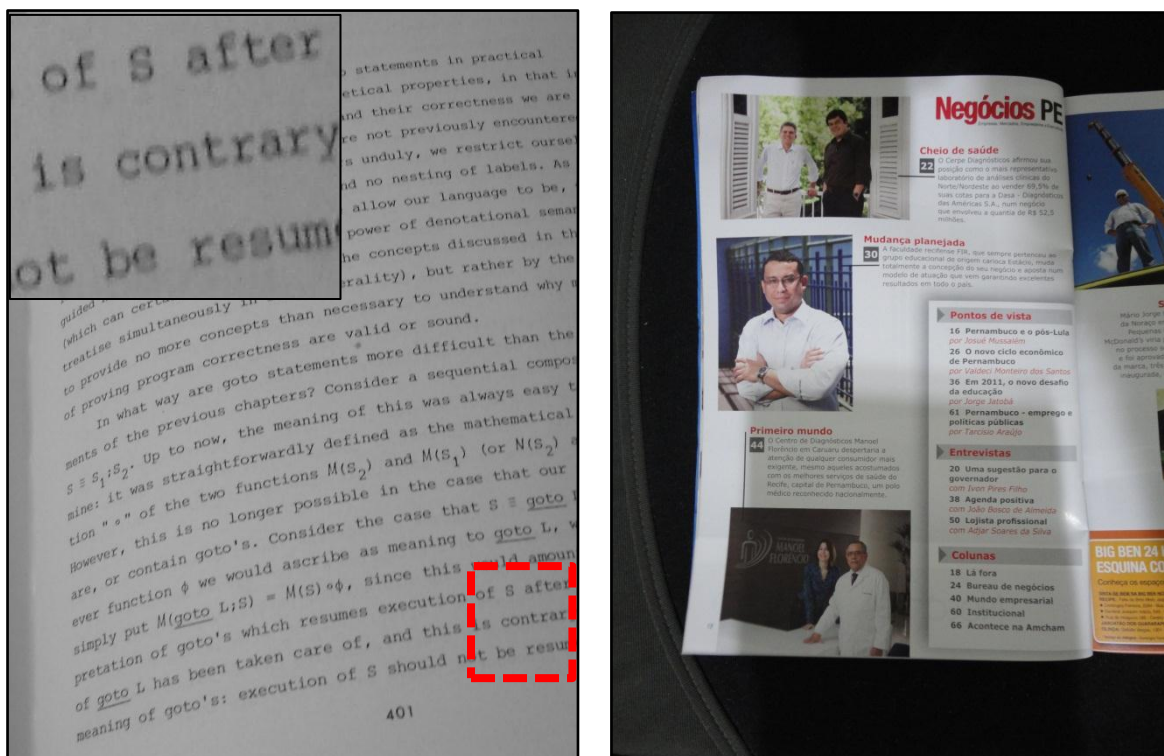


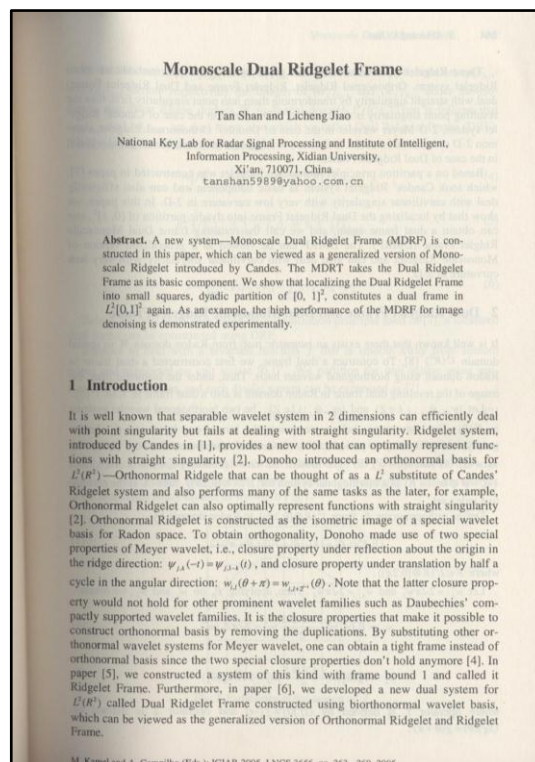
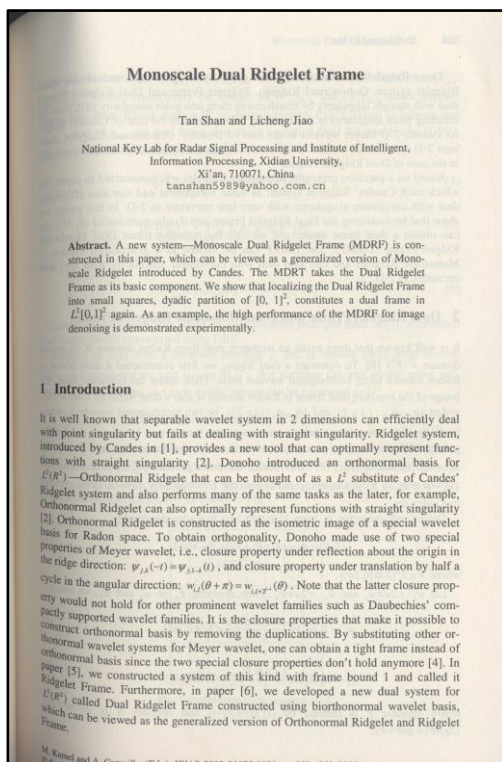
Figura 1.2 Documentos digitalizados por câmeras

## 1.1 Objetivos

Tendo em vista os problemas citados anteriormente, esta tese apresenta algoritmos para:

- Corrigir a distorção geométrica ocasionada na digitalização de documentos encadernados através de *scanners* de mesa;
- Detectar linhas de texto em imagens obtidas por *scanners* de mesa e câmeras portáteis, que é um pré-processamento da correção de distorção geométrica;
- Uniformizar iluminação irregular na digitalização por câmeras portáteis;
- Reduzir embaçamento por falta de foco na digitalização por *scanners* de mesa.

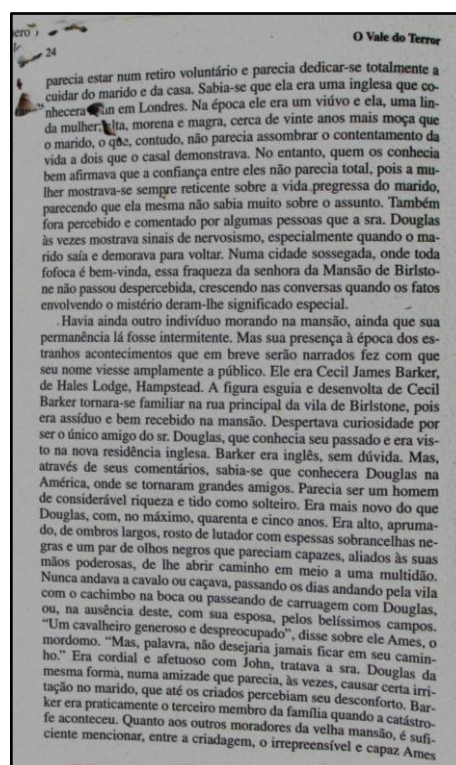
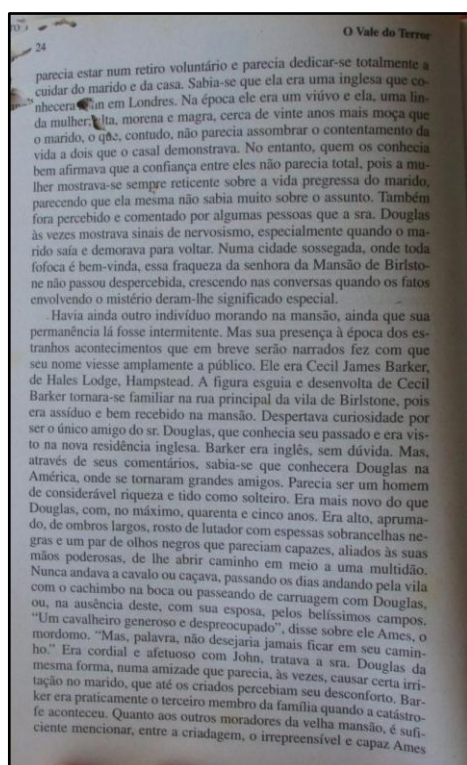
As Figuras 1.3-1.5 mostram alguns resultados das propostas a serem apresentadas nesta tese. As imagens da esquerda indicam aquela que contém a distorção, a correspondente da direita mostra o resultado de alguns dos métodos a serem vistos mais adiante.



a. Imagem com distorção geométrica

b. Imagem corrigida

Figura 1.3 Exemplo de uma proposta de remoção de distorção geométrica

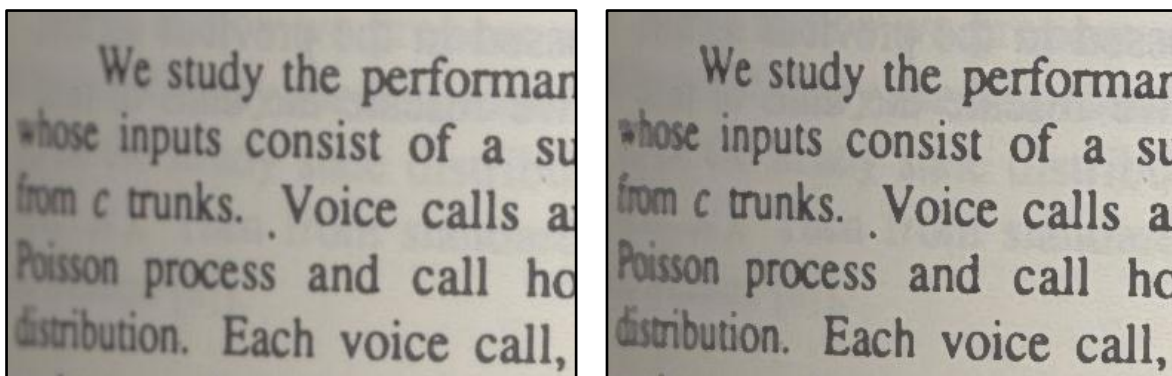


a. Imagem com iluminação irregular

b. Imagem corrigida

Figura 1.4 Exemplo de uma proposta de remoção de sombreamento





a. Imagem com embaçamento

b. Imagem corrigida

**Figura 1.5 Exemplo de uma proposta de remoção de embaçamento**

## 1.2 Contribuições

Esta tese tem como contribuições:

- Novos métodos para detectar linhas de texto:
  - (OLIVEIRA *et al.*, 2010a) – D. M. Oliveira, R. D. Lins, G. Torreão, J. Fan e M. Thielo. A New Method for Text-line Segmentation for Warped Documents. Proceedings of ICIAR 2010. LNCS 6112, pp. 398-408, Springer Verlag, 2010.
  - (OLIVEIRA *et al.*, 2011a) – D. M. Oliveira, R. D. Lins, G. Torreão, J. Fan e M. Thielo. A New Algorithm for Segmenting Warped Text-lines in Document Images. ACM SAC 2011, Taichung, Taiwan, 2011.
  - (OLIVEIRA *et al.*, 2011b) – D. M. Oliveira, R. D. Lins, G. Torreão, J. Fan and M. Thielo. An Efficient Algorithm for Segmenting Warped Text-lines in Document Images. (Submetido em 6 de Julho de 2011 ao IJDAR, em revisão).
- Estudo estatístico sobre elementos tipográficos de texto utilizado no aprimoramento da segmentação de linhas de texto:
  - (OLIVEIRA *et al.*, 2012) – D. M. Oliveira, R. D. Lins e G. S. Torreão. Characters' Shape Analysis of Printed Documents. (Submetido para ser publicado no ICPR 2012).
- O desenvolvimento de dois novos algoritmos para corrigir a distorção geométrica na digitalização por *scanners*:
  - (LINS *et al.*, 2010a) – R. D. Lins, D. M. Oliveira, G. Torreão, J. Fan e M. Thielo. A Dewarping Algorithm to Compensate Volume Binding Distortion in Scanned Documents. ACM SAC 2010, 2010.

- (LINS *et al.*, 2010b) – R. D. Lins, D. M. Oliveira, G. Torreão, J. Fan e M. Thielo. Correcting Book Binding Distortion in Scanned Documents. Proceedings of ICIAR 2010. LNCS 6112, pp. 355-365, Springer Verlag, 2010.
- Três algoritmos para uniformizar a iluminação em documentos fotografados com câmeras digitais portáteis:
  - (OLIVEIRA e LINS, 2010) – D. M. Oliveira e R. D. Lins. Generalizing Tableau to Any Color of Teaching Boards. Proceedings of ICPR 2010. pp.2411-2414, 2010.
  - (OLIVEIRA e LINS, 2009) – D. M. Oliveira e R. D. Lins. A New Method for Shading Removal and Binarization of Documents Acquired with Portable Digital Cameras. Proceedings of Third International Workshop on Camera-Based Document Analysis and Recognition, pp.3-10, 2009.
  - (OLIVEIRA e LINS, 2011) – D. M. Oliveira e R. D. Lins. Shading Removal of Camera Documents with Natural Neighbor Interpolation. (Submetido em 10 de Junho de 2011 ao IJDAR).
- Uma proposta para reduzir o embaçamento originado pela falta de foco em documentos digitalizados por *scanner*:
  - (OLIVEIRA *et al.*, 2010b) – D. M. Oliveira, R. D. Lins, G. Torreão, J. Fan and M. Thielo. A New Deblurring Algorithm for Textual Document Images. Proceedings of the International Telecommunications Symposium, Manaus, Brasil, 2010.
  - (OLIVEIRA *et al.*, 2011c) – D. M. Oliveira, R. D. Lins e G. S. Torreão. Deblurring Textual Document Images (Submetido para ser publicado no LNCS do GREC 2011).

Cada um dos artigos listados acima encontra-se no Apêndice a esta tese.

### 1.3 Imagens utilizadas

Para realizar os estudos e testes apresentados nesta tese, foram utilizadas imagens das seguintes bases:

- LiveMemory (LINS *et al.*, 2009) – imagens digitalizadas por *scanner* dos volumes dos anais da SBrT desde 1982.

- CBDAR 2007 Dewarping dataset (SHAFAIT e BRUEUL, 2007) – imagens de documentos fotografados para competição de remoção de distorção geométrica por câmeras. A base também contém linhas de texto segmentadas manualmente utilizadas na comparação de algoritmos com este propósito.
- Imagens da revista Negócios PE (NAJAIM & AGUIAR Ltd, 2011a; NAJAIM & AGUIAR Ltd, 2011b) obtidas por *scanner* combinadas com modelos de iluminação semelhantes ao obtido por câmeras digitais portáteis. Essa base foi feita para validar o algoritmo de uniformização e estará publicamente disponível para uso não comercial após a publicação do artigo (OLIVEIRA e LINS, 2011).

## 1.4 Organização da Tese

Esta tese contém seis capítulos incluindo o presente capítulo de introdução. Em cada capítulo expõem-se soluções para os problemas mencionados na seção de objetivos. Os capítulos estão estruturados conforme a seguir.

- Introdução sobre o problema a ser solucionado.
- Revisão bibliográfica dos trabalhos relacionados.
- Descrição resumida sobre cada proposta apresentada. Os detalhes sobre as propostas estão presentes nos artigos (publicados ou em processo de revisão) que estão disponíveis no apêndice desta tese.

Em anexo a esta tese pode ser encontrado um DVD que contém a base de imagens com os resultados de processamento de cada proposta, bem como os códigos fonte e executável na plataforma Java e C++ dos algoritmos desenvolvidos que não possuem restrições de confidencialidade do projeto de P&D da Hewlett-Packard/UFPE do qual participou o submitante.

O termo “documento” será utilizado nesta tese de forma ampla para designar livros, revistas, jornais ou qualquer outro objeto que contenha informação textual. Documentos impressos são os tipos de documentos mais populares, porém nessa tese também será trabalhada a informação de imagens de quadros didáticos.

## 2 Segmentação de Linhas de Texto

A segmentação de linhas de texto em imagens de documentos de texto impressos consiste em identificar grupos de caracteres que representam cada linha de um parágrafo. Este é um dos passos na execução de algoritmos de OCR, correção geométrica e identificação de texto em imagens reais.

Na literatura técnica existem várias propostas, mas apenas o grupo da Universidade de Kaiserslautern criou e utiliza a base de imagens CBDAR 2007. O grupo usa esta base para avaliar algoritmos de segmentação de linhas de texto com medidas quantitativas. Nos algoritmos (OLIVEIRA *et al.*, 2010a; OLIVEIRA *et al.*, 2011a; OLIVEIRA *et al.*, 2011b) foi conseguida substancial melhoria de desempenho. O mais recente (OLIVEIRA *et al.*, 2011b) supera as taxas de precisão<sup>1</sup>, cobertura<sup>2</sup> e medida-F<sup>3</sup> dos algoritmos concorrentes nessa base de dados. Além do que, o tempo de processamento é aplicável na prática para iteração com o usuário, sendo em média 4s para uma máquina com processador Celeron 2,0GHz e 3GB de RAM rodando no Windows 7 Professional; dos algoritmos pesquisados os únicos que apresentam tais dados são (BUKHARI *et al.*, 2009b; BUKHARI *et al.*, 2011).

### 2.1 Trabalhos relacionados

Os primeiros algoritmos deste tipo foram desenvolvidos para processamento de imagens digitalizadas por *scanner*. Neste caso, a segmentação era usada para transformar a imagem em texto, o que é muitas vezes acompanhada de uma identificação automática de regiões do documento em ilustração, parágrafo, título, número de página etc.

Como mencionado na introdução desta tese, um documento digitalizado com câmeras digitais apresenta iluminação irregular, embaçamento e distorção geométrica. Devido a

---

<sup>1</sup> Precisão – percentual de linhas identificadas corretamente dentre as linhas identificadas

<sup>2</sup> Cobertura – percentual de linhas identificadas corretamente dentre as linhas existentes

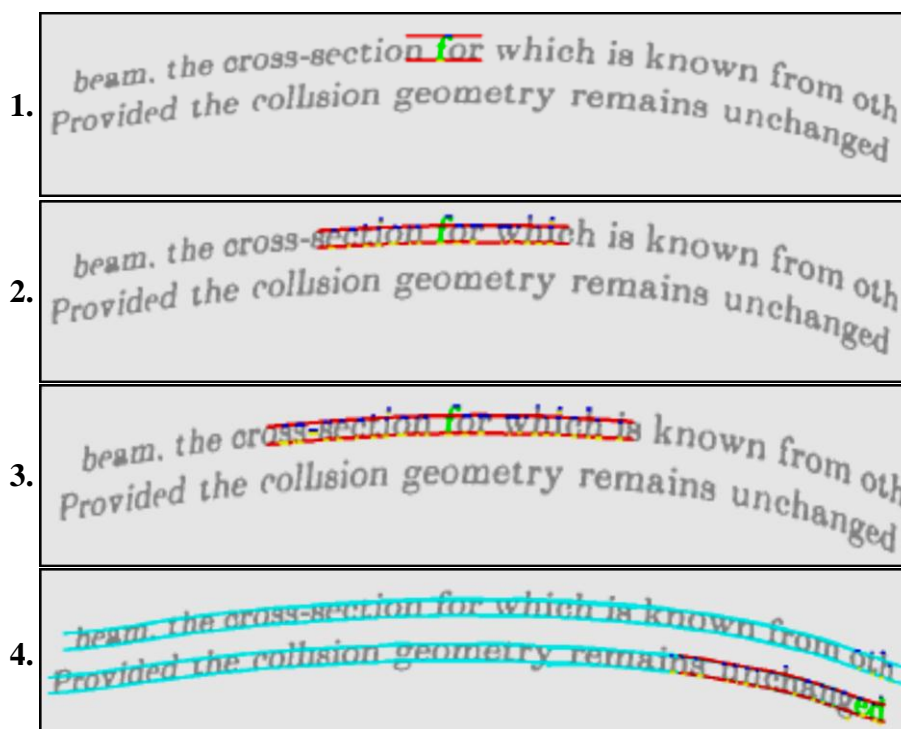
<sup>3</sup> Medida-F – unifica precisão e cobertura pela fórmula  $2 \times \frac{\text{precisão} \times \text{cobertura}}{\text{precisão} + \text{cobertura}}$

esse último defeito, os caracteres possuem tamanho e perspectiva variáveis. Tais características exigem um mecanismo de segmentação de linha de texto mais complexo.

Existem várias propostas deste tipo de segmentação na literatura técnica para documentos obtidos com câmeras digitais. Como exemplo pode-se citar o trabalho (MASALOVITCH e MESTETSKIY, 2007) que usa diagramas de Voronoi (BERG *et al.*, 2008) dos caracteres para formar uma linha de texto. Já Ulges e seus colaboradores (ULGES *et al.*, 2005) extraem linhas de texto com a análise da distância mínima e sobreposição entre os retângulos formados pelos componentes conectados.

A referência (BUKHARI *et al.*, 2008) usa dilatação morfológica para agrupar as letras em palavras. Uma vez agrupadas, o método de segmentação de *Active Contour* ou *Snakes* (KASS *et al.*, 1988), que serve para identificar bordas, é aplicado. Uma serpente (*Snake*) é inicializada por um conjunto de segmentos. Estes segmentos são deformados por “forças” internas e externas visando que a serpente se ajuste aos contornos das bordas iterativamente pela minimização de uma função de erro. Nesse caso, as forças são representadas pelo fluxo vetorial de gradiente (do inglês *Gradient Vector Flow*).

A outra proposta deste grupo é feita em (BUKHARI *et al.*, 2009a) utilizando serpentes acopladas (do inglês *coupled snakes*). Para uma determinada linha, duas serpentes são utilizadas para segmentar os contornos superiores e inferiores dos caracteres; as duas serpentes estão acopladas pelo mesmo valor de erro. O peso do erro dos pontos superiores é o dobro dos pontos inferiores, o que torna o algoritmo dependente da orientação do documento. A Figura 2.1 ilustra a formação de linhas através desse algoritmo, iniciando-se por uma letra “F”.

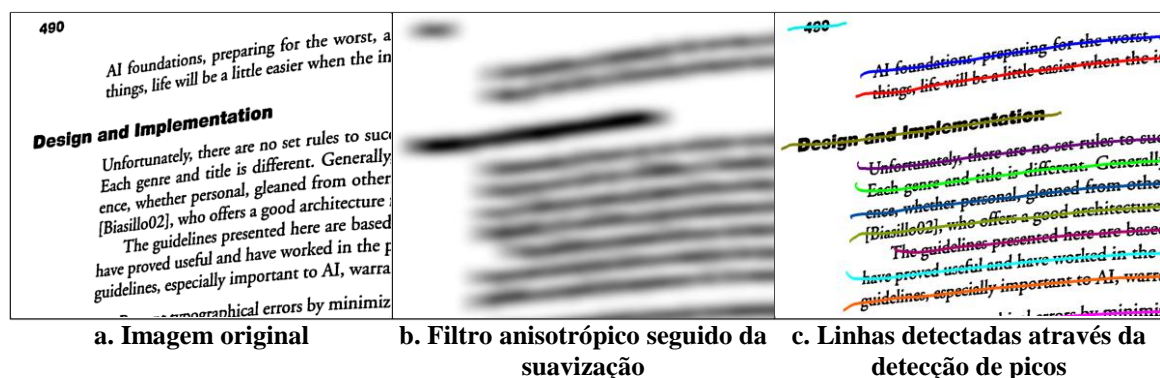


**Figura 2.1** Expansão da linha utilizando serpentes acopladas  
(BUKHARI *et al.*, 2009a)

Os mesmos autores em (BUKHARI *et al.*, 2009b; BUKHARI *et al.*, 2009c) identificam as linhas de texto em imagens com tons de cinza. Para cada *pixel* aplica-se um conjunto de máscaras, dentre as aplicações a que tiver maior valor é utilizada como valor resultante. As máscaras de convolução são obtidas pelo filtro gaussiano anisotrópico, descrito na equação (2.1), variando-se os desvios padrões nas direção  $x$  e  $y$  ( $\sigma_x$  e  $\sigma_y$ , respectivamente), bem como a direção da gaussiana (valor do parâmetro  $\theta$ ). Os desvios padrões são variados devido ao texto possuir fontes de tamanhos diferentes. Cada linha de texto possui inclinações distintas, o que justifica a utilização de diferentes valores de  $\theta$ .

Após aplicar o banco de filtros em toda a imagem, o detector de picos (HORN, 1970; RILEY, 1987) é utilizado para identificar as linhas. Em (BUKHARI *et al.*, 2011) adiciona-se a aplicação de um filtro de suavização. A Figura 2.2 mostra os passos da segmentação (BUKHARI *et al.*, 2011) sobre uma imagem.

$$G(x, y, \sigma_x, \sigma_y, \theta) = \frac{1}{2\pi\sigma_x\sigma_y} \exp \left\{ -\frac{1}{2} \left[ \left( \frac{x \cos \theta + y \sin \theta}{\sigma_x} \right)^2 + \left( \frac{-x \sin \theta + y \cos \theta}{\sigma_y} \right)^2 \right] \right\}. \quad (2.1)$$



**Figura 2.2** Detecção de linhas de texto (BUKHARI *et al.*, 2011)

Observa-se que dentre os métodos apresentados, apenas o grupo de Bukhari é que utiliza medidas objetivas na base de imagens pública (SHAFAIT e BRUEUL, 2007) para viabilizar uma comparação justa entre os algoritmos de segmentação de texto.

Os resultados descritos nesta tese apresentam um algoritmo:

- relativamente rápido; 4s de processamento em média por imagem com uma implementação na linguagem Java;
- que não depende da orientação, para isso linhas horizontais e verticais são processadas sem distinção;
- que representa os componentes da imagem numa estrutura de dados para facilitar o processamento; isto só foi obtido no desenvolvimento da segunda proposta.

Nas próximas seções deste capítulo serão apresentadas duas propostas de segmentação, sendo a segunda uma evolução da primeira.

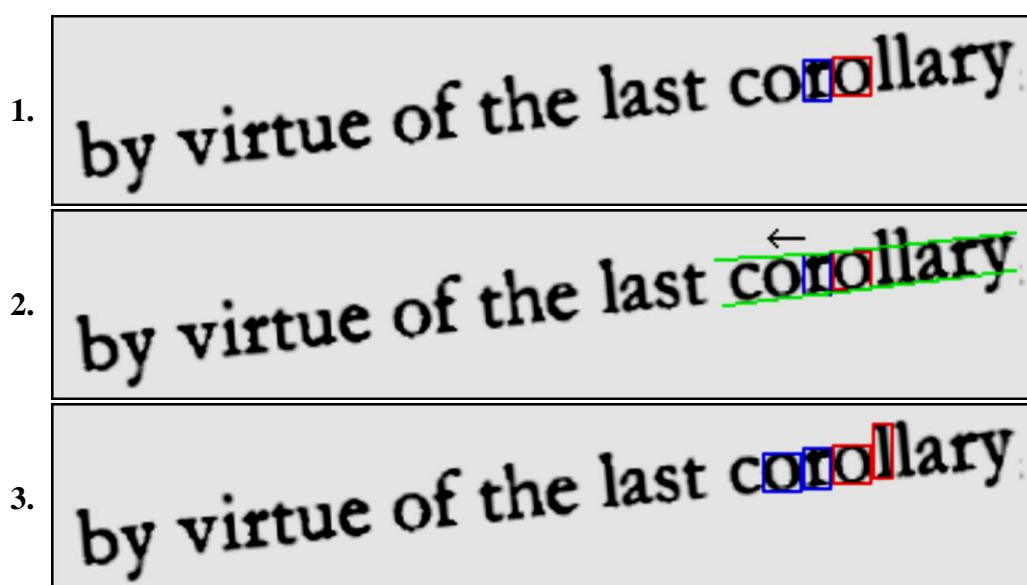
## 2.2 Primeiro Algoritmo Proposto

Dentre os grupos de pesquisa dos algoritmos descritos anteriormente, até a escrita desta tese, apenas o grupo *Image Understanding Pattern Recognition Research* da Universidade de Kaiserslautern apresentou medidas objetivas dos resultados de seus algoritmos (BUKHARI *et al.*, 2011; BUKHARI *et al.*, 2009a; BUKHARI *et al.*, 2008; BUKHARI *et al.*, 2009c; ULGES *et al.*, 2005) e outros (O’GORMAN, 1993; OLIVEIRA *et al.*, 2010a) na base de dados do CBDAR 2007 (BUKAHRI *et al.*, 2009a; BUKHARI *et al.*, 2010). Sendo (O’GORMAN, 1993) um algoritmo de segmentação para imagens obtidas por *scanner* e (OLIVEIRA *et al.*, 2010a) o primeiro algoritmo proposto que será descrito logo

a seguir. O mesmo encontra-se no Apêndice A.1 desta tese e apresenta qualidade superior ao apresentado pelo grupo da Universidade de Kaiserslautern.

Esse algoritmo foi publicada no ICIAR 2010 (OLIVEIRA *et al.*, 2010a) e tem como entrada imagens binárias. O mesmo baseou-se na idéia descrita por Ávila e Lins (ÁVILA e LINS, 2005), onde a detecção da inclinação de documentos digitalizados por *scanner* é feita através da regressão linear dos pontos superior e inferior de cada letra, à medida que a mesma é adicionada na linha de texto.

A Figura 2.3 ilustra um exemplo de processamento feito por (ÁVILA e LINS, 2005), onde a linha de texto é formada a partir do par de letras “ro” (passo 1) na palavra “*corollary*”. As linhas superiores e inferiores estão indicadas pelos segmentos em verde em que a busca é realizada no sentido da direita para esquerda (passo 2). O passo 3 indica que um novo símbolo foi adicionado à linha atual.



**Figura 2.3.** Formação de linha de texto feita em (ÁVILA e LINS, 2005)

No caso de documentos fotografados, a estimativa das linhas inferiores e superiores utilizando regressões lineares independentes não é tão eficaz, pois o texto pode conter combinações em que os pontos inferiores e superiores não formam linhas paralelas (e.g. letra maiúscula seguida de minúscula como na Figura 2.4). Sem contar com o agravante das linhas de texto serem curvas devido à distorção geométrica conforme mostrado na introdução desta tese.





**Figura 2.4 Letra maiúscula seguida de minúscula**

Para contornar tal problema, uma nova regressão foi criada inspirada na idéia apresentada por (BUKHARI *et al.*, 2009a), onde é feita uma segmentação local de *snakelets* com as linhas superiores e inferiores através da minimização do erro compartilhado das duas estimativas. No caso desta tese, propõe-se uma alternativa mais simples e não-iterativa através de uma regressão que minimiza o erro descrito na equação (2.2). Em que  $m$  indica a inclinação das retas (que é a mesma, pois, são paralelas). Os valores de  $c_T$  e  $c_B$  são as constantes de desvio das retas, superior e inferior;  $(u_{Tk}, v_{Tk})$  e  $(u_{Bk}, v_{Bk})$  são os pontos superiores e inferiores presentes na imagem, respectivamente.

$$E = \sum_{k=1}^N (m u_{Tk} + c_T - v_{Tk})^2 + \sum_{k=1}^N (m u_{Bk} + c_B - v_{Bk})^2 \quad (2.2)$$

Logo abaixo encontra-se um resumo do funcionamento do algoritmo que utiliza a imagem binarizada do documento.

1. Identificar blocos de componentes conectados
2. Identificar blocos vizinhos e colocá-los numa lista de prioridade de acordo com a similaridade descrita na equação (2.3);  $L$  e  $H$  indicam, a largura e a altura, respectivamente, de cada bloco vizinho.

$$\text{similaridade}(A, B) = \frac{\min(L_A, L_B)}{\max(L_A, L_B)} + \frac{\min(H_A, H_B)}{\max(H_A, H_B)}. \quad (2.3)$$

3. Formar as linhas de texto a partir do esvaziamento da lista de prioridade
4. Mesclar linhas de texto vizinhas

A descrição completa encontra-se disponível no apêndice desta tese, que foi publicado no ICIAR 2010 (OLIVEIRA *et al.*, 2010). A Figura 2.5 representa uma imagem binária da base CBDAR 2007; a Figura 2.6 mostra as linhas identificadas pela primeira proposta sobre a Figura 2.5 com segmentos de reta em vermelho entre os pontos centrais de componentes vizinhos.

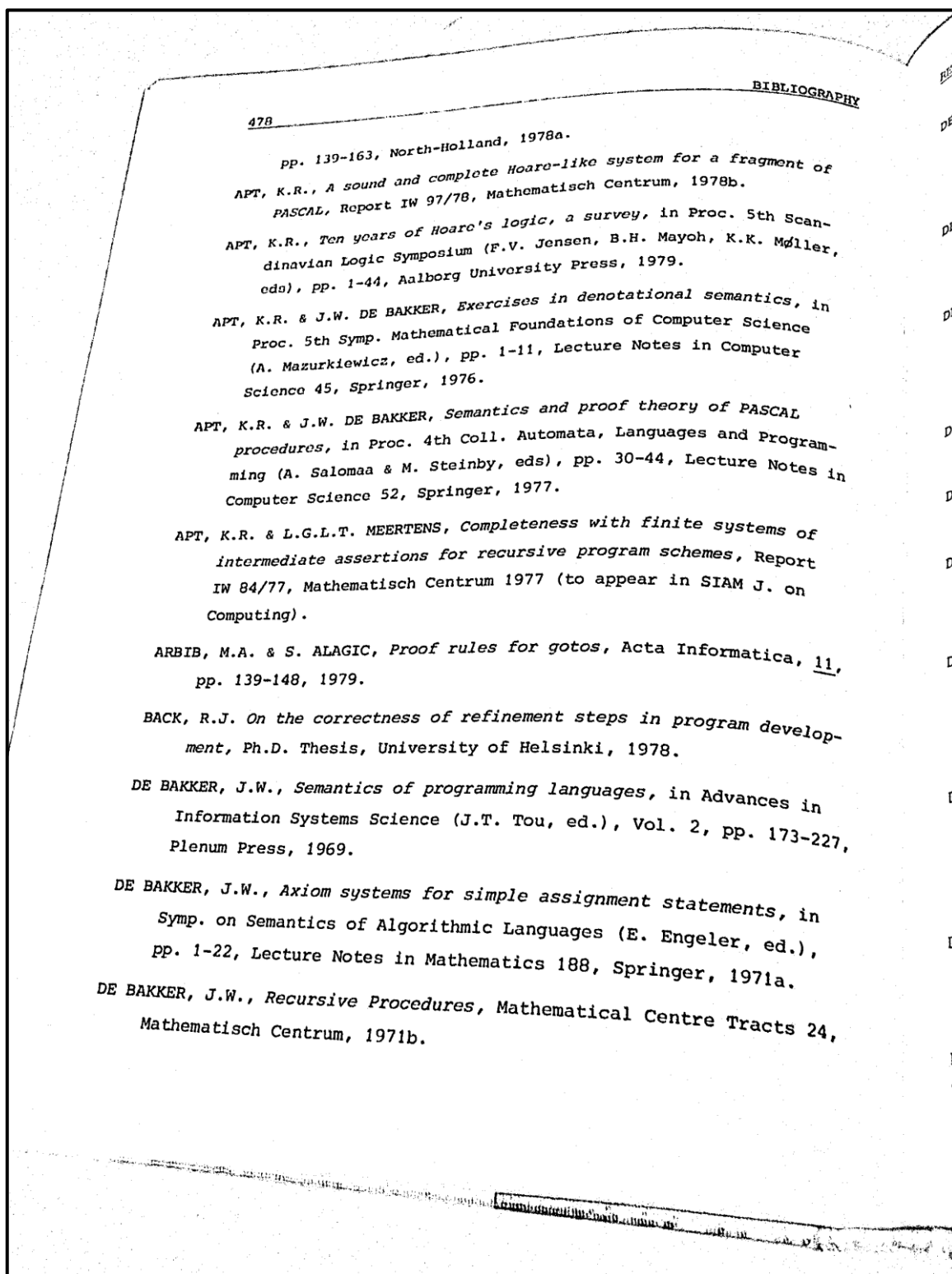


Figura 2.5 Imagem binária disponível na base CBDAR 2007

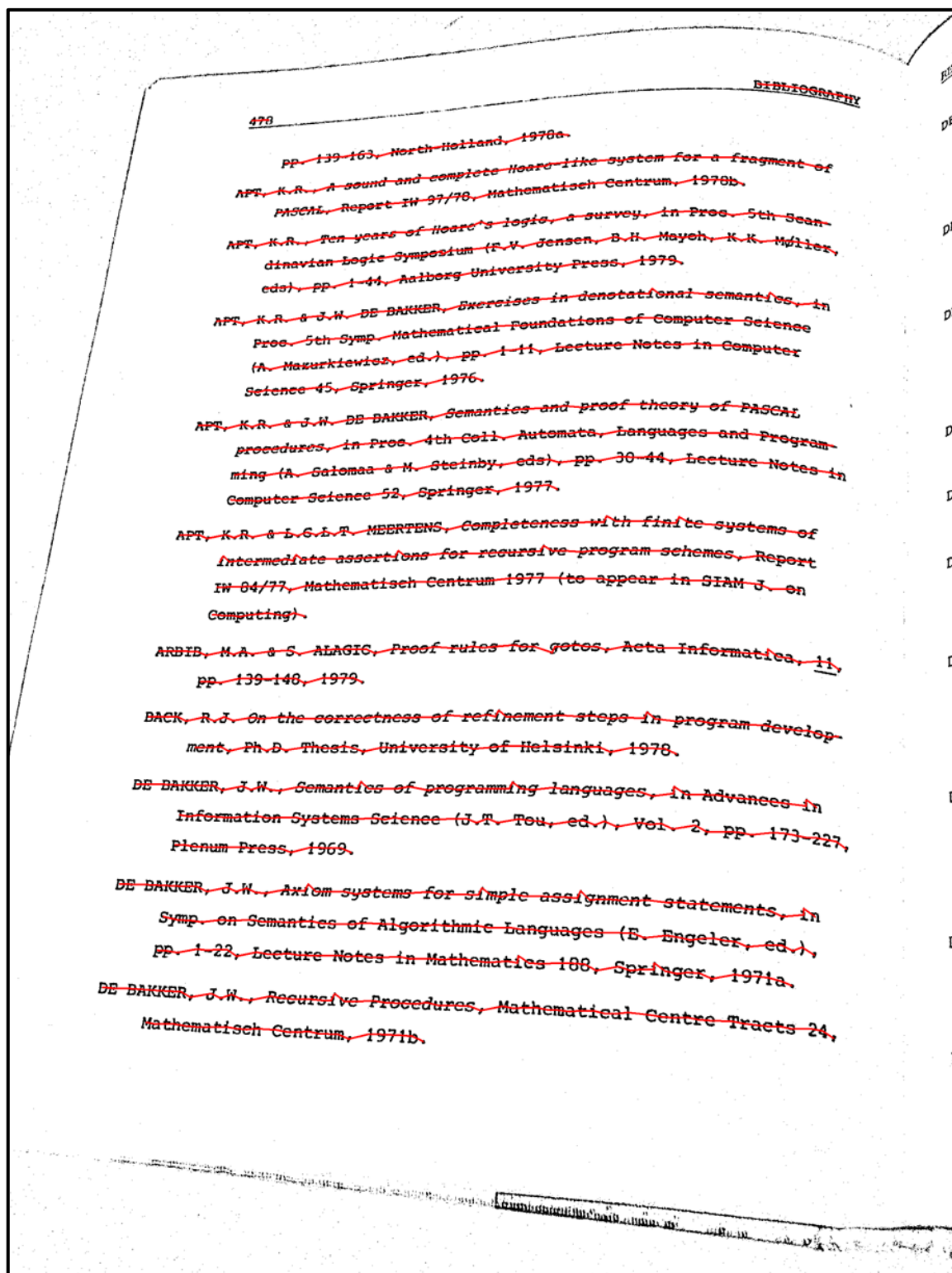


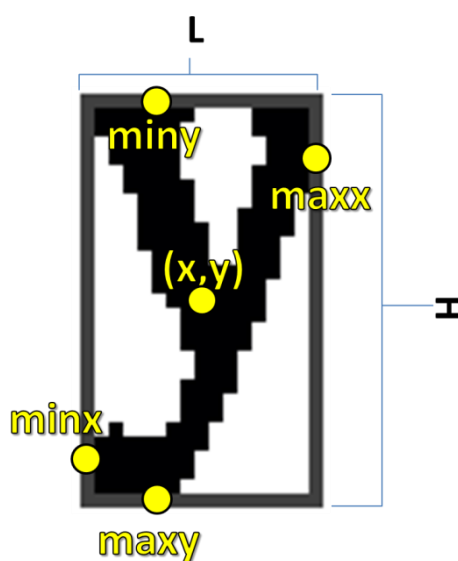
Figura 2.6 Linhas de texto detectadas da Figura 2.5 por (OLIVEIRA *et al.*, 2010a)

### 2.3 Segundo Algoritmo Proposto

A segunda proposta de segmentação de linha de texto é uma evolução do algoritmo anterior. Sua principal vantagem em relação ao anterior é devida ao uso de uma estrutura

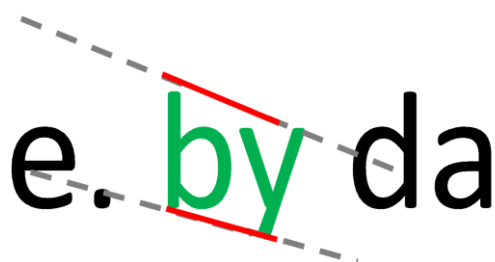
de dados no acesso às operações de busca de blocos. A estrutura de dados deve armazenar e buscar pontos bidimensionais de maneira eficiente. Também é necessário que a estrutura faça as buscas: pelos k-vizinhos de um ponto; pelos pontos dentro de um intervalo retangular. Exemplos de estruturas que atendem tais requisitos são a *kd-tree* e a *compressed quad-tree* (BERG *et al.*, 2008).

Uma vez carregada a imagem binária, os componentes conectados são detectados e armazenados na estrutura de dados multidimensional com as características apresentadas pela Figura 2.7 como: máximos e mínimos das coordenadas x/y; ponto central do retângulo definido conforme estes pontos limites.



**Figura 2.7 Informação obtida a partir dos componentes conectados**

A função de similaridade foi modificada com o fim de evitar o processamento prematuro de vizinhos com medidas semelhantes mas que não são um bom início para se formar uma linha (um exemplo são os vizinhos da preposição inglesa “by” na Figura 2.8). Portanto, incluíram-se os fatores que medem quão próximo de um quadrado são os dois blocos vizinhos. A equação (2.4) mostra esta nova medida, como o novo termo marcado em vermelho.



**Figura 2.8** Letras “b” e “y” possuem medidas semelhantes, porém não são um bom início para a formação de linha

$$\text{similaridade2}(A, B) = \frac{\min(L_A, L_B)}{\max(L_A, L_B)} + \frac{\min(H_A, H_B)}{\max(H_A, H_B)} + \frac{\min(H_A, L_A)}{\max(H_A, L_A)} + \frac{\min(H_B, L_B)}{\max(H_B, L_B)} \quad (2.4)$$

Outra modificação foi a utilização de duas listas de prioridades. Uma armazena os vizinhos que possuem inclinação do segmento, formado pelos seus pontos centrais, mais próxima com a do eixo horizontal. A outra lista é formada pelos vizinhos com essa inclinação mais semelhante com a linha vertical. Ambas são processadas, iniciando-se pela que tiver maior quantidade de elementos.

### 2.3.1 Estudo tipográfico sobre a forma dos caracteres

Para melhorar o desempenho do algoritmo de segmentação, estudos estatísticos sobre os elementos tipográficos dos caracteres foram realizados. A Figura 2.9 mostra as linhas tipográficas: ascendente, descendente, mediana e linha base. A altura *x-height* é a região compreendida entre as linhas mediana e base. O comprimento desta região pode ser obtida pela distância entre linhas paralelas obtidas pela regressão acoplada.



**Figura 2.9.** Linhas tipográficas descritas em (NAYLOR, 2010)

O tamanho da *x-height* varia de acordo com o tipo de fonte. Como não se sabe o tipo de fonte no documento a ser processado, foi medido o valor da *x-height* relativo ao tamanho dos tipos de fontes mais amplamente utilizados na atualidade. Os limites recomendados para várias medidas (e.g. espaço entre letras, colunas, palavras, etc.) são definidos em relação à 1em (BRINGHURST, 2004), que indica o tamanho da fonte. Relacionaram-se

estas medidas com a *x-height*, conforme ilustrado na Tabela 2.1, em que a primeira coluna apresenta o nome da fonte com seu tipo definido pela respectiva fonte. As fontes Impact e Courier, devido as suas medidas destoarem das demais, possuem menor e maior valores de *1em* relativos à *x-height*, respectivamente. Nota-se que há uma intersecção entre os limites entre-palavras e entre-letras, tendo um certo grau de incerteza para indicar se duas letras pertencem à mesma palavra ou não. O mesmo não acontece nos limites entre-palavras e entre-colunas, portanto é possível definir com mais precisão se dois caracteres pertencem à mesma linha ou não.

**Tabela 2.1 Medidas tipográficas relativas a *x-height***

Fonte	1em	Limite entre-letras (0,2em)	Limite entre-palavras (0,25em)	Limite entre-colunas (1,5em)
Times	251,00%	50,20%	62,75%	376,51%
Georgia	237,25%	47,45%	59,31%	355,87%
Arial	219,64%	43,93%	54,91%	329,45%
Tahoma	224,11%	44,82%	56,03%	336,17%
Courier	263,05%	52,61%	65,76%	394,57%
<b>Impact</b>	185,51%	37,10%	46,38%	278,27%
<b>Min</b>	<b>185,51%</b>	<b>37,10%</b>	<b>46,38%</b>	<b>278,27%</b>
<b>Max</b>	<b>263,05%</b>	<b>52,61%</b>	<b>65,76%</b>	<b>394,57%</b>

Outro aspecto é relativo à projeção dos caracteres em documentos impressos fotografados poder ser modelada como ortográfica, pois o tamanho dos símbolos gráficos são bem menores que a distância entre a câmera e o papel. Com base nesta modelagem, observou-se que a proporção entre altura e largura do caractere é modificada de acordo com os ângulos de inclinação (em relação aos eixos *x* e *y*), conforme a equação (2.5), onde:  $\frac{\Delta x}{\Delta y}$  indica a razão entre a altura e a largura do caractere, quando o plano do mesmo é paralelo ao plano da camera;  $\theta_x$  e  $\theta_y$  os ângulos de inclinação.

$$\text{proporção do caractere projetado} = \frac{\Delta x}{\Delta y} \times \frac{\cos \theta_y}{\cos \theta_x} \quad (2.5)$$

Por fim, um estudo inédito sobre as estatísticas de pares de símbolos de acordo com sua forma (conforme a Tabela 2.3) é apresentado na Tabela 2.2; a estatística é referente a alguns textos clássicos (e.g. A Volta no Mundo em 80 dias, Leviatã, Hamlet etc.). Cada célula desta tabela indica o percentual de um par de símbolos cujo caractere inicial possui a forma descrita na primeira coluna correspondente seguido de um símbolo da linha superior

da célula em questão; a última coluna indica os percentuais de caracteres com a forma descrita na primeira coluna.

Nesta tabela é possível observar que símbolos com alturas iguais a *x-height* são mais frequentes. Isto é esperado, pois a altura da maioria dos caracteres em textos impressos, como esta tese, é igual a *x-height*. No entanto os pares de caracteres com altura igual a *x-height* representam apenas 29,49%, sendo este valor menor que o total de todas as combinações de ascendentes, descendentes e *x-height* (retirando-se a combinação *x-height/x-height*) que é 36,77%. Este fato justifica a necessidade da regressão acoplada para “compensar” estas misturas no cálculo da *x-height* de maneira automática.

O percentual de 2,49% para caracteres “inferiores” tem como causa a existência de páginas de índice em algumas obras, pois essas páginas possuem muitos “pontos finais” seguidos (i.e. o símbolo “.”). Outro aspecto observado é relativo ao percentual de espaço ser igual a 14,82%, o que indica que há em média 5,75 símbolos visíveis a cada espaço; o autor desta tese não encontrou nenhuma referência confiável relativa ao tamanho médio das palavras para comprovar se este valor está coerente.

**Tabela 2.2 Percentuais de pares de caracteres conforme sua altura tipográfica**

Próximo Entrada	Asc.	Desc.	<i>x-height</i>	Grande	Inferior	Superior	Espaço	Outros	Total
<b>Asc.</b>	5,39%	0,73%	13,19%	0,02%	0,73%	0,03%	4,20%	0,15%	<b>24,44%</b>
<b>Desc.</b>	0,57%	0,12%	2,37%	0,00%	0,26%	0,01%	1,04%	0,06%	<b>4,43%</b>
<b><i>x-height</i></b>	11,71%	2,69%	29,49%	0,02%	1,35%	0,07%	7,11%	0,30%	<b>52,75%</b>
<b>Grande</b>	0,04%	0,00%	0,01%	0,00%	0,01%	0,00%	0,02%	0,00%	<b>0,09%</b>
<b>Inferior</b>	0,41%	0,02%	0,23%	0,02%	0,12%	0,11%	1,55%	0,02%	<b>2,49%</b>
<b>Superior</b>	0,17%	0,00%	0,08%	0,01%	0,00%	0,03%	0,07%	0,00%	<b>0,37%</b>
<b>Espaço</b>	6,00%	0,84%	7,26%	0,02%	0,01%	0,07%	0,60%	0,02%	<b>14,82%</b>
<b>Outros</b>	0,13%	0,02%	0,10%	0,00%	0,00%	0,05%	0,23%	0,05%	<b>0,60%</b>





### 2.3.2 Resumo da segunda proposta

Um resumo do presente algoritmo encontra-se logo abaixo. A descrição completa encontra-se disponível no Apêndice, que foi publicado no *ACM Symposium on Applied Computing 2011* (OLIVEIRA *et al.*, 2011a). A descrição também está disponível em (OLIVEIRA *et al.*, 2011b) que contém a otimização dos parâmetros e a inclusão de precisão e medida-F do presente algoritmo e dos outros.

1. Identificar componentes conectados e adicionar na estrutura multi-dimensional;
2. Encontrar os k-vizinhos de todos os pontos da estrutura multi-dimensional;
3. Processar k-vizinhos e colocar cada um na lista de prioridade correspondente;
4. Formar linhas de texto a partir das listas de prioridades, começando pela que tiver mais elementos;
5. Identificar colunas de texto;
6. Mesclar linhas de texto que estiverem na mesma coluna.

Um exemplo de resultado pode ser visto na Figura 2.10 obtido a partir da Figura 2.5; as linhas de texto detectadas estão representadas pelos segmentos em vermelho entre os componentes vizinhos.

- BIBLIOGRAPHY
- 478
- ~~pp. 139-163, North-Holland, 1978a.~~
- ~~APT, K.R., A sound and complete Hoare-like system for a fragment of PASCAL, Report IW 97/78, Mathematisch Centrum, 1978b.~~
- ~~APT, K.R., Ten years of Hoare's logic, a survey, in Proc. 5th Scandinavian Logic Symposium (F.V. Jensen, B.H. Mayoh, K.K. Müller, eds), pp. 1-44, Aalborg University Press, 1979.~~
- ~~APT, K.R. & J.W. DE BAKKER, Exercises in denotational semantics, in Proc. 5th Symp. Mathematical Foundations of Computer Science (A. Mazurkiewicz, ed.), pp. 1-11, Lecture Notes in Computer Science 45, Springer, 1976.~~
- ~~APT, K.R. & J.W. DE BAKKER, Semantics and proof theory of PASCAL procedures, in Proc. 4th Coll. Automata, Languages and Programming (A. Salomaa & M. Steinby, eds), pp. 30-44, Lecture Notes in Computer Science 52, Springer, 1977.~~
- ~~APT, K.R. & L.G.L.T. MEERTENS, Completeness with finite systems of intermediate assertions for recursive program schemes, Report IW 84/77, Mathematisch Centrum 1977 (to appear in SIAM J. on Computing).~~
- ~~ARBIB, M.A. & S. ALAGIC, Proof rules for gotos, Acta Informatica, 11, pp. 139-148, 1979.~~
- ~~BACK, R.J., On the correctness of refinement steps in program development, Ph.D. Thesis, University of Helsinki, 1978.~~
- ~~DE BAKKER, J.W., Semantics of programming languages, in Advances in Information Systems Science (J.T. Fou, ed.), Vol. 2, pp. 173-227, Plenum Press, 1969.~~
- ~~DE BAKKER, J.W., Axiom systems for simple assignment statements, in Symp. on Semantics of Algorithmic Languages (E. Engeler, ed.), pp. 1-22, Lecture Notes in Mathematics 188, Springer, 1971a.~~
- ~~DE BAKKER, J.W., Recursive Procedures, Mathematical Centre Tracts 24, Mathematisch Centrum, 1971b.~~

Figura 2.10 Ilustração da identificação de linhas pela segmentação (OLIVEIRA *et al.*, 2011)

## 2.4 Resultados

Bukhari, Shafait e Breuel em (BUKHARI *et al.*, 2011) apresentam um novo método de segmentação de linhas de texto e fazem uma comparação dos algoritmos (ULGES *et al.*, 2005; BUKHARI *et al.*, 2008; BUKHARI *et al.*, 2009a; BUKHARI *et al.*, 2009c; O’GORMAN, 1993; OLIVEIRA *et al.*, 2010a), lembrando que (O’GORMAN, 1993) é uma segmentação para documentos digitalizados por *scanner* e que (OLIVEIRA *et al.*, 2010a) é a primeira proposta apresentada nesta tese. A base de dados utilizada na comparação é a da competição de correção geométrica do CBDAR 2007 (SHAFAIT e BREUEL, 2007). Esta base contém três tipos de imagens:

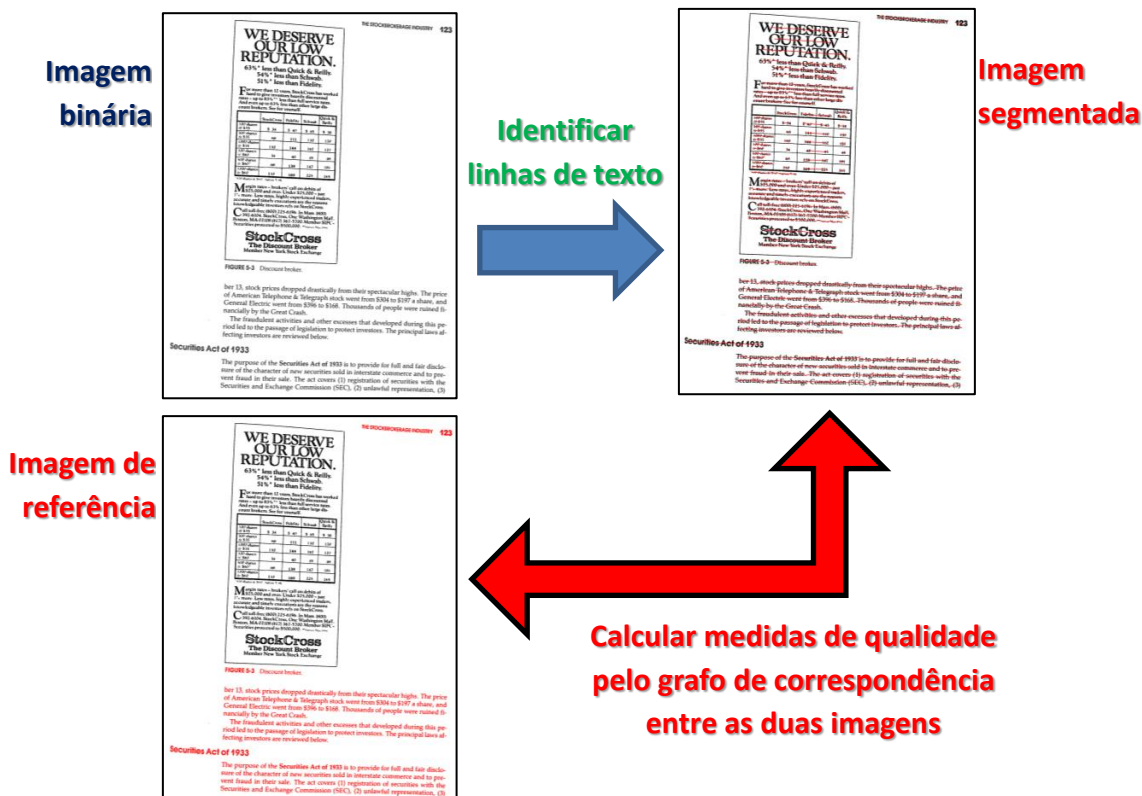
- uma capturada por câmera digital portátil em escalas de cinza;
- outra binária obtida através da binarização das imagens capturadas;
- e uma colorida que é resultado da atribuição de uma cor para cada linha na imagem binária; esta será chamada aqui de referência.

Para comparar os resultados, os algoritmos de segmentação devem ter como entrada a imagem binária ou de intensidade da base do CBDAR 2007 e gerar a imagem colorida no mesmo formato da imagem de referência.

Para obter as medidas de desempenho, cria-se um grafo de correspondência entre os *pixels* da imagem resultante e a da base; cada nó representa uma cor que indica a linha de texto de cada imagem. As arestas do grafo representam que linhas de texto das duas imagens têm *pixels* em comum; o peso da aresta indica a quantidade dos mesmos. As cores branco e preto denotam, respectivamente, fundo (parte não impressa do documento) e ruído (desenhos, gráficos, etc.). Uma aresta é dita significativa se  $w_i/P \geq Tr$  e  $w_i \geq Ta$ , onde:

- $w_i$  indica o peso da aresta;
- $P$  é a quantidade total de *pixels* numa linha de texto;
- $w_i/P$  indica a proporção de *pixels* em comum entre as linhas na imagem de referência e processada;
- $Tr$  e  $Ta$  são limites relativo e absoluto para os pesos, respectivamente.

A Figura 2.11 mostra o processo de obtenção das medidas de qualidade dos algoritmos de segmentação de linhas de texto descrito nos parágrafos anteriores.

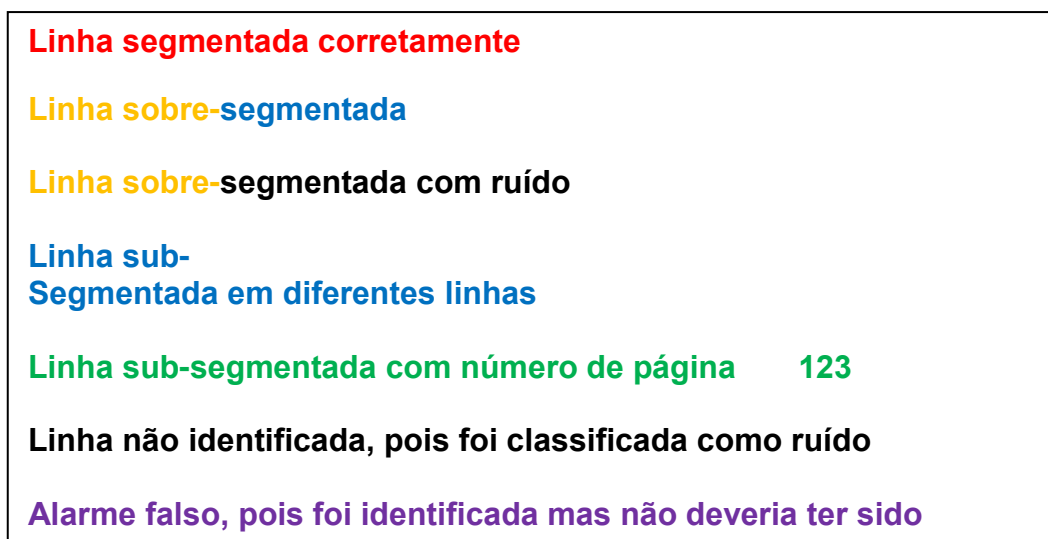


**Figura 2.11** Obtenção das medidas de qualidade para segmentações de linhas de texto

Com base nesses dados, as medidas abaixo são calculadas (BUKHARI *et al.*, 2010a); a Figura 2.12 mostra um exemplo dessas medidas.

- Número de linhas de referência ( $N_g$ ) – número total de linhas de referência em toda base de dados.
- Número de linhas segmentadas ( $N_s$ ) – número total de linhas segmentadas pelo algoritmo a ser avaliado que não são falso-positivas;
- Número de linhas segmentadas corretamente ( $N_{o2o}$ ) – quantidade total de arestas que são a única ligação entre uma linha segmentada e uma linha referência.
- Número de sobre segmentações ( $N_{os}$ ) – total de arestas significativas que linhas de referência possuem menos o número de linhas de referência.
- Número de sub-segmentações ( $N_{us}$ ) – total de arestas significativas que linhas segmentadas possuem menos o número de linhas segmentadas.
- Número de linhas sobre-segmentadas ( $N_{oc}$ ) – total de linhas de referência com mais de uma aresta significativa.
- Número de linhas sub-segmentadas ( $N_{uc}$ ) – total de linhas segmentadas com mais de uma aresta significativa.

- Número de linhas perdidas ( $N_{mc}$ ) – total de linhas de referência que não possuem uma aresta que liga às linhas segmentadas.
- Número de alarmes falsos ( $N_{fa}$ ) – total de linhas segmentadas que não possuem uma aresta que liga as linhas de referência.
- Percentual de linhas segmentadas corretamente ( $P_{o2o}$ ) –  $N_{o2o}/N_g$
- Percentual de linhas sobre-segmentadas ( $P_{oc}$ ) –  $N_{oc}/N_g$
- Percentual de linhas sub-segmentadas ( $P_{uc}$ ) –  $N_{uc}/N_g$
- Percentual de linhas perdidas ( $P_{mc}$ ) –  $N_{mc}/N_g$



**Figura 2.12 Ilustração das medidas usadas na comparação da segmentação: ruídos em preto puro; linhas detectadas em cores**

Além dessas medidas, propõe-se a adição neste contexto de três novas medidas já conhecidas da área de reconhecimento de padrões (MAKHOUL *et al.*, 2010): precisão, cobertura e medida-f. As suas definições encontram-se nas equações (2.6)-(2.8).

$$PR = \text{precisão} = \frac{\text{linhas corretas } (N_{o2o})}{\text{linhas segmentadas } (N_s + N_{fa})} \quad (2.6)$$

$$RE = \text{cobertura} = \frac{\text{linhas corretas } (N_{o2o})}{\text{linhas de referência } (N_g)} \quad (2.7)$$

$$FM = \text{medida - f} = \frac{1}{\left(\frac{0,5}{PR} + \frac{0,5}{RE}\right)} = 2 \times \frac{PR \times RE}{(PR + RE)} \quad (2.8)$$

Onde:

- o número total de linhas corretas é  $N_{o2o}$ , que é igual a  $P_{o2o} \times N_g$ ;
- o total de linhas segmentadas é  $N_s$  somado com  $N_{fa}$  (i.e.  $N_s + N_{fa}$ ), pois  $N_s$  é definido como “número total de linhas segmentadas pelo algoritmo a ser avaliado **que não são falso positivas (i.e. alarme falso)**”

A medida-f é a média harmônica entre precisão e cobertura. Este tipo de média é mais apropriado na avaliação dos algoritmos de segmentação, pois o valor resultante tende ao mínimo das duas medidas, o que não acontece na média aritmética, por exemplo.

A Tabela 2.5 e a Tabela 2.6 mostram os resultados com as medidas de (BUKHARI *et al.*, 2010a) e as três novas medidas, respectivamente, obtidas pelos algoritmos propostos bem como outros métodos de segmentação apresentados no início deste capítulo (BUKHARI *et al.*, 2010a; BUKHARI *et al.*, 2010b) onde:

- Seg2 RE denota a segunda segmentação com melhor cobertura (OLIVEIRA *et al.*, 2011);
- Seg2 FM denota a segunda segmentação com melhor medida-F (OLIVEIRA *et al.*, 2011);
- Seg1 significa a primeira segmentação (OLIVEIRA *et al.*, 2010a);
- R(G) e R(B) denota a segmentação (BUKHARI *et al.*, 2009c) que tiveram imagens em escalas de cinza e binárias como entrada, respectivamente;
- B-S indica *baby snakes* (BUKHARI *et al.*, 2008);
- C-S denota *coupled snakes* (BUKHARI *et al.*, 2009a);
- C-S v2 representa a versão estendida de serpentes acopladas não publicada, mas está em fase de revisão para a revista IJDAR (BUKHARI *et al.*, 2010a);
- RLFB (G) e RLFB (B) denotam a segmentação (BUKHARI *et al.*, 2011) que tiveram imagens em escalas de cinza e binárias como entrada, respectivamente;
- N-D indica o algoritmo baseado na distância entre vizinhos (ULGES *et al.*, 2005);
- Doctsrum denota o algoritmo Doctsrum (O’GORMAN, 1993), que foi desenvolvido para imagens escaneadas sem distorção geométrica.

É importante informar que o autor desta tese detectou um erro de 0,5% nas linhas de referências gravadas pelos promotores de competição (BUKHARI *et al.*, 2010a; BUKHARI *et al.*, 2010b). Portanto, os percentuais  $P_{oc}$  e  $P_{uc}$  dos outros algoritmos podem variar em torno desse valor. O valor de  $N_g$  é 3091 para outros; isto está explícito na Tabela 2.5. No caso da segunda proposta, este valor é igual a 3103.

**Tabela 2.5 Medidas de qualidade dos algoritmos de segmentação de linhas de texto**

Med. Algo.	N <sub>s</sub>	P <sub>o2o</sub>	P <sub>oc</sub>	P <sub>uc</sub>	P <sub>mc</sub>	N <sub>os</sub>	N <sub>us</sub>	N <sub>fa</sub>	N <sub>s</sub> +N <sub>fa</sub>
Seg2 RE	3095	<b>97,84%</b>	1,77%	<b>0,19%</b>	<b>0,90%</b>	57	<b>7</b>	735	<b>3830</b>
Seg2 FM	3072	97,33%	<b>1,26%</b>	0,61%	0,93%	<b>39</b>	21	<b>645</b>	<b>3717</b>
Seg1	2924	91,10%	21,71%	1,81%	4,43%	682	57	785	<b>3709</b>
N-D	3134	93,47%	3,33%	0,55%	2,59%	134	<b>18</b>	2491	5625
R (G)	3045	89,10%	3,53%	3,85%	0,91%	115	131	1186	4231
R (B)	3115	89,65%	4,40%	3,30%	0,29%	144	110	2183	5298
C-Sv2	3093	<b>95,21%</b>	1,68%	1,59%	<b>0%</b>	54	51	3277	<b>6370</b>
C-S	2799	78,26%	<b>1,26%</b>	9,06%	<b>0%</b>	<b>39</b>	359	3251	<b>6050</b>
B-S	3371	87,58%	5,79%	2,91%	<b>0%</b>	294	117	13199	<b>16570</b>
Doctsrum	6852	21,26%	66,90%	51,50%	<b>0%</b>	4383	2096	6066	<b>12918</b>
RLFB (B)	3204	93,63%	3,66%	1,49%	0,71%	194	59	2062	5266
RLFB (G)	3133	92,75%	3,98%	1,81%	0,78%	136	70	1826	4959

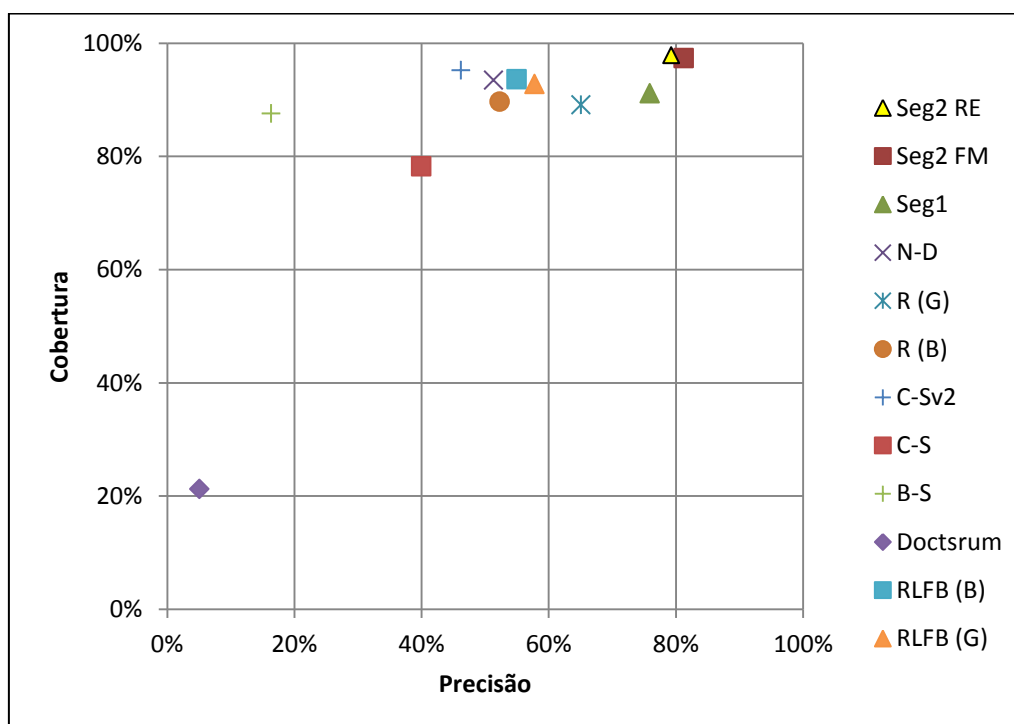
**Tabela 2.6 Outras medidas sugeridas de qualidade dos algoritmos**

Med. Algo.	N <sub>o2o</sub>	N <sub>g</sub>	N <sub>s</sub> +N <sub>fa</sub>	PR	RE	FM
Seg2 RE	3036,00	3103	3830	79,27%	<b>97,84%</b>	87,58%
Seg2 FM	3020,00	3103	3717	<b>81,25%</b>	97,33%	<b>88,56%</b>
Seg1	2815,90	3091	3709	75,92%	91,10%	82,82%
N-D	2889,16	3091	5625	51,36%	93,47%	66,30%
R (G)	2754,08	3091	4231	<b>65,09%</b>	89,10%	<b>75,23%</b>
R (B)	2771,08	3091	5298	52,30%	89,65%	66,06%
C-S v2	2942,94	3091	6370	<b>46,20%</b>	<b>95,21%</b>	62,21%
C-S	2419,02	3091	6050	<b>39,98%</b>	78,26%	52,93%
B-S	2707,10	3091	16570	<b>16,34%</b>	87,58%	27,54%
Doctsrum	657,15	3091	12918	<b>5,09%</b>	21,26%	8,21%
RLFB (B)	2894,10	3091	5266	54,96%	93,63%	69,26%
RLFB (G)	2866,90	3091	4959	57,81%	92,75%	71,23%

### 2.4.1 Comparação das medidas de qualidade

A Tabela 2.5 e a Tabela 2.6 mostram diversas medidas de qualidade, dentre estas, Bukhari e seus colegas (BUKHARI *et al.*, 2010a; BUKHARI *et al.*, 2010b; BUKHARI *et al.*, 2011) indicam que a medida principal é a cobertura (*RE*). Esta tese argumenta que a medida-f (*FM*) deva ser a medida principal, pois a cobertura não inclui os alarmes falsos ( $N_{fa}$ ) na sua definição. Neste caso, a primeira (*Seg1*) e segunda (*Seg2 RE* e *Seg2 FM*) proposta apresentam melhor desempenho se comparada com os outros algoritmos na medida-f. O mesmo é observado para a medida de precisão (*PR*).

O desempenho dos dois algoritmos propostos pode ser mais bem visualizado na Figura 2.13, que contém um gráfico onde cada ponto possui suas coordenadas definidas pela precisão e cobertura do algoritmo correspondente. Observa-se que os pontos da segunda segmentação estão bem mais próximos do ponto ideal (i.e. precisão e cobertura iguais a 100%) que os demais. Este fato reforça que a segunda proposta de segmentação é a melhor dentre os outros algoritmos.



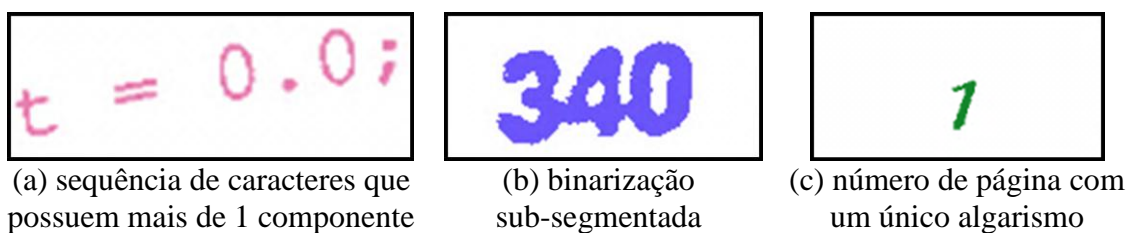
**Figura 2.13** Gráfico das medidas de Cobertura vs Precisão das segmentações

Na Figura 2.13, retirando-se os pontos da segunda segmentação, a primeira proposta está mais próxima do ponto ideal que os outros métodos, bem como um maior valor de



medida-f presente na Tabela 2.6. Portanto, houve uma evolução na segunda proposta em relação a primeira.

No tocante às outras medidas, a segunda proposta tem melhor desempenho no quesito de sobre-segmentação ( $P_{oc}$  e  $N_{os}$ ) e quantidade de linhas corretas ( $P_{o2o}$ ), sem o registro de junção de linhas inferiores e superiores. As linhas perdidas da segunda proposta podem ser vistas na Figura 2.14 e constituem de: sequência de caracteres que possuem mais de 1 componente (a) (símbolos “=” e “;”); junção de caracteres pela binarização (b); caracteres isolados (c).



**Figura 2.14** Exemplos de linhas perdidas

Outra vantagem de ambas propostas é a baixa quantidade de alarmes falsos, se comparada com outros algoritmos na Tabela 2.6. Para exemplificar esse tipo de erro é exibida a Figura 2.15 extraída da base CBDAR 2007 com letras pretas e vermelhas indicando ruído e linhas de texto, respectivamente. A detecção feita pela segunda proposta sobre essa figura está disponível na Figura 2.16, na qual observa-se a detecção de linhas de texto em uma ilustração textual. Note-se que é difícil discernir os caracteres das ilustrações e dos existentes no texto principal.

Os outros algoritmos superam as segmentações propostas no percentual de linhas perdidas ( $P_{mc}$ ) – o que em alguns casos é de 0%. No entanto, esses baixos valores dos algoritmos são ofuscados pela alta quantidade de linhas segmentadas (i.e.  $N_{fa}+N_m$  em vermelho na Tabela 2.5), sendo em alguns casos o dobro de linhas existentes no conjunto de imagens do CBDAR 2007. Ou seja, mais da metade das linhas de texto foram identificadas incorretamente – vide os valores em vermelho na coluna PR na Tabela 2.6.

## WE DESERVE OUR LOW REPUTATION.

63%\* less than Quick & Reilly.  
54%\* less than Schwab.  
51%\* less than Fidelity.

For more than 12 years, StockCross has worked hard to give investors heavily discounted rates - up to 83%\*\* less than full service rates. And even up to 63% less than other large discount brokers. See for yourself.

	StockCross	Fidelity	Schwab	Quick & Reilly
100 shares @ \$35	\$ 34	\$ 40	\$ 45	\$ 38
500 shares @ \$35	68	111	110	130
1000 shares @ \$35	110	144	162	127
100 shares @ \$60	34	40	45	48
500 shares @ \$60*	68	139	147	181
1000 shares @ \$60	110	169	225	165

\*100 shares @ \$50. \*\*Surveys 5/94.  
Margin rates - brokers' call on debits of \$25,000 and over. Under \$25,000 - just 1% more. Low rates, highly experienced traders, accurate and timely executions are the reasons knowledgeable investors rely on StockCross.

Call toll-free (800) 225-6196. In Mass. (800) 392-6104. StockCross, One Washington Mall, Boston, MA 02108 (617) 367-5700. Member SIPC - Securities protected to \$500,000. \*\*Surveys Mar. 1994

**StockCross**  
The Discount Broker  
Member New York Stock Exchange

Ilustração textual

FIGURE 5-3 Discount broker.

ber 13, stock prices dropped drastically from their spectacular highs. The price of American Telephone & Telegraph stock went from \$304 to \$197 a share, and General Electric went from \$396 to \$168. Thousands of people were ruined financially by the Great Crash.

The fraudulent activities and other excesses that developed during this period led to the passage of legislation to protect investors. The principal laws affecting investors are reviewed below.

### Securities Act of 1933

The purpose of the **Securities Act of 1933** is to provide for full and fair disclosure of the character of new securities sold in interstate commerce and to prevent fraud in their sale. The act covers (1) registration of securities with the Securities and Exchange Commission (SEC), (2) unlawful representation, (3)

Figura 2.15 Imagem de referência obtida da base CBDAR 2007

company  
e combi  
fered by  
rms that

Nicolaus  
pus and  
aus is a  
size, it  
a larger  
n its re-

regional  
offices  
office.  
ervices.  
s. They  
nission  
scount

provide  
orders  
nat are  
station

e pub-  
l with  
nking  
s busi-  
argely  
public,  
ve the

nt by  
t the  
years  
nt. In  
very-  
15 a  
ocks  
ince  
f the  
ock  
em-

THE STOCKBROKERAGE INDUSTRY **123**

**~~WE DESERVE  
OUR LOW  
REPUTATION.~~**

63% <sup>^</sup> less than Quick & Reilly.  
54% <sup>^</sup> less than Schwab.  
51% <sup>^</sup> less than Fidelity.

For more than 12 years, StockCross has worked hard to give investors heavily discounted rates—up to 63% <sup>^</sup> less than full service rates. And even up to 63% <sup>^</sup> less than other large discount brokers. See for yourself.

	StockCross	Fidelity—Schwab	Quick & Reilly
100 shares @ \$75	\$ 34	\$ 40 — \$ 45	\$ 38
500 shares @ \$75	68	111 — 110	130
1000 shares @ \$75	110	144 — 162	127
100 shares @ \$60	34	40 — 45	46
500 shares @ \$60	68	139 — 147	181
1000 shares @ \$60	110	169 — 225	165

**M**argin rates—brokers' call on debits of \$25,000 and over—under \$25,000—just 1% more. Low rates, highly experienced traders, accurate and timely executions are the reasons knowledgeable investors rely on StockCross.

**C**all toll-free (800) 225-6196. In Mass. (800) 392-6104. StockCross, One Washington Mall, Boston, MA 02100 (617) 367-5700. Member SIPC. Securities protected to \$500,000.

**StockCross**  
The Discount Broker  
Member New York Stock Exchange

FIGURE 5-3 Discount broker.

ber 13, stock prices dropped drastically from their spectacular highs. The price of American Telephone & Telegraph stock went from \$304 to \$197 a share, and General Electric went from \$396 to \$168. Thousands of people were ruined financially by the Great Crash.

The fraudulent activities and other excesses that developed during this period led to the passage of legislation to protect investors. The principal laws affecting investors are reviewed below.

**Securities Act of 1933**

The purpose of the Securities Act of 1933 is to provide for full and fair disclosure of the character of new securities sold in interstate commerce and to prevent fraud in their sale. The act covers (1) registration of securities with the Securities and Exchange Commission (SEC), (2) unlawful representation, (3)

Linhas de texto detectadas na ilustração

Figura 2.16 Exemplo de alarme falso

Um avanço observado da primeira para a segunda proposta é a diminuição de sobre segmentação ( $P_{oc}$  e  $N_{os}$ ). Isto foi obtido através da inclusão de sinais de pontuação na segunda proposta. Um exemplo de processamento com valor de cobertura de 100% pode ser visto na Figura 2.17 com os segmentos de retas entre componentes vizinhos. Essa

figura foi processada em 0,950s num laptop Dell Inspiron 1525 rodando Windows 7 Professional com processador Celeron 2,0GHz e 3GB de RAM. O tempo médio de execução desse algoritmo é de 4 segundos. O método descrito em (BUKHARI *et al.*, 2011) leva 3,24 minutos rodando numa máquina com 40GB de memória RAM, processador 2,53 GHz (modelo não informado) e no sistema operacional Linux Ubuntu. A mesma referência informa que um método anterior do grupo (BUKHARI *et al.*, 2009b) leva 17 minutos no mesmo modelo de máquina. Portanto, os métodos propostos são bem mais rápidos que os algoritmos de Bukhari e seus colegas numa máquina com configuração inferior em termos de processador e memória.

Na Tabela 2.5 e Tabela 2.6 o método Docstrum apresenta desempenho bem inferior aos outros algoritmos. Este algoritmo foi desenvolvido para detectar linhas de texto em imagens escaneadas sem distorção geométrica. Como as imagens do CBDAR 2007 apresentam um cenário bem mais adverso, o algoritmo Docstrum não conseguiu obter um resultado que pudesse ser comparado com os demais métodos.

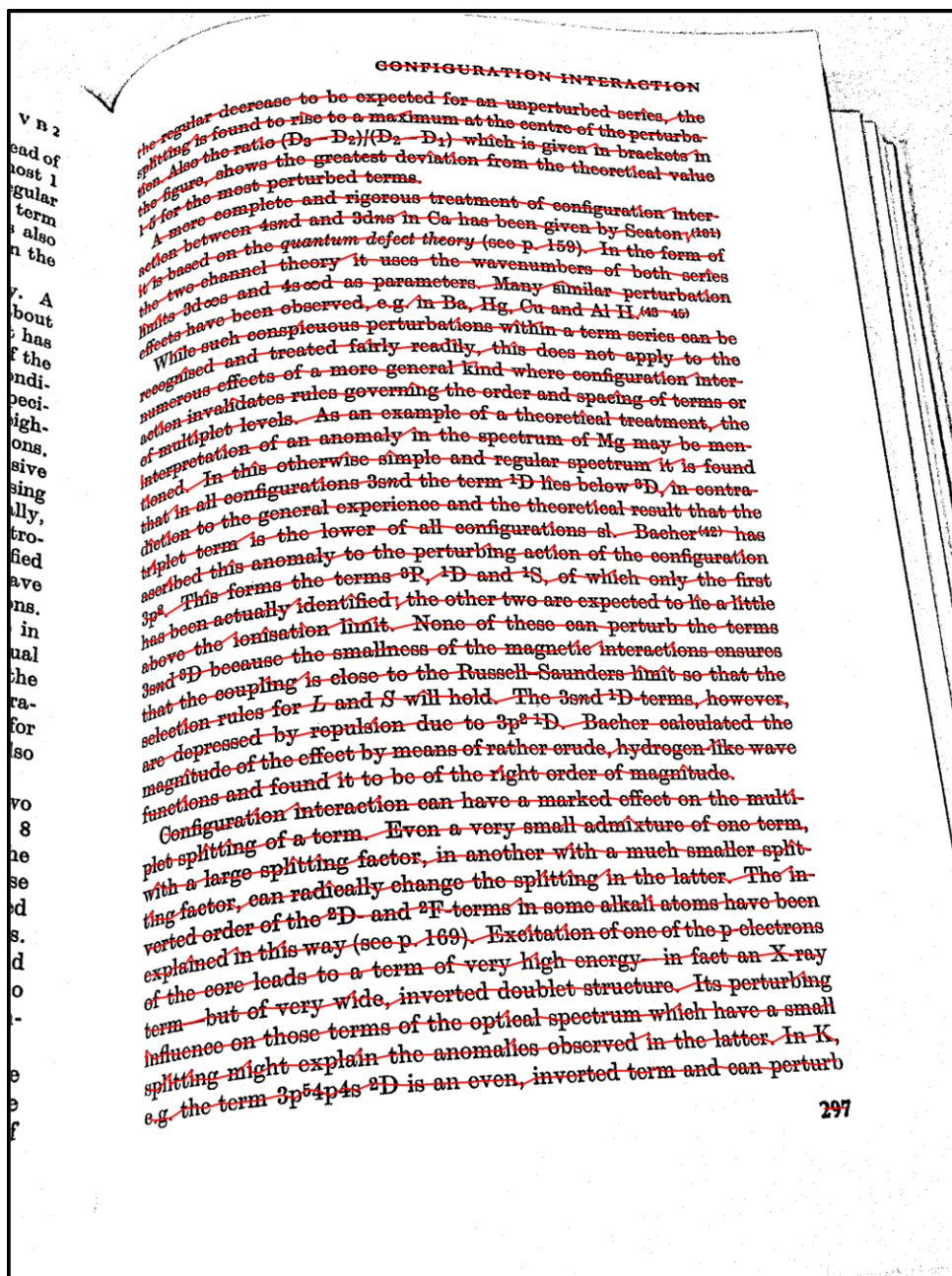


Figura 2.17 Exemplo de processamento feito com sucesso.

### 3 Correção de distorção geométrica

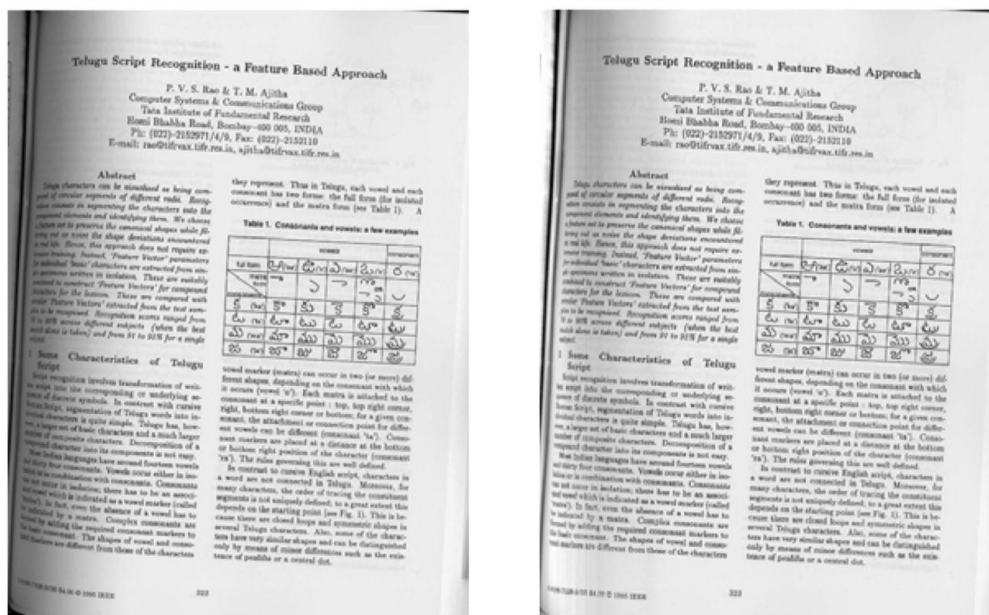
A introdução mostrou na Figura 1.1 que a digitalização de documentos por *scanner* apresenta distorção geométrica. Foi observado que o seu processo de captura é bem peculiar, com projeção ortográfica na direção do movimento do sensor, e projetiva na direção ortogonal a esse movimento (SCHUBERT, 2000; TAN *et al.*, 2006). Enquanto que as imagens de câmeras possuem projeção projetiva em ambas direções.

Tendo em vista tal problema, dois algoritmos são propostos para corrigir a distorção geométrica em imagens digitalizadas por *scanners*. No mais recente, a correção geométrica trata dos dois casos apresentados na Figura 1.1, sendo esta solução inédita.

#### 3.1 Trabalhos relacionados

Ukida e Konish (UKIDA e KONISH, 2001) removem a distorção geométrica em imagens obtidas por *scanner* usando três fontes de luz fixas (vermelho, azul e verde). Ukida e outros colegas (UKIDA *et al.*, 2007) reconstrõem a forma e cor usando um *scanner* com quatro fontes de luz que estão simetricamente posicionadas na linha de captura. Como a maioria dos *scanners* contêm apenas uma fonte de luz, não é possível aplicar os métodos de Ukida de forma universal.

Tan e seus colaboradores apresentam em (TAN *et al.*, 2006) um algoritmo de correção geométrica para documentos obtidos com *scanner* para o cenário da Figura 1.1.b. A geometria tridimensional do documento é reconstituída utilizando o método de forma definida pela sombra (do inglês *shape-from-shading*). Este método requer vários parâmetros do *scanner* (distância focal, ganho e defasagem da transformação fotoelétrica, ângulo do raio de luz com a superfície de vidro do *scanner* etc.) que raramente são informados pelos fabricantes. A Figura 3.1 mostra um exemplo de correção geométrica pelo algoritmo (TAN *et al.*, 2006), que também faz a remoção de sombreamento.

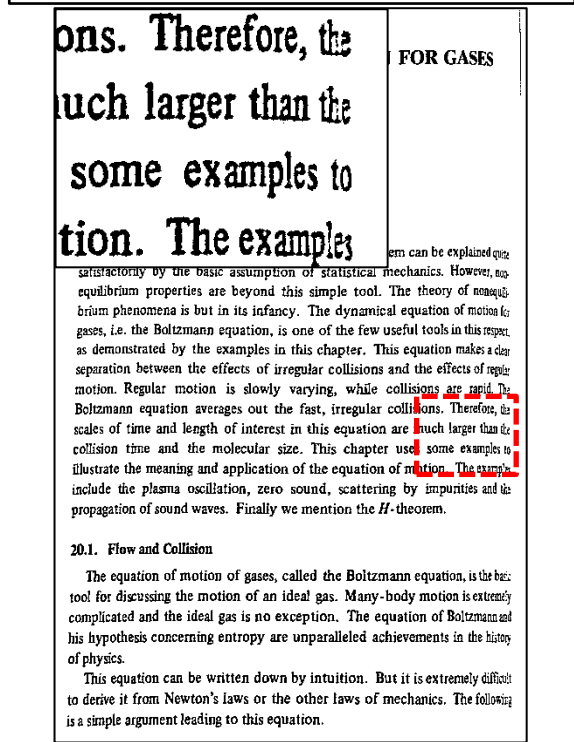
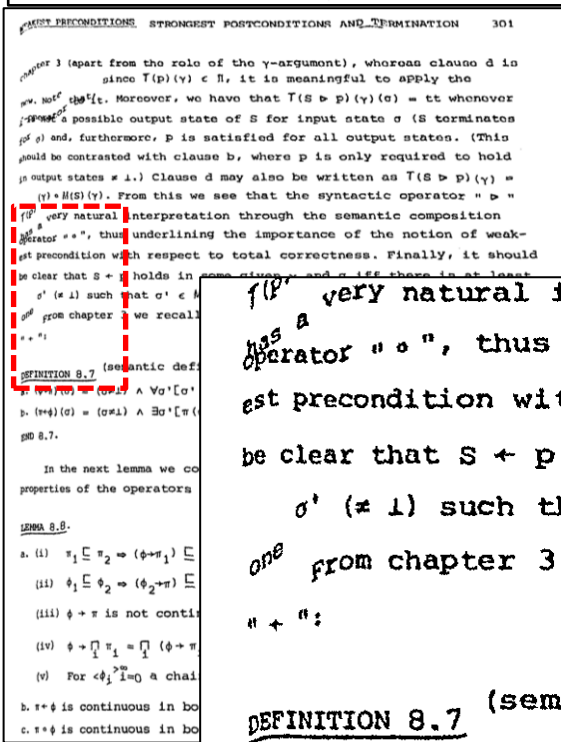
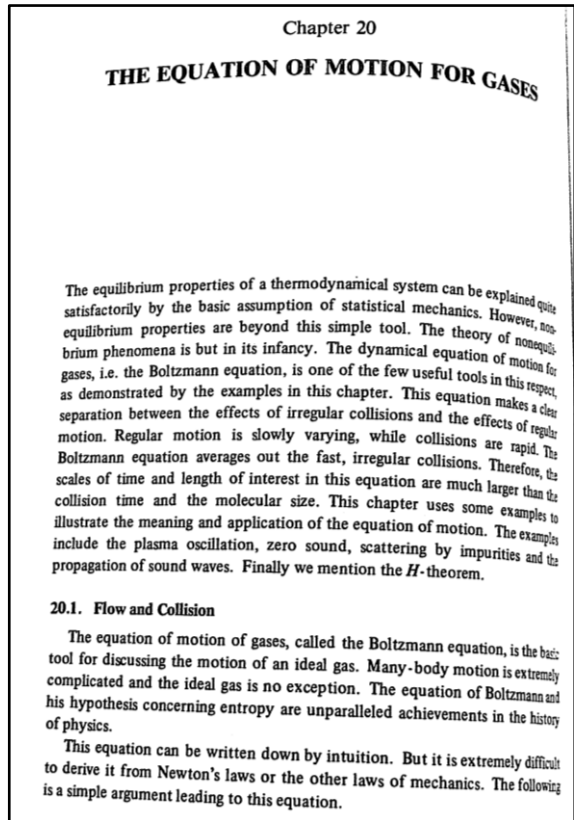
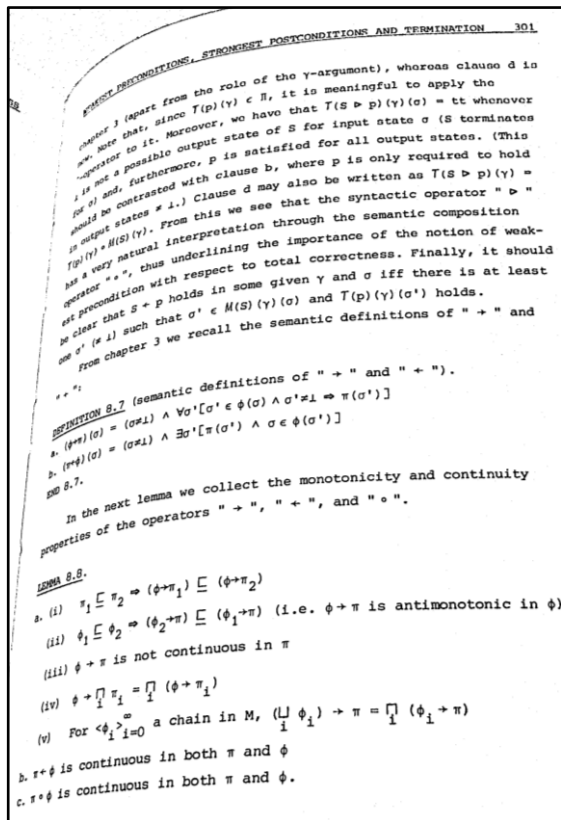


**Figura 3.1 Correção geométrica proposta por (TAN *et al.*, 2006) que requer vários parâmetros de scanners**

Existem vários outros métodos de obtenção de formas pela sombra que utilizam fotos de objeto obtidas por câmera digitais. Existem dois levantamentos sobre esta área (ZHANG *et al.*, 1999; DUROU *et al.*, 2007); a maioria das soluções são iterativas, possuindo um alto custo computacional e mesmo assim não obtêm correção geométrica satisfatória (STAMATOPOULOS *et al.*, 2008).

Outra proposta apresentada na referência (PINTUS *et al.*, 2009) modifica o *scanner* para capturar duas imagens simultaneamente. O modelo geométrico tridimensional do documento é obtido através de técnicas estereográficas. A grande desvantagem desse método é a necessidade de modificação física do *scanner*.

Há também vários métodos que removem a distorção de documentos obtidos por câmeras digitais, a maioria desses algoritmos usa apenas a imagem do documento para fazer a correção. Os métodos geralmente assumem que as linhas de texto são paralelas; uma vez identificada restaura-se o paralelismo “perdido” devido a distorção geométrica. Uma abordagem muito utilizada é transladar as letras; (BUKHARI *et al.*, 2009d; ULGES *et al.*, 2005) são dois exemplos deste tipo de abordagem. No entanto, esta forma de tratamento não corrige a perspectiva, devido ao fato da imagem de entrada apresentar letras inclinadas e mais estreitas que as demais. Em (MASALOVITCH e MESTETSKIY, 2007) remove-se as linhas de texto utilizando curvas de Bézier. A Figura 3.2 mostra os imagens distorcidas e seus resultados de correção geométrica feita por (BUKHARI *et al.*, 2009d; MASALOVITCH e MESTETSKIY, 2007).



a. Correção geométrica através de “mudança de lugar” sobre a imagem acima (BUKHARI *et al.*, 2009d)

b. Correção geométrica utilizando curvas de Bézier sobre a imagem acima proposta por (MASALOVITCH e MESTETSKIY, 2007)

Figura 3.2 Resultado de algoritmos de correção geométrica para câmeras digitais



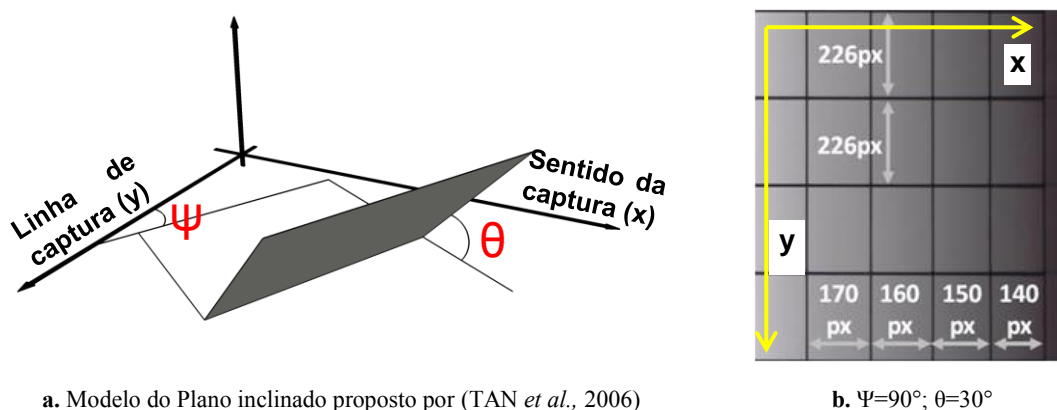
A equipe de Stamatopoulos faz em (STAMATOPOULOS *et al.*, 2008) uma aproximação polinomial de terceiro grau com os caracteres da borda do documento. Fu e seus colegas (FU *et al.*, 2007) fizeram um procedimento similar usando um cilindro em três dimensões.

Esta tese propõe dois algoritmos de correção geométrica para imagens digitalizadas por *scanner*. A próxima seção mostra um estudo (LINS *et al.*, 2010b) sobre o modelo de projeção de *scanner* que comprova o modelo teorizado por (SCHUBERT, 2000; TAN *et al.*, 2006). E nas duas próximas seções são apresentados os dois algoritmos propostos.

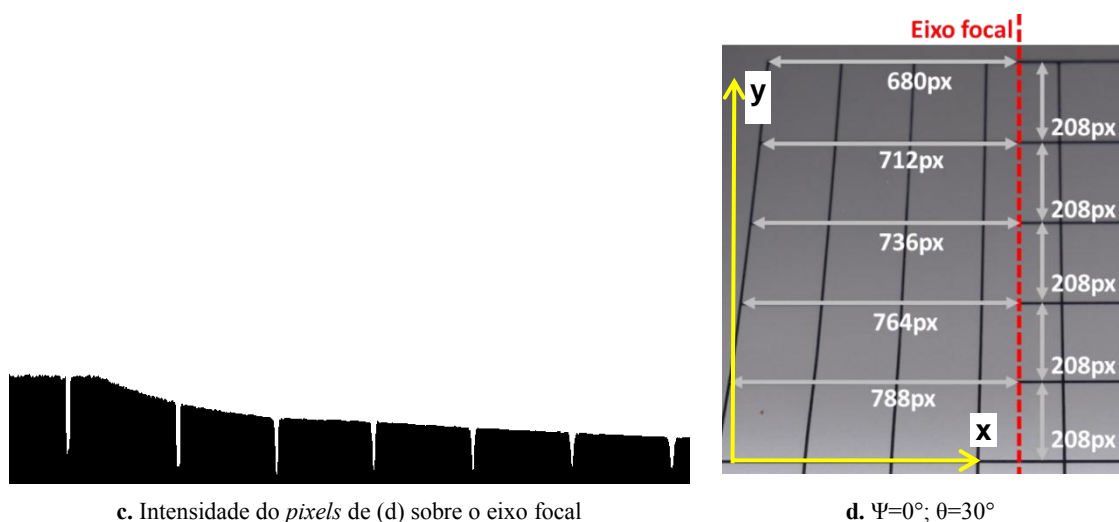
### 3.2 Modelo de captura de imagens por *scanner*

Para verificar o modelo de projeção do *scanner* utilizou-se um papel quadriculado fixado sobre um plano inclinado (TAN *et al.*, 2006) conforme ilustrado na Figura 3.3.a. Esse papel foi digitalizado com o HP ScanJet 5300c na resolução de 300 dpi. A Figura 3.3.b foi obtida com  $\Psi=0^\circ$  e  $\theta=30^\circ$  (situação da Figura 1.1.a); a Figura 3.3.d com  $\Psi=90^\circ$  e  $\theta=30^\circ$  (situação da Figura 1.1.b); onde  $\Psi$  e  $\theta$  são os ângulos observados na Figura 3.3.a.

É possível observar nessas ilustrações que a altura do quadrado é constante com relação ao documento, independente da distância entre o papel e a superfície do *scanner* (o que indica a projeção ortogonal). Só que a largura do quadrado é menor nas regiões em que essa distância é maior (devido à projeção projetiva) e mais próximo do eixo focal. Nota-se também a região mais distante é mais escura em que a distância é maior, como observado na secção transversal do eixo focal da Figura 3.3.c.



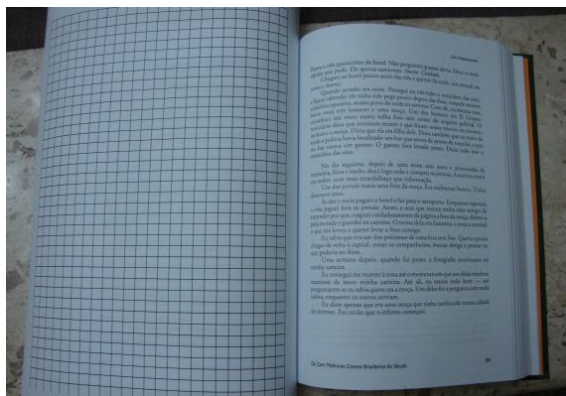
**Figura 3.3** Comprovação sobre o modelo de projeção de *scanner*



**Figura 3.3 (cont.)**

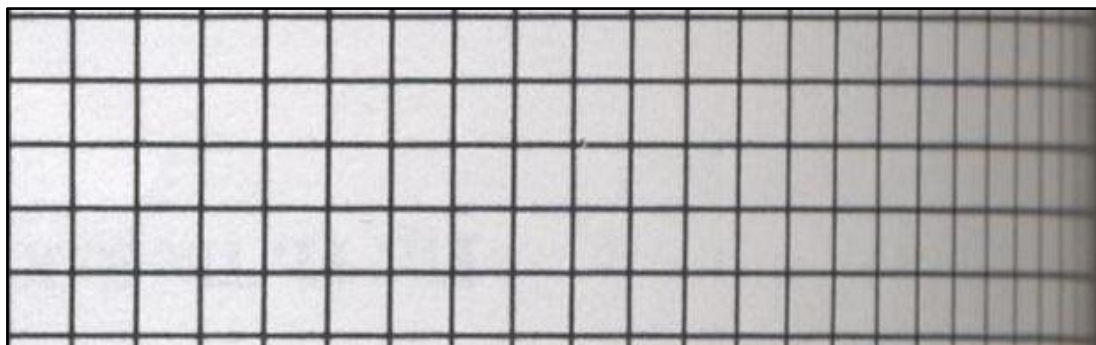
### 3.3 Primeiro Algoritmo Proposto

Para criar a primeira proposta de método de correção geométrica, observou-se que existe uma relação entre a quantidade de luz refletida e a altura do papel, o que pode ser observado na Figura 3.3.c. Para obter esta relação, um papel quadriculado é posto no meio de um livro — onde a distorção é mais acentuada. Digitaliza-se esta página com o *scanner*. A Figura 3.4 mostra uma exemplo de papel quadriculado posto sobre o meio do livro.



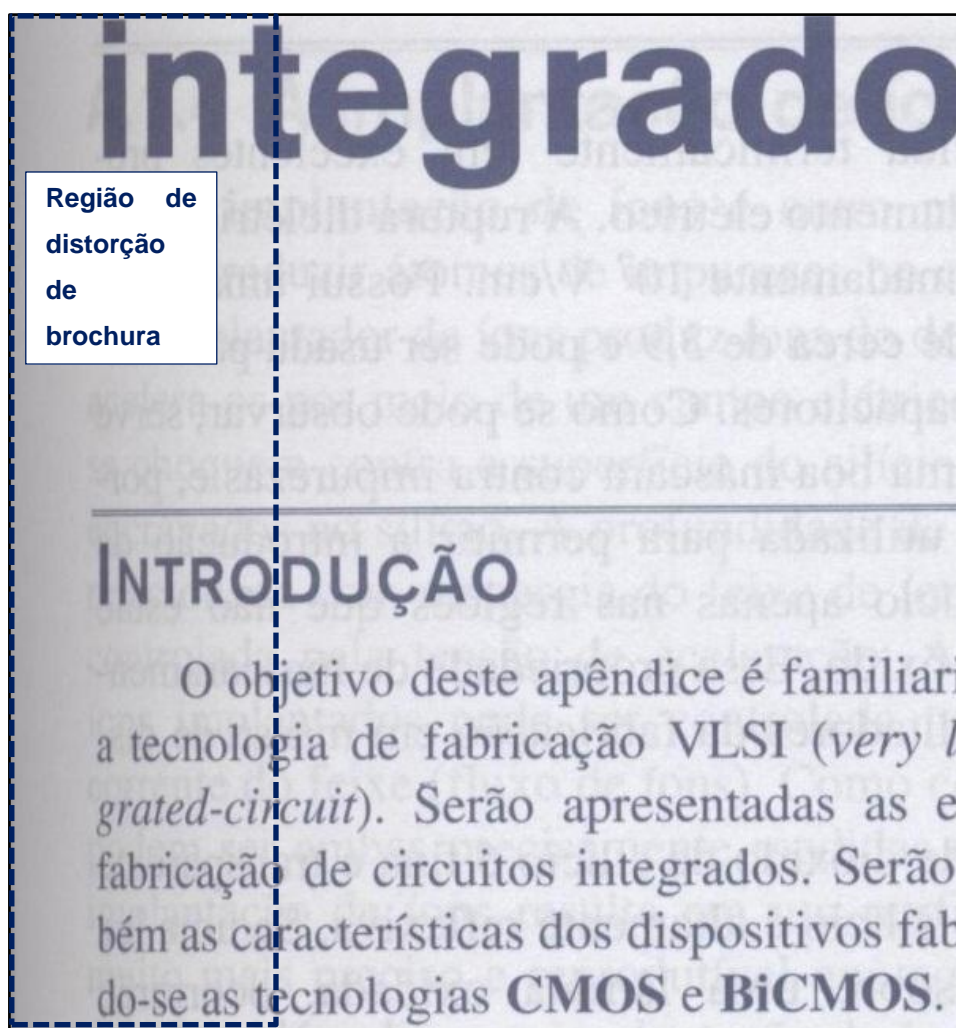
**Figura 3.4 Livro com papel quadriculado posto sobre o meio do livro**

Devido à alta quantidade de quadrados na superfície plana, é possível inferir a medida do lado sem distorção geométrica. Como sabe-se o tamanho com e sem distorção, é possível inferir a altura do quadrado utilizando o teorema de Pitágoras. A Figura 3.5 mostra a imagem usada para o cálculo da relação entre altura e intensidade.



**Figura 3.5** Imagem usada para calcular a relação entre intensidade e altura

Para corrigir a imagem, identifica-se a intensidade do papel em cada coluna da imagem. A altura é inferida através da relação intensidade/altura obtida pela imagem de calibração. Este algoritmo foi publicado em (LINS *et al.*, 2010a). As Figuras 3.6 e 3.8 mostram exemplos de imagens distorcidas e nas Figuras 3.7 e 3.9 as imagens obtidas pelo método proposto, respectivamente.

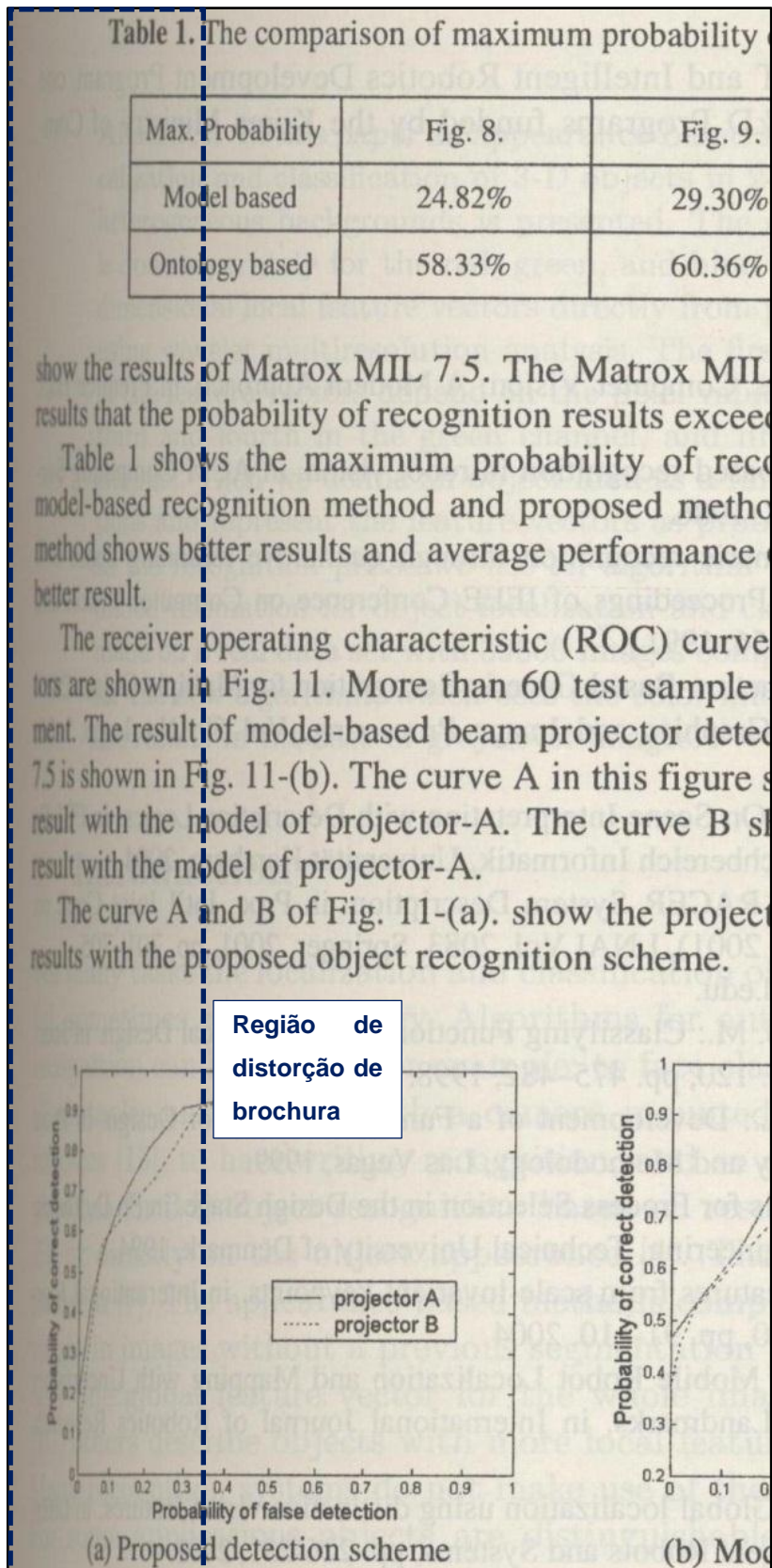


**Figura 3.6** Imagem escaneada com distorção geométrica de dimensões iguais a 1810x3208 pixels.

Nas Figuras 3.6 e 3.8 as letras estão mais estreitas na região de brochura quando comparadas com a parte não pertencente a mesma. A versão corrigida das figuras (3.7 e 3.9, respectivamente) apresenta as letras mais largas (região de distorção de brochura), o que torna as imagens de saída mais largas que as de entrada. Caso haja um ruído na estimação da sombra, haverá um aumento notável na parte fora da região de brochura (região “expandida” erroneamente). Vale ressaltar que o algoritmo não identifica as regiões de brochura; estas foram marcadas apenas para fins ilustrativos.



**Figura 3.7** Aplicação da primeira proposta de correção geométrica na Figura 3.6, que alterou as dimensões para 1939x3208 *pixels*.



**Figura 3.8** Imagem escaneada com distorção geométrica de dimensões iguais a 1472x2744 pixels.

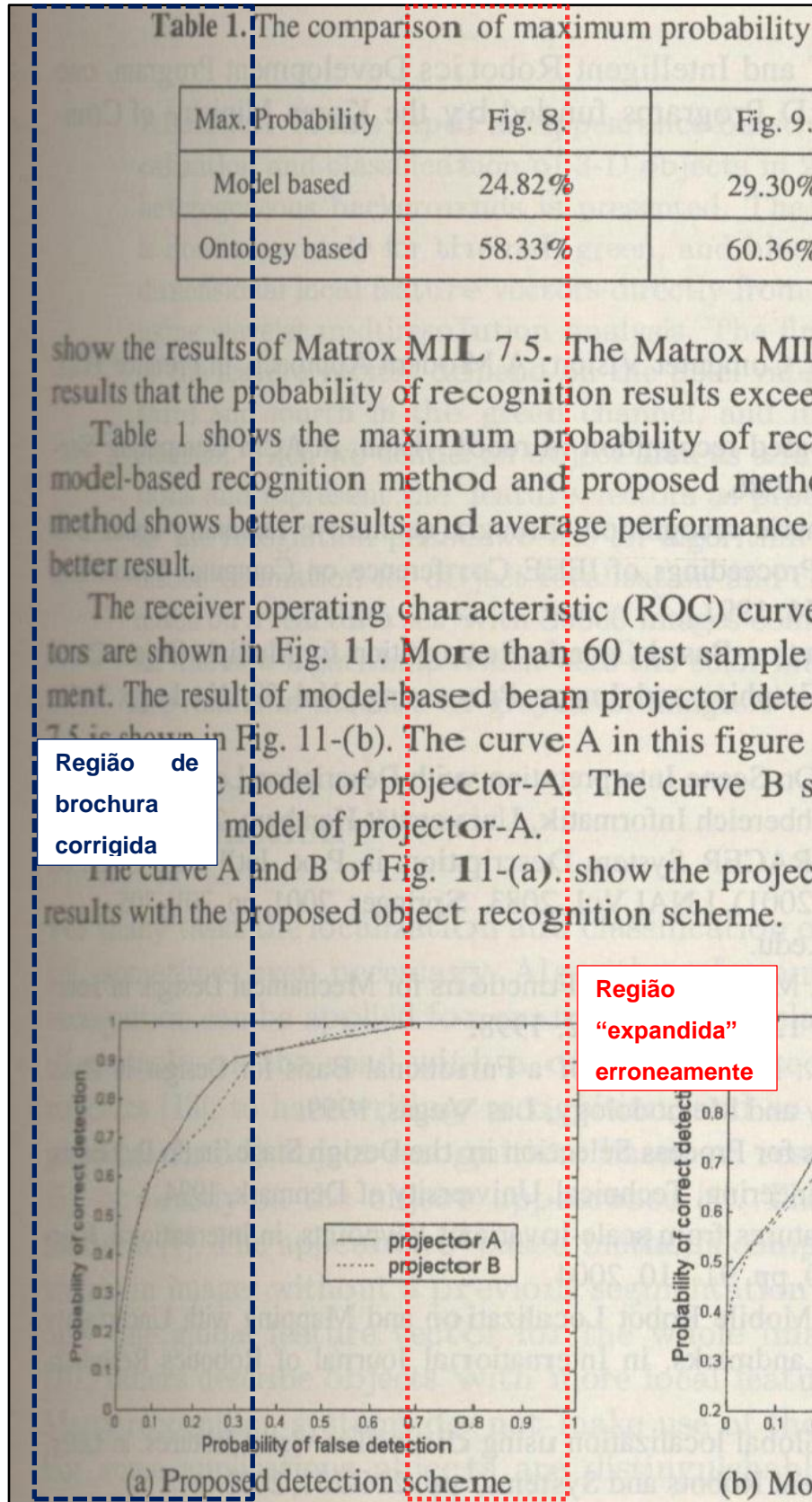


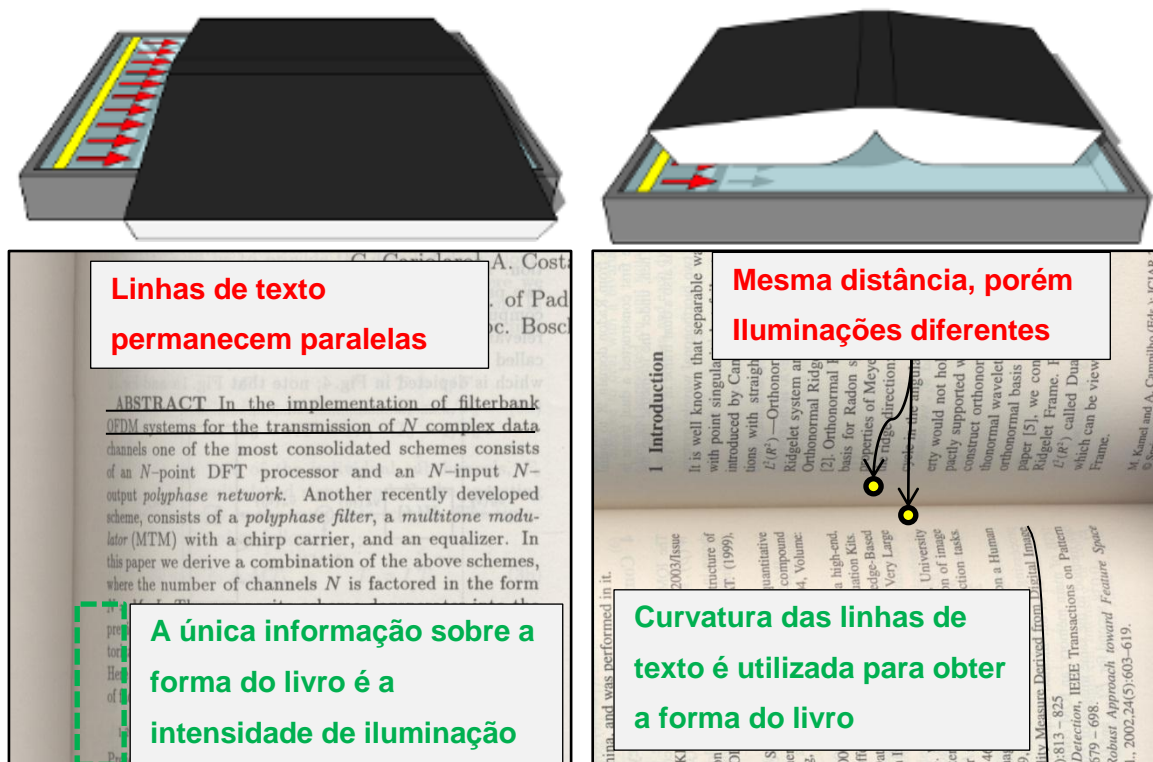
Figura 3.9 Aplicação da primeira proposta de correção geométrica na Figura 3.8, que alterou as dimensões para 1562x2744 pixels.

### 3.4 Segundo Algoritmo Proposto

A proposta anterior resolve apenas o caso da Figura 3.10.a, onde a linha permanece reta devido ao processo de captura do *scanner* apresentado na primeira seção deste capítulo. Não é possível tratar da mesma forma o outro cenário em que o eixo do livro é paralelo à linha de captura (vide Figura 3.10.b). Um contra-exemplo encontra-se na própria Figura 3.10.b, em que a região perto da dobra do livro possui geometrias similares, mas níveis de sobreamento distintos. Portanto, não é possível girar a Figura 3.10.b e estimar a altura pelo seu sobreamento. Para facilitar a leitura, a Figura 1.1. foi replicada com informações adicionais na Figura 3.10.

Outro problema da primeira proposta, é que letras da região plana à superfície de vidro podem apresentar larguras maiores que as demais. Isto é observado na região central das Figuras 3.7 e 3.9.

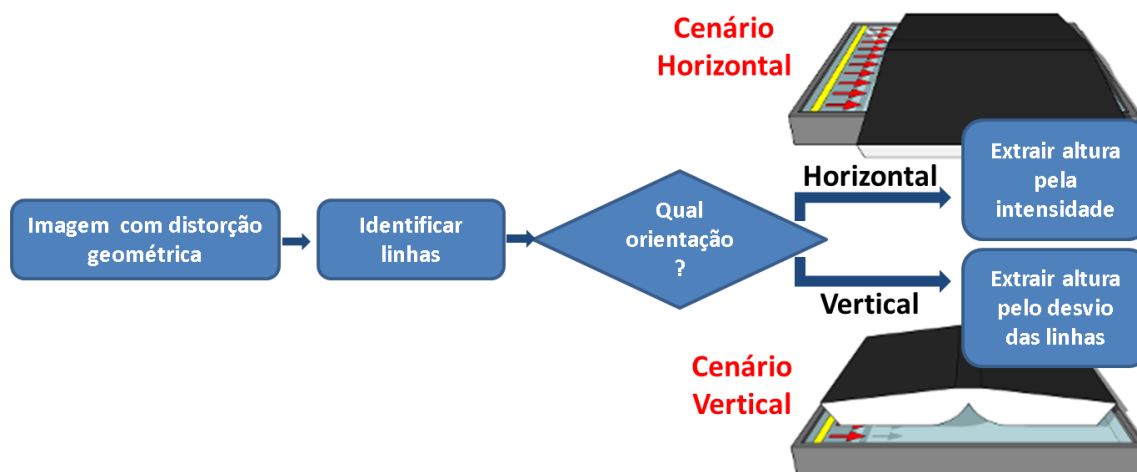
Levando em consideração os problemas supracitados, observou-se que a relação intensidade/altura pode ser aproximada por uma função linear conforme ilustrado na Figura 3.3.c. Com relação à correção geométrica quando o eixo do livro é paralelo à captura, nota-se que as linhas são distorcidas na direção do eixo focal do *scanner*. Neste caso a altura é inferida pelo grau de desvio de tais linhas.



a. Dobra ortogonal ao movimento do sensor      b. Dobra paralela ao movimento do sensor

**Figura 3.10** Cenários de digitalização por *scanners*

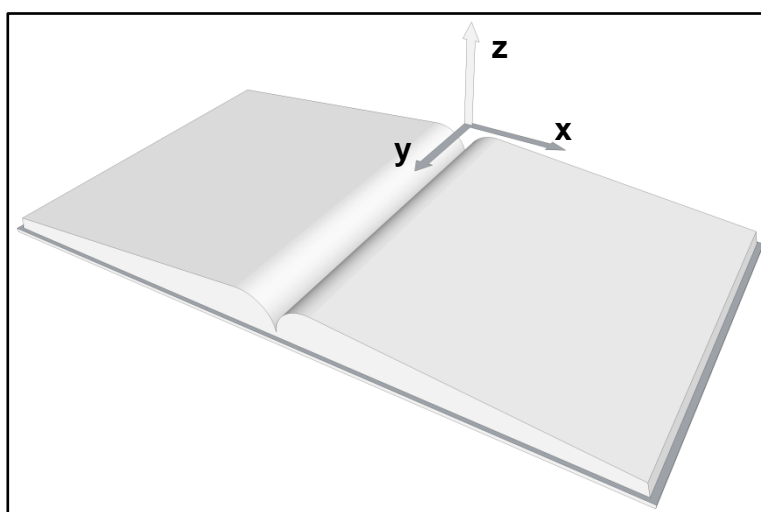
Para identificar tais linhas, aplica-se um dos algoritmos apresentados no capítulo anterior. A segmentação serve também para detectar a orientação do documento. Uma vez feita a segmentação, o algoritmo extrai a altura pelo método da relação linear entre intensidade/altura ou pelos desvio das linhas como ilustrado na Figura 3.11.



**Figura 3.11** Identificação de método de extração da altura do livro

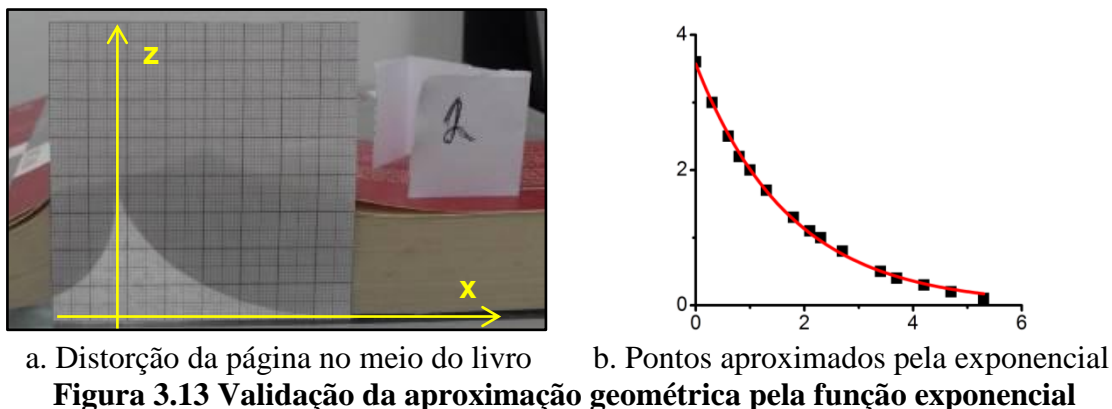
Com os valores das alturas extraídas, é possível aproximar a superfície do livro por uma função exponencial conforme a equação (3.1), onde as constantes  $Z_0$  e  $\lambda$  indicam o início e o quão acentuada é a curva, respectivamente. As coordenadas  $x$ ,  $y$  e  $z$  desta equação estão dispostas conforme Figura 3.12. A Figura 3.13 mostra um exemplo de superfície medida e sua aproximação.

$$z(x) = Z_0 \times e^{\lambda x} . \quad (3.1)$$



**Figura 3.12** Coordenadas  $x$ ,  $y$  e  $z$  relativas à página do livro digitalizado





Uma vantagem de modelar-se a distorção pela equação (3.1), é que não necessário identificar a região de brochura, pois região oposta a esta área é assíntota ao eixo-x. Portanto, cálculo do comprimento entre duas coordenadas-x da equação (3.1) na região oposta à brochura não proporciona alteração perceptível.

As Figuras 3.15 e 3.17 mostram resultados do processamento sobre as Figuras 3.14 e 3.16, respectivamente. Neste caso o eixo do livro é ortogonal ao da linha de captura (cenário horizontal da Figura 3.11). Nas Figuras 3.18 e 3.20, os eixos são paralelos (cenário horizontal da Figura 3.11); os resultados da correção estão disponíveis nas Figuras 3.19 e 3.21. Para facilitar a visualização, as Figuras 3.18-3.21 foram rotacionadas em 90°.

Sobre essas figuras, é possível observar:

- “aumenta-se” a largura das letras apenas na região de brochura;
- o algoritmo aplicado no cenário horizontal produz figuras com larguras maiores (Figuras 3.15 e 3.17) e de mesma altura que as imagens de entrada (Figuras 3.14 e 3.16);
- nas figuras que se encaixam no cenário vertical da Figura 3.11, o paralelismo das linhas de texto é restabelecido;
- as imagens resultantes do cenário vertical (Figuras 3.19 e 3.21) apresentam largura e altura maiores que as imagens de entrada (Figuras 3.18 e 3.20); o aumento da altura se deve à restituição do paralelismo.

Mais detalhes do algoritmo podem ser vistos em (LINS *et al.*, 2010b) no Apêndice desta tese.

Região de distorção de brochura

Fig. 10. ontology (left) vs. model-based

Table 1. The comparison of maximum probability

Max. Probability	Fig. 8.	Fi
Model based	24.82%	29.
Ontology based	58.33%	60.

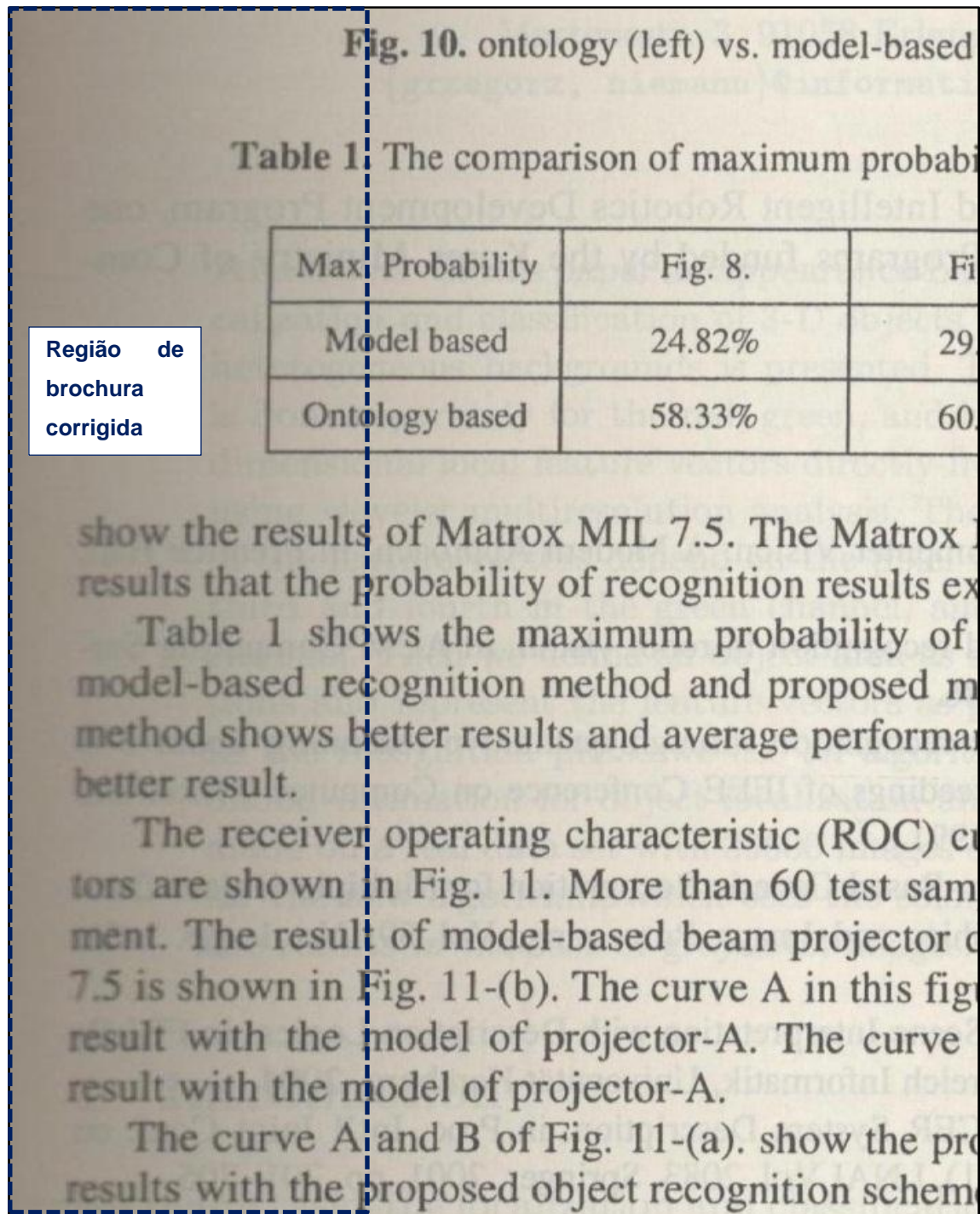
show the results of Matrox MIL 7.5. The Matrox I results that the probability of recognition results ex

Table 1 shows the maximum probability of model-based recognition method and proposed method shows better results and average performance better result.

The receiver operating characteristic (ROC) curves are shown in Fig. 11. More than 60 test samples. The result of model-based beam projector of 7.5 is shown in Fig. 11-(b). The curve A in this figure result with the model of projector-A. The curve result with the model of projector-A.

The curve A and B of Fig. 11-(a). show the proposed results with the proposed object recognition scheme

Figura 3.14 Imagem escaneada no cenário horizontal da Figura 3.11 com dimensões iguais a 1472x2744 pixels.



Região de brochura corrigida

**Figura 3.15** Aplicação da segunda proposta de correção geométrica na Figura 3.14, que alterou as dimensões para 1653x2744 pixels.

Images	IE	AG	STD
SAR image No.1	3.7441	8.9689	65.2375
SAR image No.2	3.7806	8.2740	66.4160
2-level WT fusion result	5.0756	5.3747	61.1428
5-level WT fusion result	5.0706	1.3382	55.1222
LP directional filter banks result	5.0793	11.8071	69.4398

**3.3 Experiments Analysis**

From the above experiments, we obtain two groups of numerical data (see Table 1 and Table 2). According to the index in section 3.1 and combining visual effect, firstly we can find that the information entropy of the images after fusion by several methods is higher than original images. It shows that the information capacity is increased. Furthermore the information entropy obtained by LPDFB method is higher than by 2-level and 5-level WT methods. Also the average grads and standard deviation of fusion images are increased. It shows that fusion images reflect the detail feature of images better and each index obtained by LPDFB method is higher than by 2-level and 5-level WT methods. It is benefit for target estimation and auto-classification. In SAR image experiment section 3.2.2, the fusion results by WT idea not good and the average grads of fusion images are even lower than original images. It indicates that wavelets are not good at capture the detail and texture feature while LPDFB technique can.

From the visual effect, in section 3.2.1, the blurred parts are enhanced by several methods after fusion. 2-level WT method makes the detail part of the image not clear, for example, the plate of the clock is still vague, while 5-level wavelet transform makes the images more clear but the serrate edges in quadrante clock is appeared. The method presented in this paper not only makes the detail image inosculate together but reconstructed fusion result is clear and overcome the serrate edge. In SAR image experiment section 3.2.2, 2-level WT enhances the information of original images, but makes texture of the light area vague, while 5-level WT result is dissatisfied. The reason is that for SAR images with inherent speckle noise, WT strips the low frequency information directly. Consequently the detail information is lost in a certain extent. The loss is related to the decomposition level. The less level is, the more loss of detail information with more spectral information retained, and the more level is, the better spatial detail information representation with worse spectral performance. The method presented in this paper enhances the information of original images, also retains the texture and detail effectively and leads to the clear enhancement images, especially for SAR images. In conclusion, combining the indexes and the visual effect of fusion results, the method in this paper is preponderant.

**4 Conclusion**

Based on the wavelets, a Laplacian Pyramid Directional Filter Banks (LPDFB) is constructed in this paper. Furthermore a method of image enhancement via fusion based on the Laplacian Pyramid Directional Filter Banks (LPDFB) is given. It consid-

**Figura 3.16 Imagem escaneada no cenário horizontal da Figura 3.11 com dimensões iguais a 1544x2728 pixels.**

		Image Enhancement via Fusion Based on LPDFB 245		
<b>Table 2</b> Comparison of SAR image fusion results using different methods				
	Região de brochura corrigida	IE	AG	STD
SAR image		3.7441	8.9689	65.2375
SAR image		3.7806	8.2740	66.4160
2-level WT fusion result		5.0756	5.3747	61.1428
5-level WT fusion result		5.0706	1.3382	55.1222
LP directional filter banks result		5.0793	11.8071	69.4398

**3.3 Experiments Analysis**

From the above experiments, we obtain two groups of numerical data (see Table 1 and Table 2). According to the index in section 3.1 and combining visual effect, firstly we can find that the information entropy of the images after fusion by several methods is higher than original images. It shows that the information capacity is increased. Further more the information entropy obtained by LPDFB method is higher than by 2-level and 5-level WT methods. Also the average grads and standard deviation of fusion images are increased. It shows that fusion images reflect the detail feature of images better and each index obtained by LPDFB method is higher than by 2-level and 5-level WT methods. It is benefit for target estimation and auto-classification. In SAR image experiment section 3.2.2, the fusion results by WT idea not good and the average grads of fusion images are even lower than original images. It indicates that wavelets are not good at capture the detail and texture feature while LPDFB technique can.

From the visual effect, in section 3.2.1, the blurred parts are enhanced by several methods after fusion. 2-level WT method makes the detail part of the image not clear, for example, the plate of the clock is still vague, while 5-level wavelet transform makes the images more clear but the serrate edges in quadrate clock is appeared. The method presented in this paper not only makes the detail image inosculate together but reconstructed fusion result is clear and overcome the serrate edge. In SAR image experiment section 3.2.2, 2-level WT enhances the information of original images, but makes texture of the light area vague, while 5-level WT result is dissatisfied. The reason is that for SAR images with inherent speckle noise, WT strips the low frequency information directly. Consequently the detail information is lost in a certain extent. The loss is related to the decomposition level. The less level is, the more loss of detail information with more spectral information retained, and the more level is, the better spatial detail information representation with worse spectral performance. The method presented in this paper enhances the information of original images, also retains the texture and detail effectively and leads to the clear enhancement images, especially for SAR images. In conclusion, combining the indexes and the visual effect of fusion results, the method in this paper is preponderant.

**4 Conclusion**

Based on the wavelets, a Laplacian Pyramid Directional Filter Banks (LPDFB) is constructed in this paper. Furthermore a method of image enhancement via fusion based on the Laplacian Pyramid Directional Filter Banks (LPDFB) is given. It consid-

**Figura 3.17** Aplicação da segunda proposta de correção geométrica na Figura 3.16, que alterou as dimensões para 1544x2728 pixels.

$J = 4$ . In all cases, the original (wrapped and unwrapped) signal was contaminated with noise less than 5 dB, to consider only low SNR environments. The first column shows the original (wrapped and unwrapped) signal, the second column the contaminated images (wrapped and unwrapped with an algorithm that does not tolerate phase residues) and the third one the results obtained by using the filter COMP\_UN. The simulated complex image was contaminated with noise model 2.

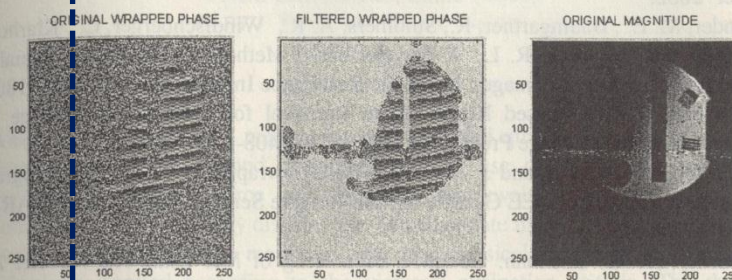


Fig. 2. De-noising of phantom image. Wavelet Bior2.6,  $J=4$ , filter AB\_HARD

Figure 2 shows a phantom image in its original version, where a total of 28097 phase residues were detected. After filtering with AB\_HARD, the number of remaining phase residues was 4990 for an 82% decrement. In this case, once the de-noising process has finished, further processing was applied in order to make more uniform the intensity of the background pixels, making the filtered image values in purely noisy regions to have low intensity.

#### 4 Discussion and Conclusions

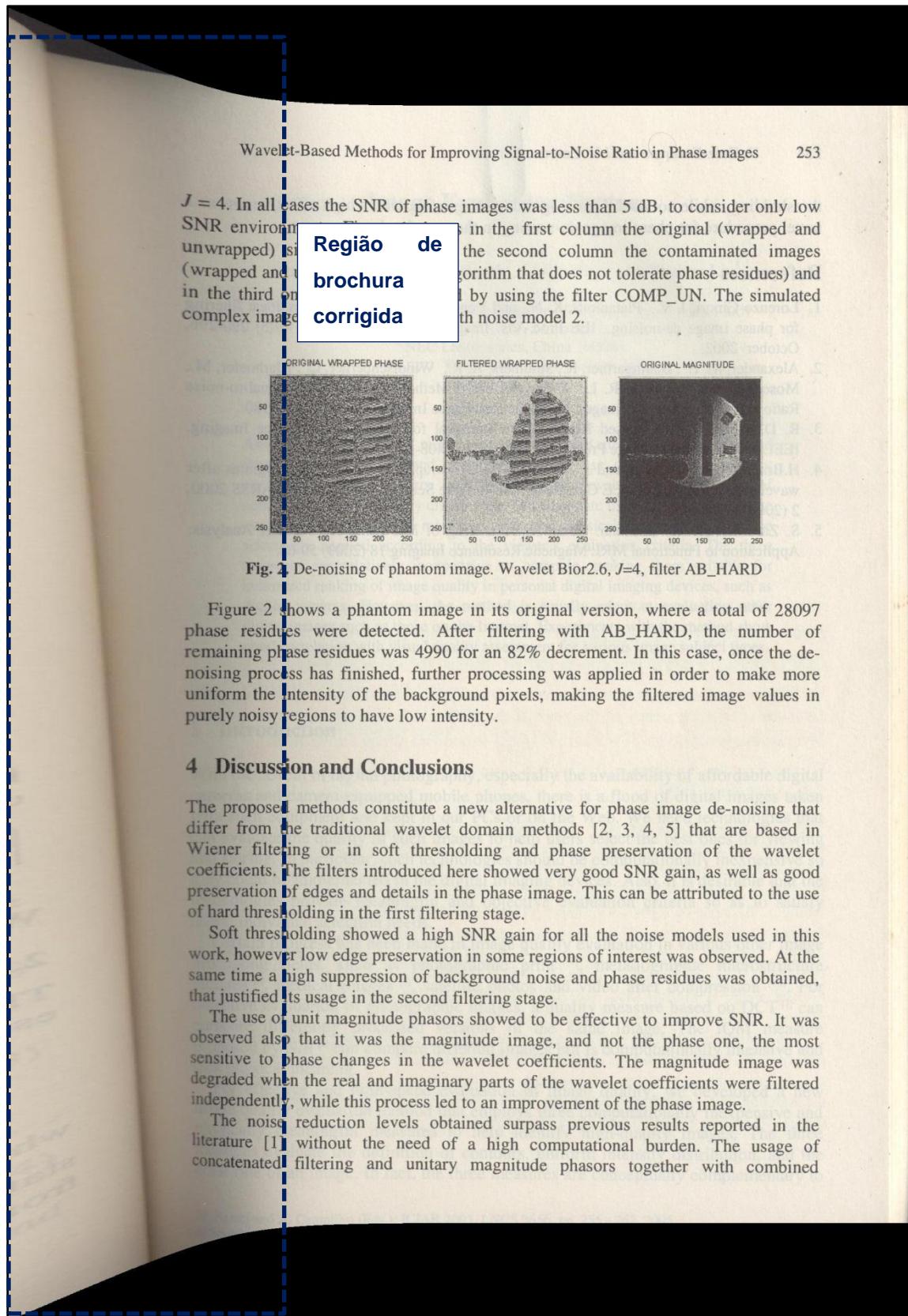
The proposed methods constitute a new alternative for phase image de-noising that differ from the traditional wavelet domain methods [2, 3, 4, 5] that are based in Wiener filtering or in soft thresholding and phase preservation of the wavelet coefficients. The filters introduced here showed very good SNR gain, as well as good preservation of edges and details in the phase image. This can be attributed to the use of hard thresholding in the first filtering stage.

Soft thresholding showed a high SNR gain for all the noise models used in this work, however low edge preservation in some regions of interest was observed. At the same time a high suppression of background noise and phase residues was obtained, that justified its usage in the second filtering stage.

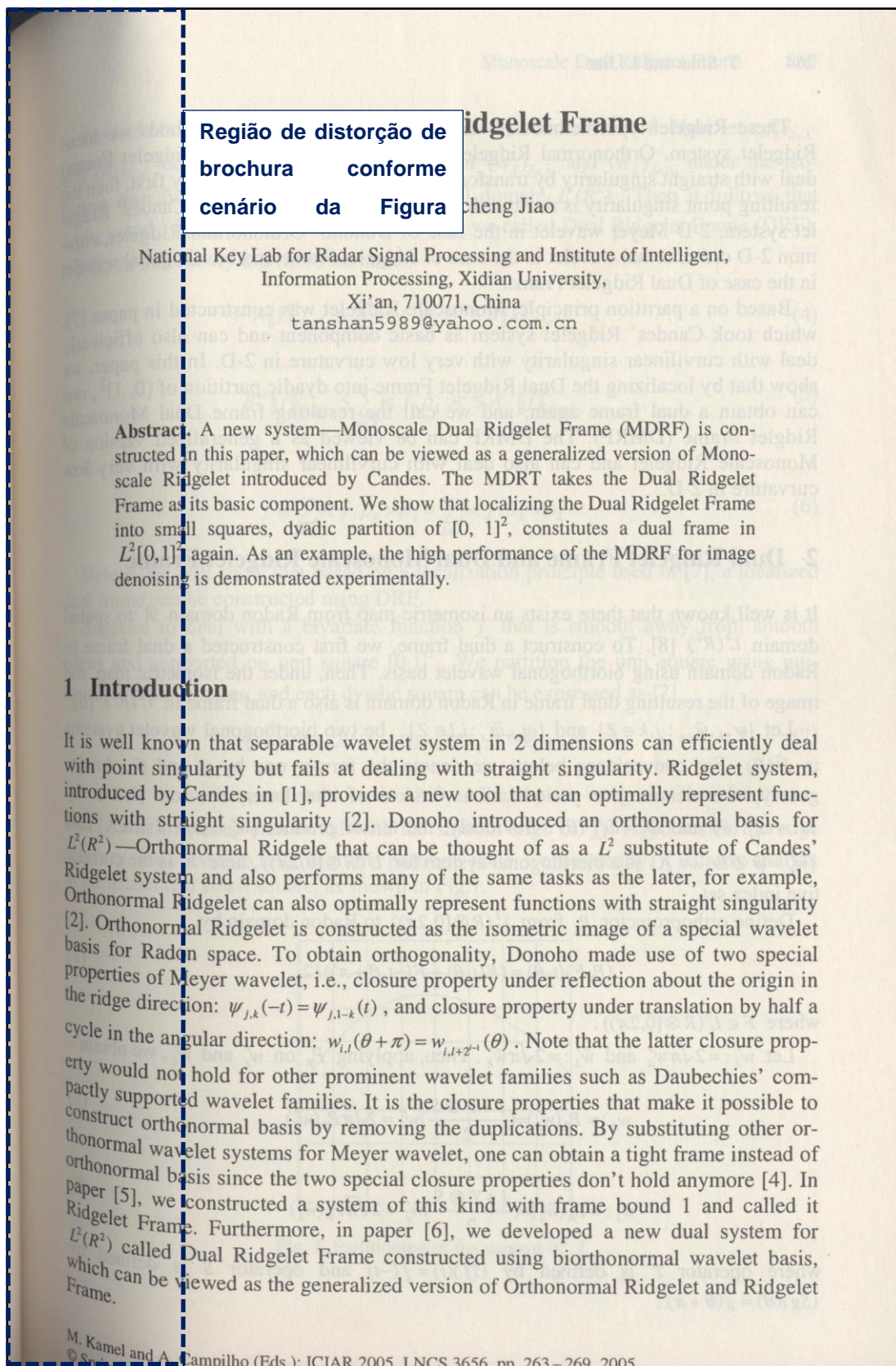
The use of unit magnitude phasors showed to be effective to improve SNR. It was observed also that it was the magnitude image, and not the phase one, the most sensitive to phase changes in the wavelet coefficients. The magnitude image was degraded when the real and imaginary parts of the wavelet coefficients were filtered independently, while this process led to an improvement of the phase image.

The noise reduction levels obtained surpass previous results reported in the literature [1] without the need of a high computational burden. The usage of concatenated filtering and unitary magnitude phasors together with combined

Figura 3.18 Imagem escaneada no cenário vertical da Figura 3.11 com dimensões iguais a 2551x1768 pixels.

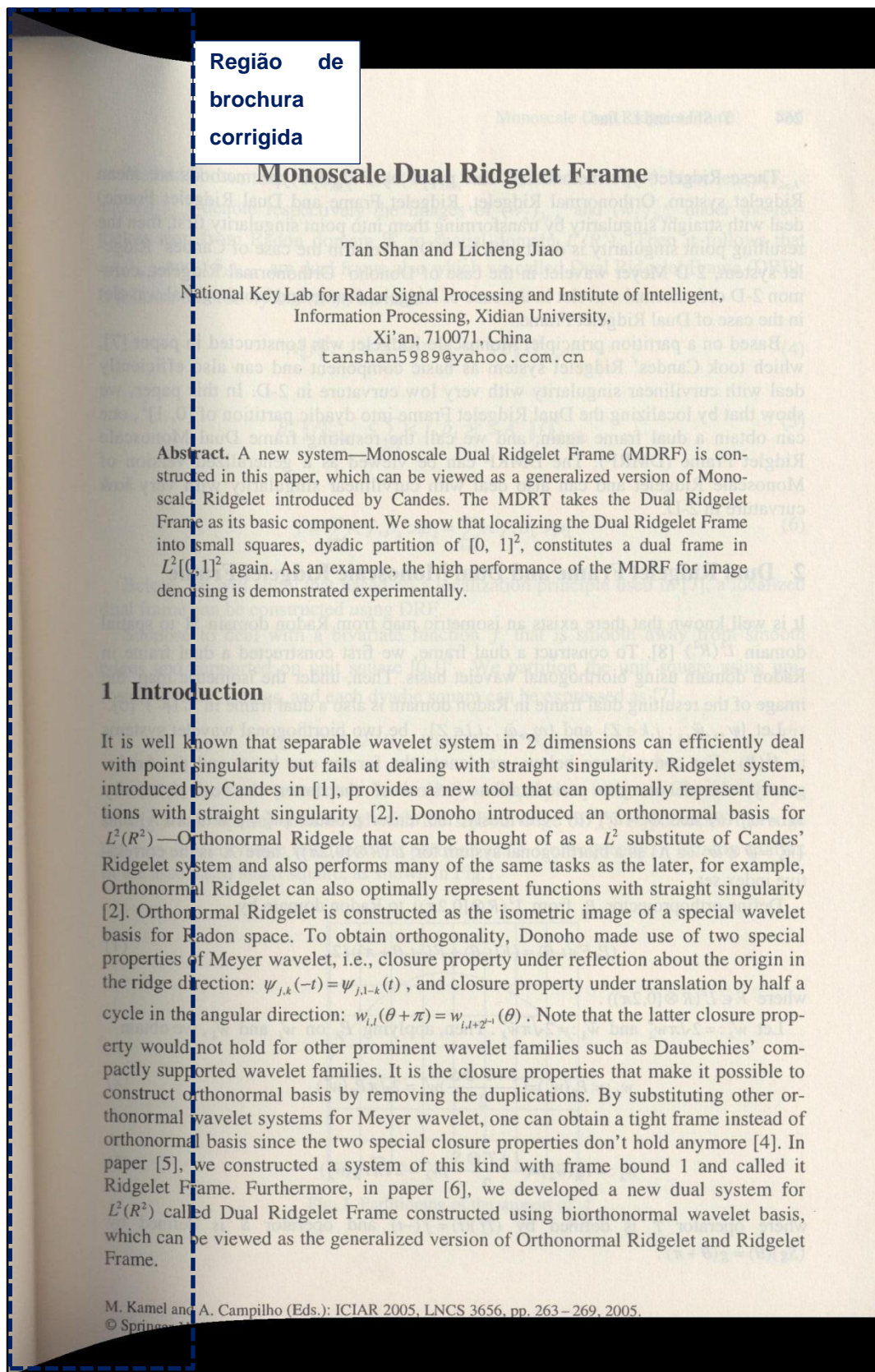


**Figura 3.19** Aplicação da segunda proposta de correção geométrica na Figura 3.18, que alterou as dimensões para 2858x1927 pixels.



**Figura 3.20** Imagem escaneada no cenário vertical da Figura 3.11 com dimensões iguais a 2551x1680 pixels.





**Figura 3.21** Aplicação da segunda proposta de correção geométrica na Figura 3.20 que alterou as dimensões para 2771x1773 pixels.

## 4 Remoção de Embaçamento (*blur*)

A introdução desta tese mostrou exemplos de embaçamento (*blur*) em documentos digitalizados por *scanners* e câmeras digitais portáteis. Vários fatores causam o embaçamento, como (LINS, 2009):

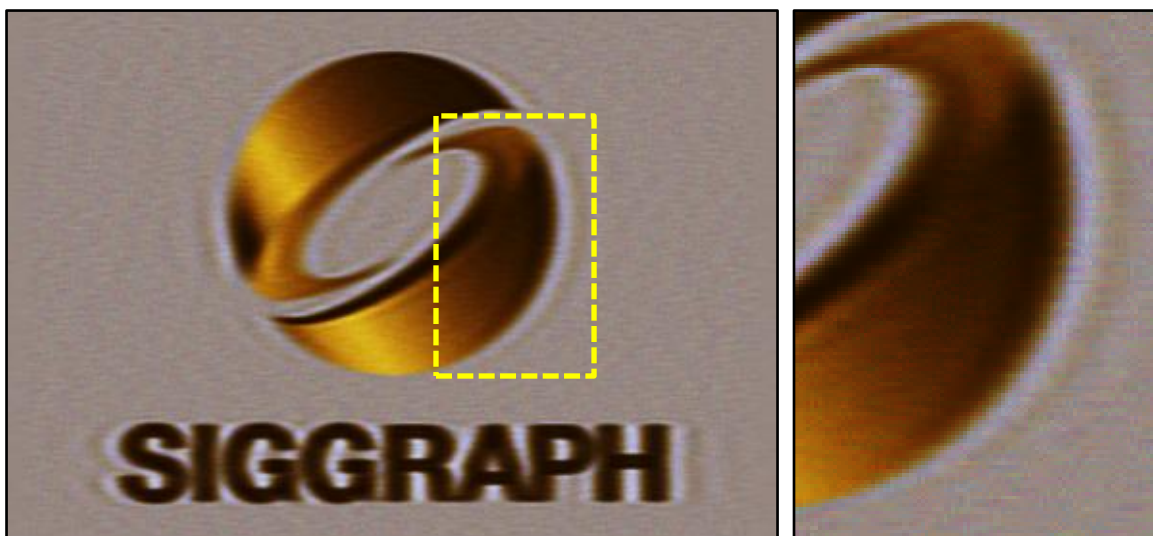
- tremer o dispositivo no momento da captura;
- o objeto da cena estar em movimento;
- a lente de captura não estar em foco;
- manchas de tinta.

Na literatura técnica existem várias propostas de algoritmos para corrigir o embaçamento de fotografias, a maioria possui um alto custo computacional e são difíceis de implementar. Neste capítulo propõe-se um algoritmo simples para remoção de embaçamento ocasionado pela falta de foco em lente de *scanners*.

### 4.1 Trabalhos relacionados

A imagem a ser processada pode ser modelada como o consequência da convolução de um filtro sobre a imagem sem ruído. Neste caso, a imagem resultante também pode ser obtida a partir da multiplicação dos termos no domínio da frequência. A saída desejada do algoritmo é a imagem sem ruído, onde não é informado o filtro de embaçamento.

A função de embaçamento é conhecida pela sigla PSF (do inglês *Point of Spread Function*), o que seria equivalente a função de resposta de um filtro a um impulso, só que em duas dimensões. A deconvolução é feita utilizando algoritmos iterativos (GONZALEZ e WOODS, 2008; RICHARDSON, 1972; LUCY, 1974). Para identificar a PSF, vários algoritmos são propostos, dentre eles pode-se citar (DEMOMENT, 1989; NEELAMANI *et al.*, 1989; ROTH *et al.*, 2005; RUDIN *et al.*, 1992; JOSHI, 2008). Um defeito muito comum desses algoritmos é o aparecimento de “oscilações” (*ringing*) na imagem corrigida (JOSHI, 2008), como pode ser visto na Figura 4.1.



**Figura 4.1** Oscilações após a aplicação da remoção de embaçamento

No caso de documentos digitalizados, o nível de embaçamento é variável, o que é claramente visível nas Figuras 1.1 e 1.2 presentes na introdução desta tese. Portanto, as alternativas anteriores não são adequadas para resolver tal problema.

## 4.2 Algoritmo proposto

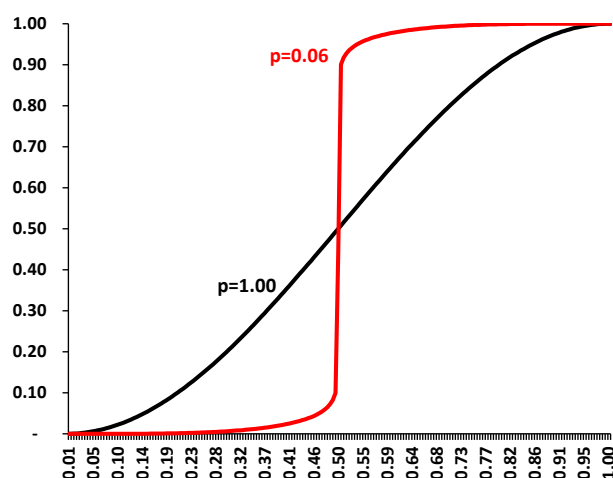
Tendo em vista os fatos apresentados no início do capítulo, propõe-se um filtro simples para reduzir embaçamento de imagens em escalas de cinza com o embaçamento de falta de foco. Este algoritmo foi publicado no ITS 2010 e GREC 2011 conforme descrito nas referências (OLIVEIRA *et al.*, 2010b) e (OLIVEIRA *et al.*, 2011c), respectivamente. O filtro consiste em pesquisar ao redor de cada pixel os valores de intensidade mais alta e baixa. O novo valor do pixel é calculado pela equação (4.1), onde:

- $S(t)$  é a função-S definida pela equação (4.2),  $p$  é o parâmetro livre atribuído no intervalo  $(0, 1]$  que indica a “agressividade” da correção, quanto menor for seu valor mais evidente será a mudança (a Figura 4.2 mostra a função com valores de  $p$  iguais a 1 e 0,06);  $t$  está definido entre 0 e 1 sendo multiplicado por  $\pi$  para:
  - $S(0) = 0$  (mínimo) e  $S(1) = 1$  (máximo);
  - quando o valor de  $t$  estiver entre 0 e 0,5, a função tem valor menor que a função identidade (i.e. a função torna o valor mais próximo de 0);
  - caso contrário (i.e.  $t$  entre 0,5 e 1), a função tem valor maior que a função identidade (i.e. a função torna o valor mais próximo de 1).

- $min$  e  $max$  é a menor e maior intensidade encontrada ao redor do ponto a ser corrigido, respectivamente;
- $I_b$  é o valor da intensidade do *pixel* atual, este valor é escalonado para o intervalo  $[0,1]$  pela divisão por  $(max - min)$ , o resultado é utilizado como entrada da função  $S(t)$ ; após o cálculo da função  $S(t)$  restabelece-se a escala para o intervalo  $[min, max]$ .

$$I_u = \left[ S \left( \frac{I_b - min}{(max - min)} \right) \times (max - min) \right] + min \quad (4.1)$$

$$S(t) = 0,5 - 0,5 \times \text{sinal}(\cos t\pi) \times |\cos t\pi|^p \quad (4.2)$$

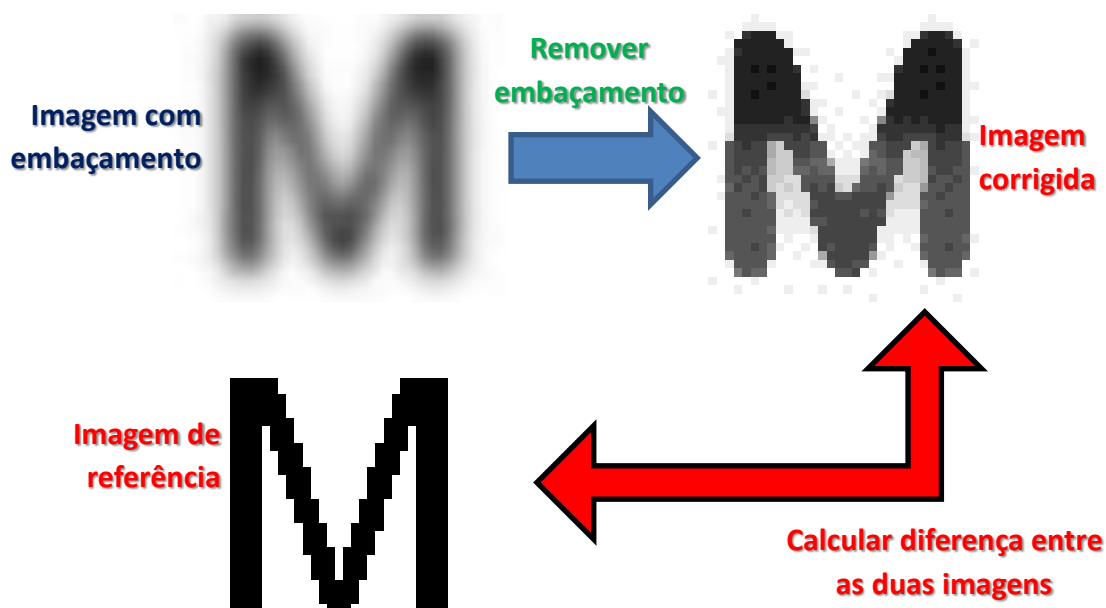


**Figura 4.2** Função  $S(t)$  para  $p$  igual a 1,00 e 0,06

Vale ressaltar que caso  $S(t) = t$ , a equação (4.1) transforma-se em  $I_u = I_b$ , ou seja, não há a modificação na imagem de entrada.

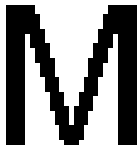
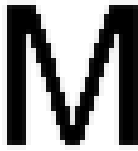
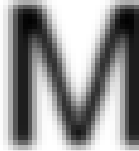

#### 4.2.1 Resultados

O desempenho deste método é validado através da geração de imagens embaçadas por computadores a partir de uma imagem sem o defeito. Letras maiúsculas e minúsculas do alfabeto foram criadas sem embaçamento com o ImageJ (IMAGEJ, 2011), essas imagens serão definidas como de referência. A partir dessas imagens o defeito foi reproduzido, também utilizando o ImageJ. A imagem com embaçamento é utilizada como entrada do algoritmo de remoção, as imagens de saída são comparadas com as de referências, obtendo-se a medida de qualidade. A Figura 4.3 ilustra tal procedimento para obtenção da medida de qualidade.



**Figura 4.3** Obtenção das medidas de qualidade para a remoção de embaçamento

A Figura 4.4. mostra a letra M com vários níveis de embaçamento; este fator de embaçamento foi definido por (IMAGEJ, 2011). Esta figura também mostra a medida SSIM (WANG *et al.*, 2004) em relação as imagens embaçadas e referência da letra M. Este valor indica o grau de similaridade entre duas imagens compreendido entre 0 e 1, onde quanto mais perceptível for a disparidade entre as imagens mais próximo de 0 será o seu valor, por outro lado o valor 1 indica que as imagens são idênticas.

			
Referência	1 (0,999)	3 (0,954)	5 (0,910)
	“Nível de embaçamento” (IMAGEJ, 2011) e similaridades em parênteses entre as imagens embaçadas e de referência		

**Figura 4.4** Letra “M” gerada por computador com vários níveis de embaçamento

O desempenho do algoritmo proposto pode ser observado na Tabela 4.1 e mostra as similaridades entre as imagens de referência e corrigida (com valor de p igual a 0,06), onde a segunda é a combinação da correção com tamanho de janela do filtro (última linha da tabela) sobre a imagem com nível de embaçamento (primeira coluna da tabela) indicado na célula correspondente. É possível também observar a letra “M” corrigida em cada combinação referente à célula. O percentual em parênteses representa o ganho de

informação das imagens corrigidas em relação às embaçadas definido pela equação (4.3). Esta equação utiliza:

- a similaridade SSIM entre as imagens de referência e embaçadas;
- a similaridade SSIM entre as imagens de referência e corrigidas.

Para o ganho de informação ser positivo, a imagem corrigida precisa ser mais similar à imagem de referência que a imagem embaçada (i.e. a similaridade SSIM mais próxima de 1). Caso isto aconteça, o termo da esquerda na equação (4.3) será maior que 1. Este é o caso de todas as combinações presentes na Tabela 4.1.

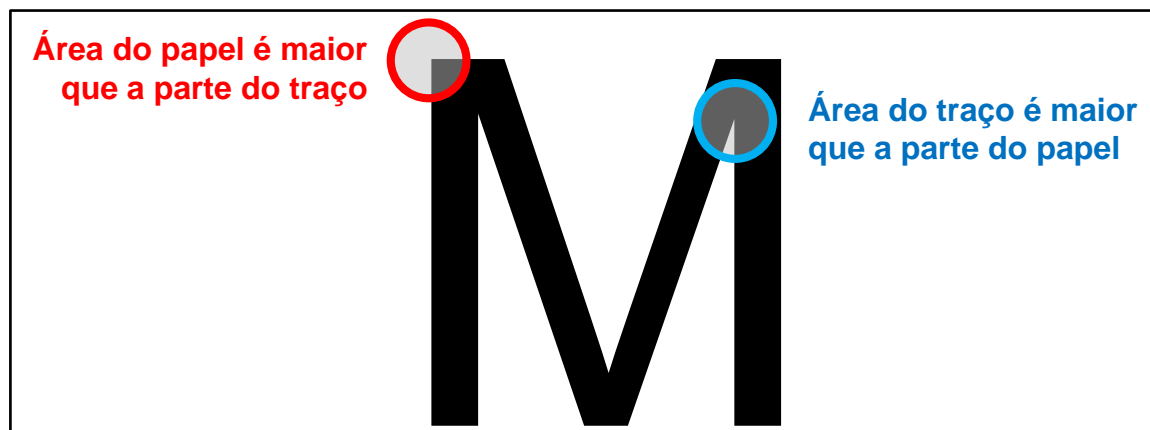
Na Tabela 4.1, foram ressaltados de vermelho e negrito os valores de similaridade que tiveram maior ganho de informação na mesma linha, pois a mesma imagem de entrada foi utilizada na aplicação do algoritmo proposto. Nota-se que os maiores ganhos não são relativos ao uso de maiores janelas, portanto utilizar um tamanho maior que necessário pode refletir numa queda de desempenho como ilustrado nesta tabela (i.e. o ganho é menor nas células em que a janela é maior).

$$\text{Ganho de informação} = \frac{\text{Similaridade entre imagens **corrigidas e original**}}{\text{Similaridade entre imagens **embaçadas e original**}} - 1 \quad (4.3)$$

Tabela 4.1 Média SSIM entre as imagens sem embaçamento e corrigidas com  $p = 0,06$ 

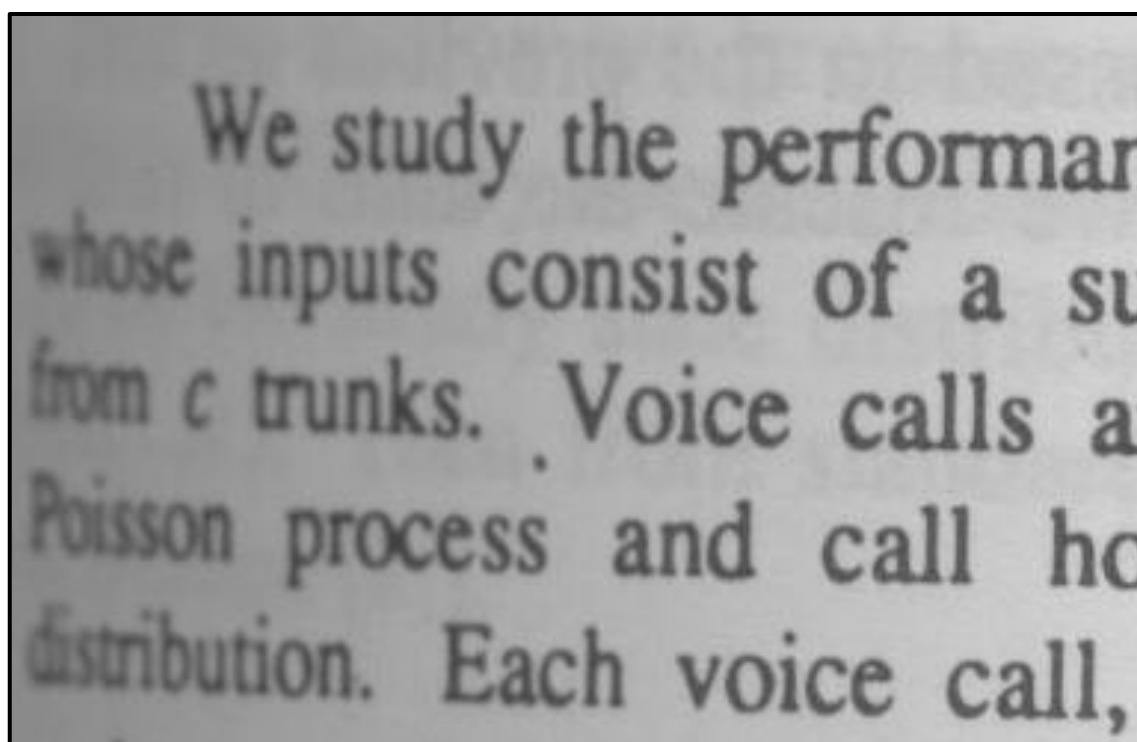
<b>Nível 5</b>	0,942 (+3,56%)	<b>0,947</b> <b>(+4,06%)</b>	0,940 (+3,30%)	0,941 (+3,38%)	0,942 (+3,53%)
<b>Nível 4</b>	0,968 (+3,85%)	<b>0,973</b> <b>(+4,37%)</b>	0,970 (+4,09%)	0,971 (+4,17%)	0,972 (+4,27%)
<b>Nível 3</b>	0,987 (+3,44%)	<b>0,991</b> <b>(+3,85%)</b>	0,991 (+3,82%)	0,990 (+3,78%)	0,991 (+3,80%)
<b>Nível 2</b>	0,999 (+2,30%)	<b>0,998</b> <b>(+2,25%)</b>	0,998 (+2,23%)	0,998 (+2,23%)	0,998 (+2,22%)
<b>Nível 1</b>	<b>1,000</b> <b>(+0,07%)</b>	1,000 (+0,07%)	1,000 (+0,07%)	1,000 (+0,07%)	1,000 (+0,07%)
Nível Janela	1	2	3	4	5

Observa-se que a degradação de desempenho ocorre em regiões periféricas das letras, onde no traço, a região predominante ao redor é do papel (i.e. círculo vermelho da Figura 4.5). O contrário também é possível, conforme ilustrado no círculo azul da Figura 4.5. Em ambos casos há a perda de informação, como pôde ser observado na Tabela 4.1 nas imagens da letra “M”.



**Figura 4.5** Regiões periféricas da letra “M”

A Figura 4.6 mostra os resultados do algoritmo com janela 5x5 e vários valores de  $p$ , quanto menor o valor  $p$ , mais agressiva é a correção; observa-se que em (d) as bordas dos caracteres da parte esquerda estão bem realçadas. A Figura 4.7 fixa  $p$  em 0,25, variando-se os tamanhos de janela. Neste caso nota-se que não é possível retirar o ruído onde o nível de embaçamento é maior que o traço da letra.



a. Imagem original

**Figura 4.6** Resultado do filtro de remoção de embaçamento com janela 3x3



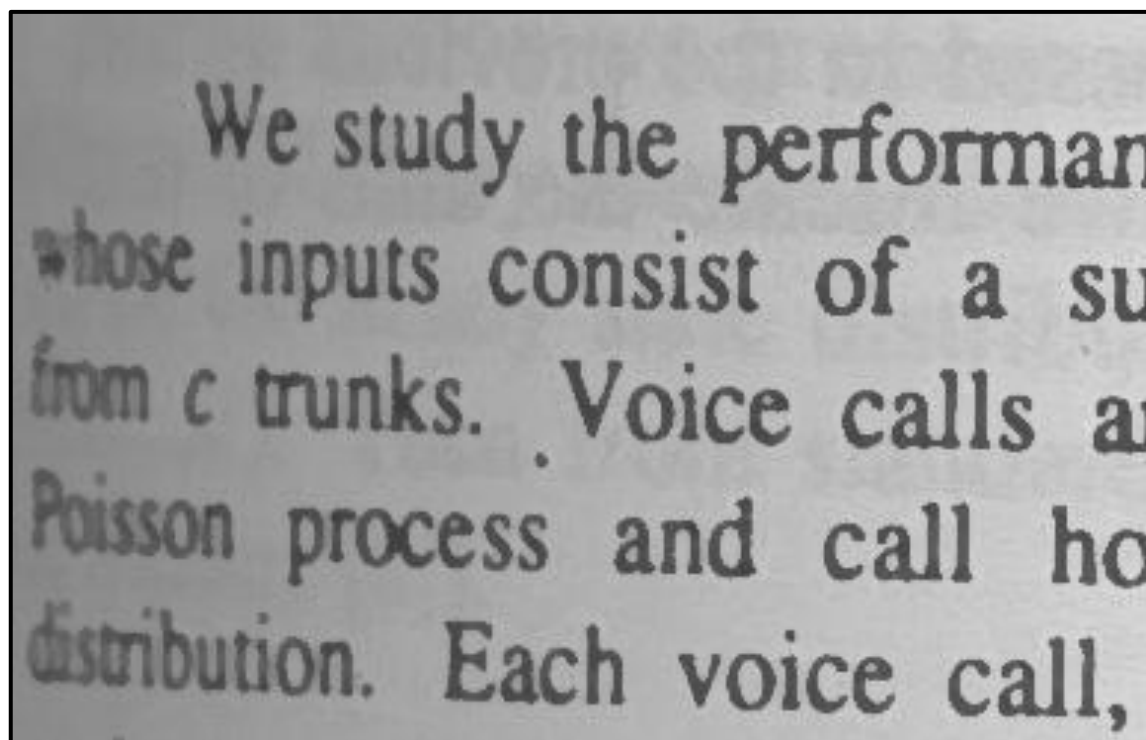
We study the performance of a system whose inputs consist of a sum of voice calls from  $c$  trunks. Voice calls arrive according to a Poisson process and call holding times have an exponential distribution. Each voice call, when it arrives, is served by one of the  $c$  trunks.

b. Correção com  $p = 1,00$

We study the performance of a system whose inputs consist of a sum of voice calls from  $c$  trunks. Voice calls arrive according to a Poisson process and call holding times have an exponential distribution. Each voice call, when it arrives, is served by one of the  $c$  trunks.

c. Correção com  $p = 0,50$

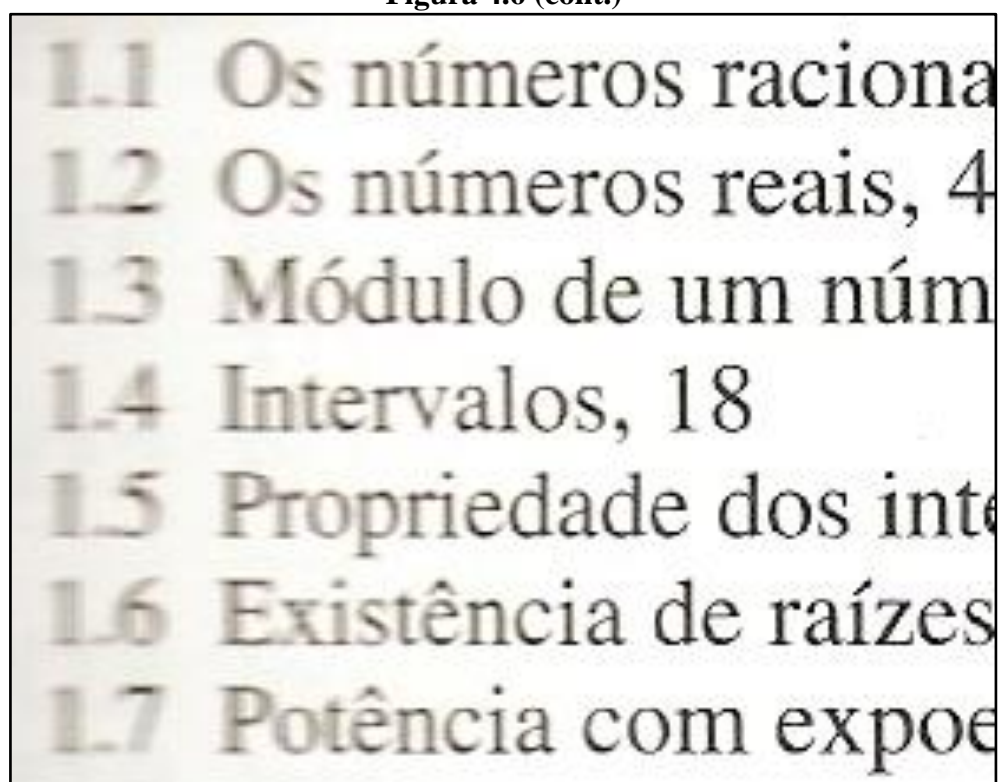
Figura 4.6 (cont.)



We study the performance of a system whose inputs consist of a sum of  $c$  trunks. Voice calls arrive as a Poisson process and call holding times have an exponential distribution. Each voice call,

d. Correção com  $p = 0,25$

Figura 4.6 (cont.)



1.1 Os números racionais, 4  
1.2 Os números reais, 4  
1.3 Módulo de um número, 4  
1.4 Intervalos, 18  
1.5 Propriedade dos intervalos, 4  
1.6 Existência de raízes, 4  
1.7 Potência com expoente, 4

a. Imagem original

Figura 4.7 Resultado do filtro de remoção de embaçamento com  $p = 0,25$

1.1	Os números racionais
1.2	Os números reais, 4
1.3	Módulo de um número
1.4	Intervalos, 18
1.5	Propriedade dos intervalos
1.6	Existência de raízes
1.7	Potência com expoente

b. Correção com janela 3x3

1.1	Os números racionais
1.2	Os números reais, 4
1.3	Módulo de um número
1.4	Intervalos, 18
1.5	Propriedade dos intervalos
1.6	Existência de raízes
1.7	Potência com expoente

c. Correção com janela 5x5

Figura 4.7 (cont.)

1.1	Os números racionais
1.2	Os números reais, 4
1.3	Módulo de um número
1.4	Intervalos, 18
1.5	Propriedade dos intervalos
1.6	Existência de raízes
1.7	Potência com expoente

d. Correção com janela 7x7

**Figura 4.7 (cont.)**

## 5 Remoção de Sombreamento

A introdução desta tese mostrou como a iluminação pode ser irregular nos documentos digitalizados por *scanner* e cameras. Tal característica não é, em vários casos, esteticamente “agradável”, sem contar que deteriora o desempenho dos algoritmos que processam este tipo de imagem.

Este capítulo tem como objetivo mostrar três propostas para resolver este problema, sendo que em uma delas corrige-se documentos que contêm ilustrações, desenhos, gráficos etc. A maioria das propostas na literatura técnica corrigem documentos que contenham apenas texto. O mais novo algoritmo gera imagens de alta qualidade com tempo de processamento relativamente rápido – que no pior caso foi de 11s para uma imagem de 14Mpixels.

### 5.1 Trabalhos relacionados

Zhang e seus colaboradores, na referência (Zhang *et al.*, 2007), extraem a imagem que representa o valor de fundo do papel aplicando um janela grande de suavização. Nas regiões que não fazem parte do fundo (caracteres, desenhos, etc.) é utilizada a estimativa baseada em função radial (do inglês Radial Basis Function). A referência (SILVA *et al.*, 2010) substitui as partes de interferência frente-verso por pixel de fundo calculados através da média dos valores dos pontos vizinhos.

As referências (HORN, 1974; BLAKE, 1985; FUNT *et al.*, 1992; OH *et al.*, 2001) usam a técnica de Retinex (LAND e MCCANN, 1971) para remover o sobreamento com variação suave. Brown e seus colegas (BROWN *et al.*, 2007) resolvem este problema utilizando três dispositivos de captura e uma fonte de laser. Esta fonte obtém a normal de superfície para corrigir a variação de intensidade de luz. Os métodos descritos anteriormente necessitam da configuração de ambiente por pessoas especializadas para corrigir a imagem distorcida. Observa-se que em muitos casos esta correção não é possível, pois a única informação disponível para a correção é a própria imagem do documento.

Como dito anteriormente, não foi encontrado nenhum método que melhore a iluminação de documentos com figuras e desenhos onde a única informação é a imagem do documento, como exemplificado na Figura 1.2.b. Nesse caso, o uso de janela de local não é

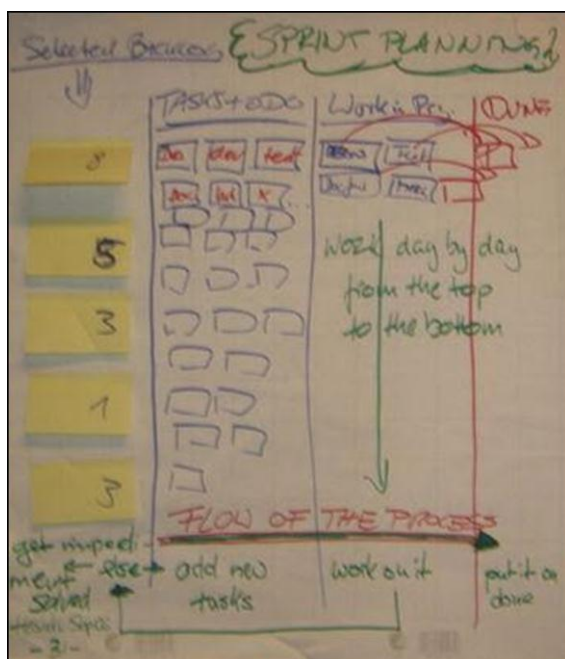
adequado, pois ela pode não ser grande o suficiente para incluir pontos pertencentes ao fundo do documento. Portanto, é necessária uma abordagem global a ser vista nas próximas seções.

## 5.2 Primeiro Algoritmo Proposto

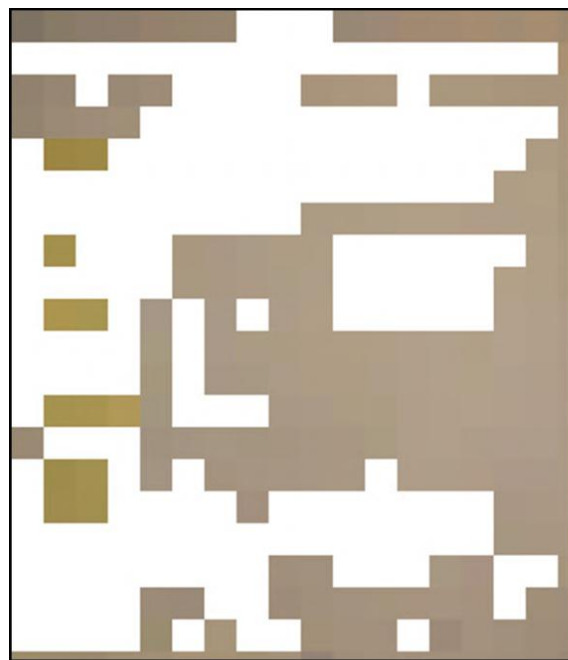
A primeira proposta foi desenvolvida durante o mestrado do autor desta tese (OLIVEIRA e LINS, 2010). A dissertação apresenta o Tableau, que é um sistema desenvolvido para processar quadrados didáticos de cores negra, branca e verde com identificação da borda do quadro, correção de perspectiva e remoção de sombreamento. Este último será apresentado a seguir.

O método consiste na identificação de blocos não-intersectantes “uniformes”, i.e. com pouca variação de cor (Figura 5.1.b). Desses blocos, extrai-se qual pertence ao fundo do quadro a partir da divisão dos blocos em conjuntos disjuntos através de uma regra transitiva que define se dois blocos pertencem ao mesmo conjunto. Nesta, um dos conjuntos define os blocos que fazem parte do fundo do quadro (Figura 5.1.c). Depois extrai-se a cor dos blocos que não fazem parte do fundo através da cor do bloco mais próximo. Com a cor do fundo estimada, é feita a remoção de sombreamento, que está presente na Figura 5.1.d.

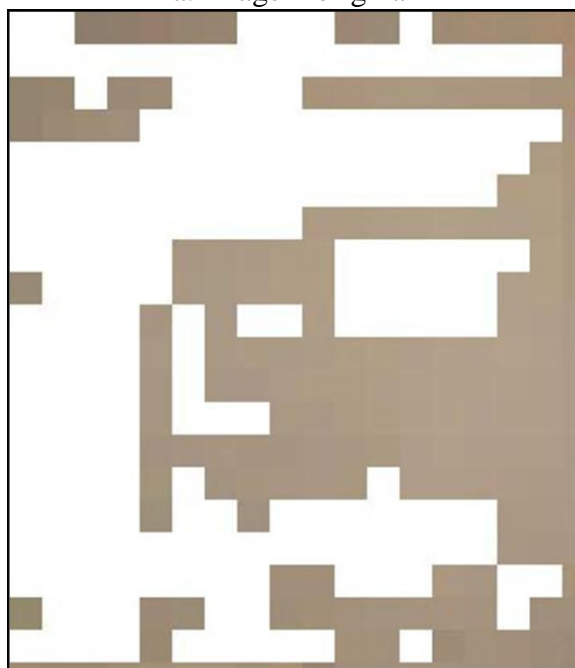
O sistema identifica a cor do quadro e calcula qual a cor mais próxima (branco puro, preto puro e verde) utilizando o mecanismo (CIE, 2001). Uma vez a cor de fundo identificada, o Tableau requisita ao usuário para escolher outra tonalidade ou confirmar o fundo obtido. A Figura 5.2 mostra o resultado do processamento deste algoritmo num quadro verde em (b) com a entrada em (a).



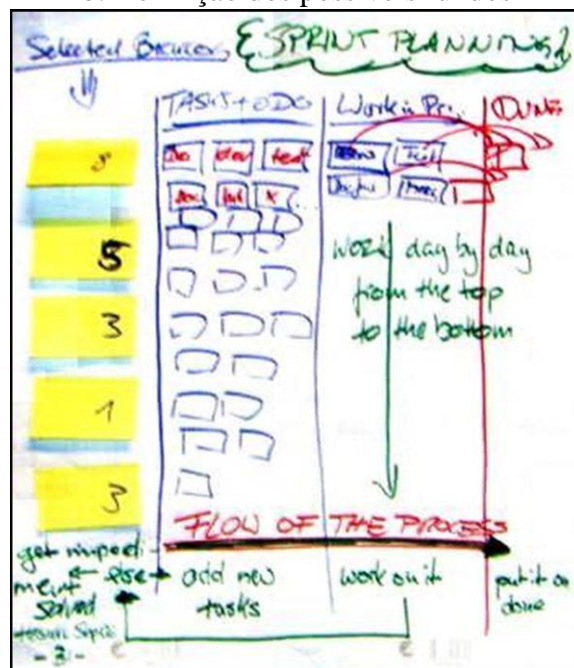
a. Imagem original



b. Definição dos possíveis fundos

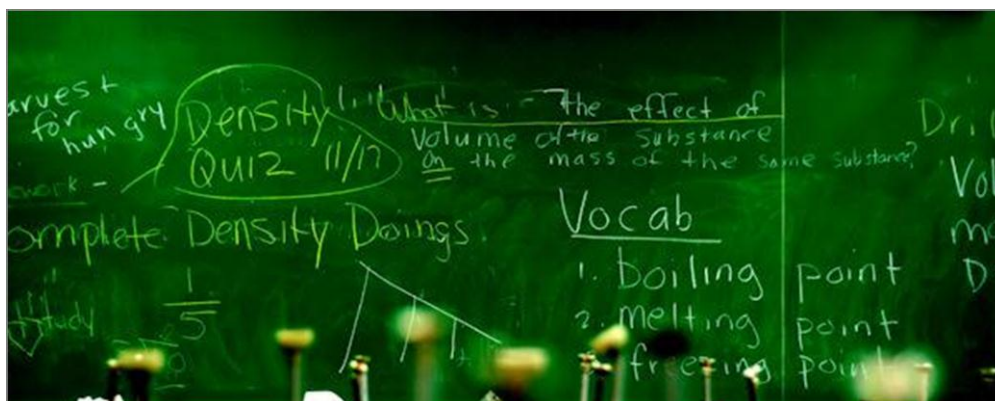


c. Extração de blocos do fundo do quadro

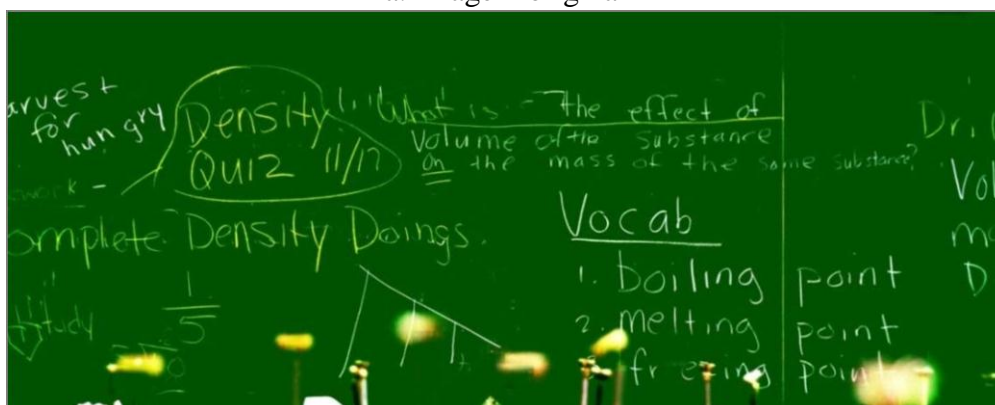


d. Resultado do realce

**Figura 5.1** Remoção de sombreamento aplicada em um quadro branco



a. Imagem original



b. Imagem processada

**Figura 5.2 Remoção de sombreamento aplicada num quadro verde**

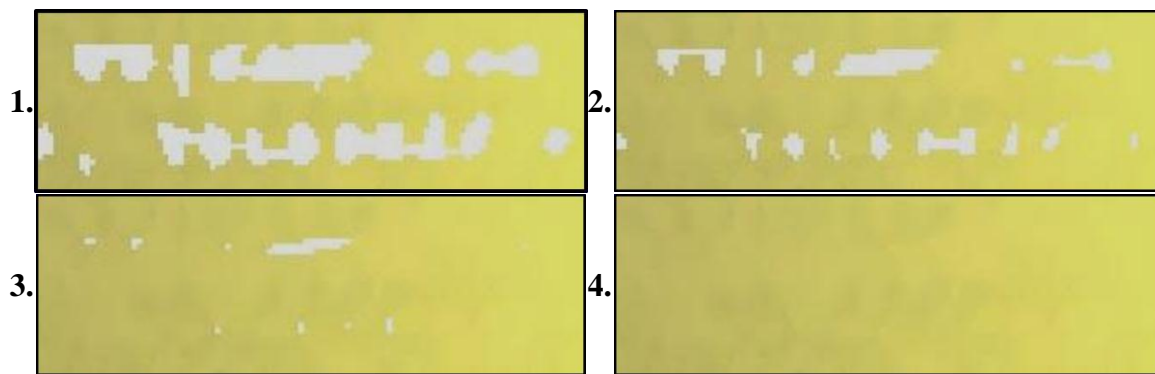
### 5.3 Segundo Algoritmo Proposto

A segunda proposta está publicada em (OLIVEIRA e LINS, 2009) e se baseia na primeira diferenciando-se:

- no uso de blocos intersectantes de pixels entre vizinhos;
- no procedimento de agrupamento de blocos;
- na estimativa das cores dos blocos que não fazem parte do fundo.

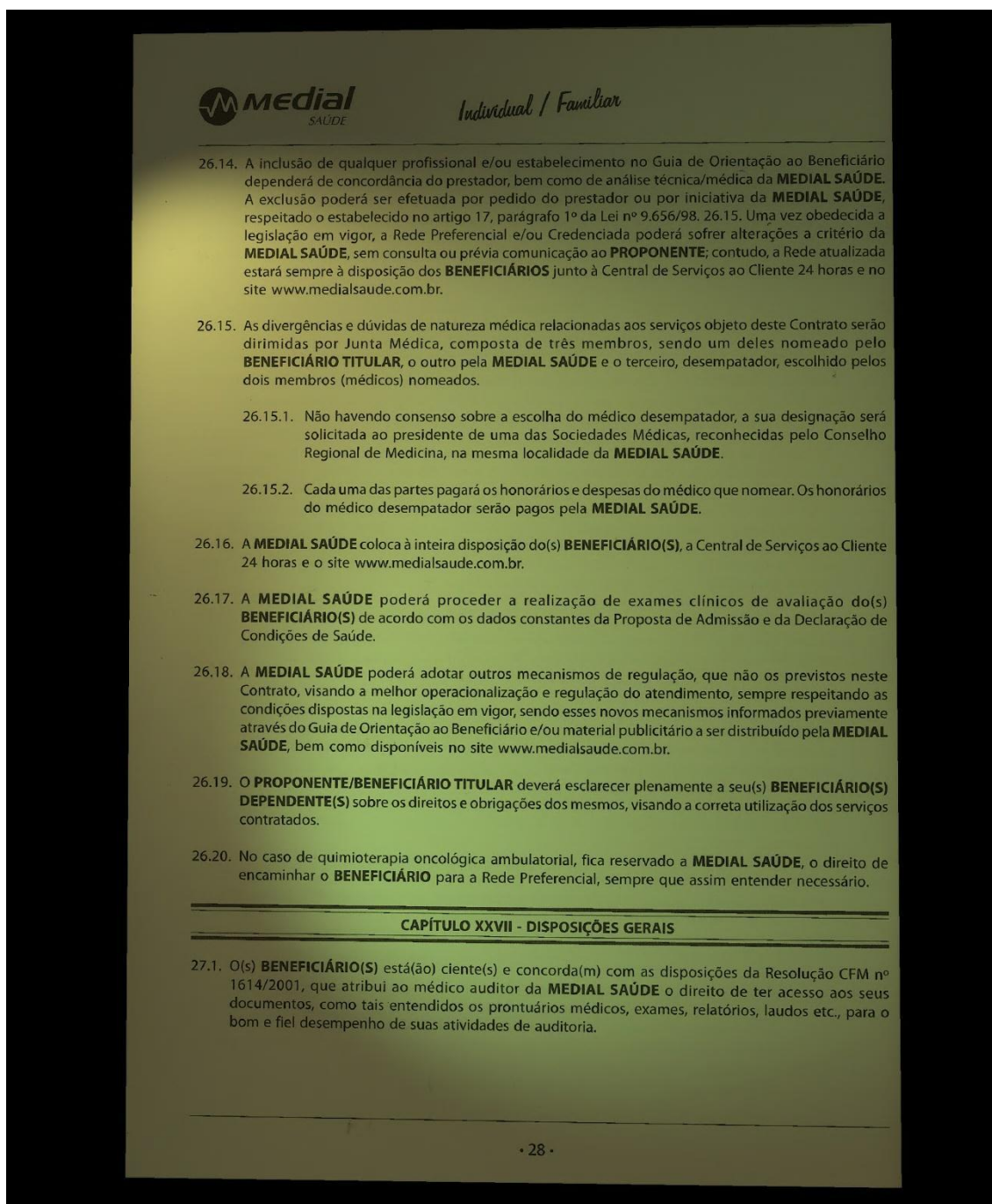
A primeira mudança indica que a imagem é dividida em blocos, mas o tamanho da janela de busca é maior que o tamanho do bloco. Ou seja, há intersecção dos pixels entre as janelas. Com relação ao segundo ponto, o mecanismo de agrupamento anterior ignora a localização dos blocos da imagem, pois compara todos os blocos entre si. Para cada bloco, busca-se os primeiros que estiverem nas direções verticais, horizontais e diagonais de 45° e seguido da aplicação da regra transitiva entre esses dois blocos. Com relação ao procedimento de estimativa de cores, é feita uma “dilatação” iterativa conforme ilustrado na Figura 5.3.



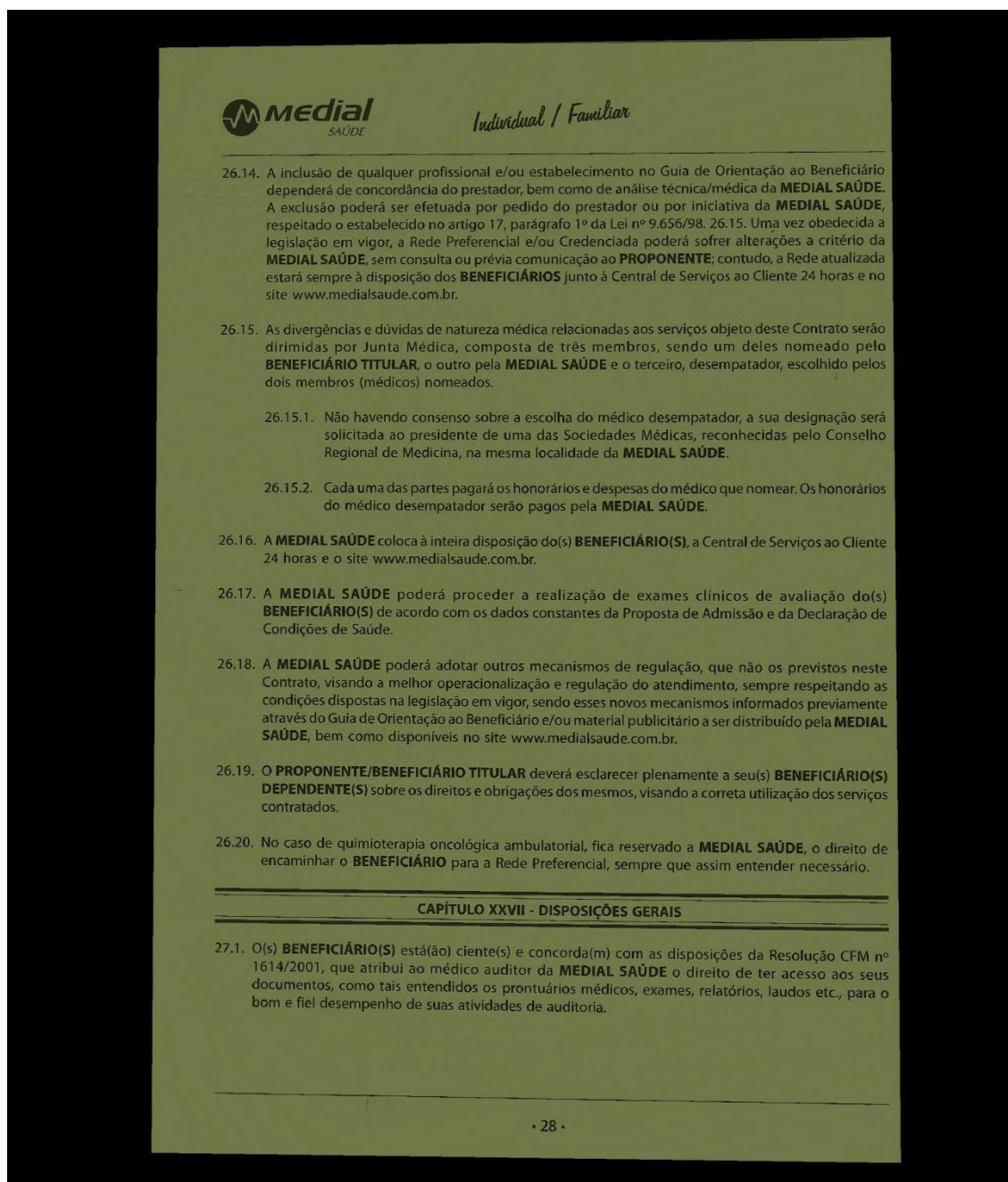


**Figura 5.3 Dilatação iterativa para estimar cor do papel**

A Figura 5.5 mostra a remoção do sobreamento aplicada sobre uma imagem com iluminação gerada por computador (Figura 5.4). A Figura 5.7 mostra o resultado desta proposta sobre a Figura 5.6.



**Figura 5.4 Documento com iluminação gerada por computador**



**Figura 5.5 Processamento da segunda proposta de remoção de sobreposição sobre a Figura 5.4**

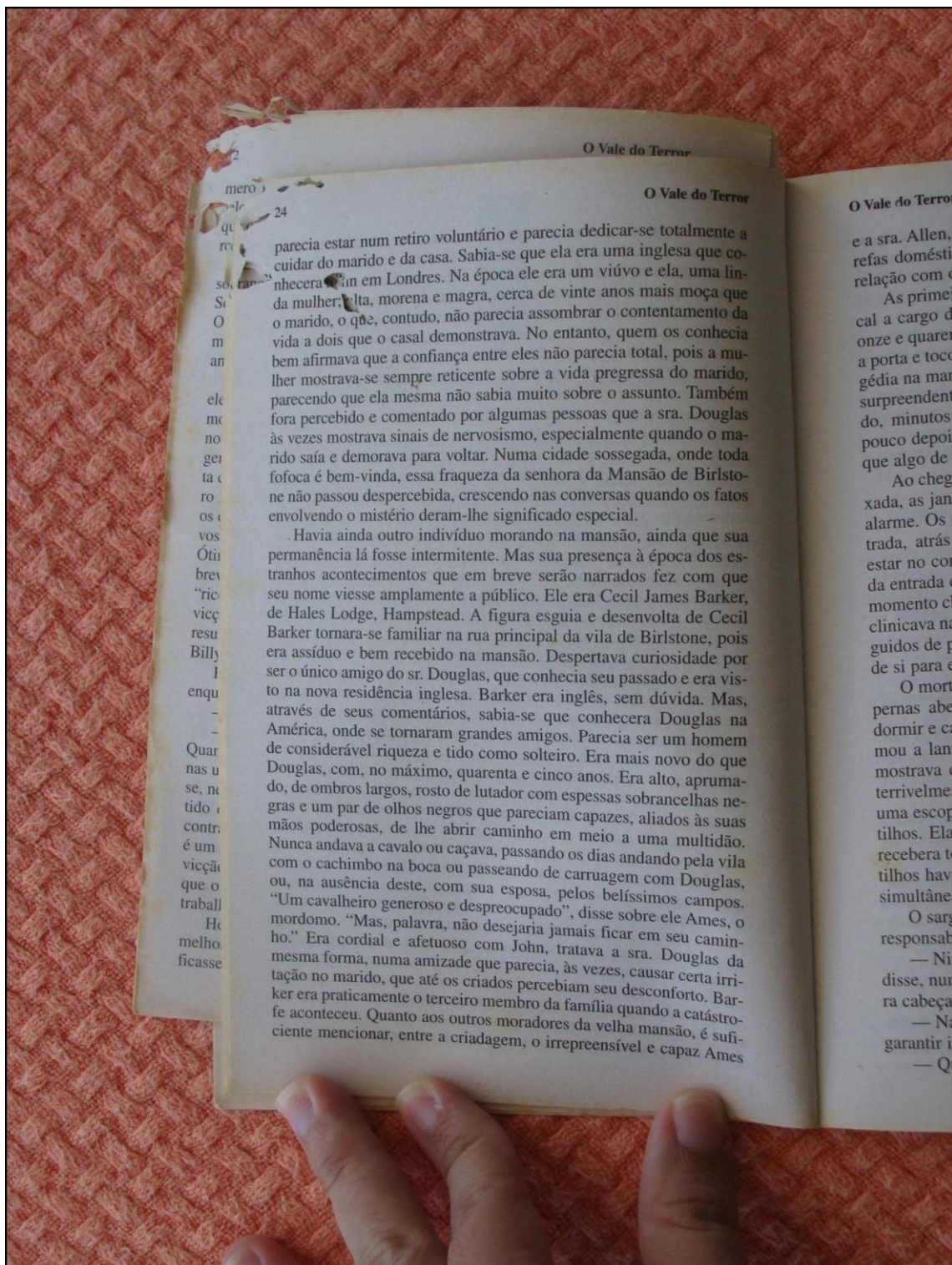
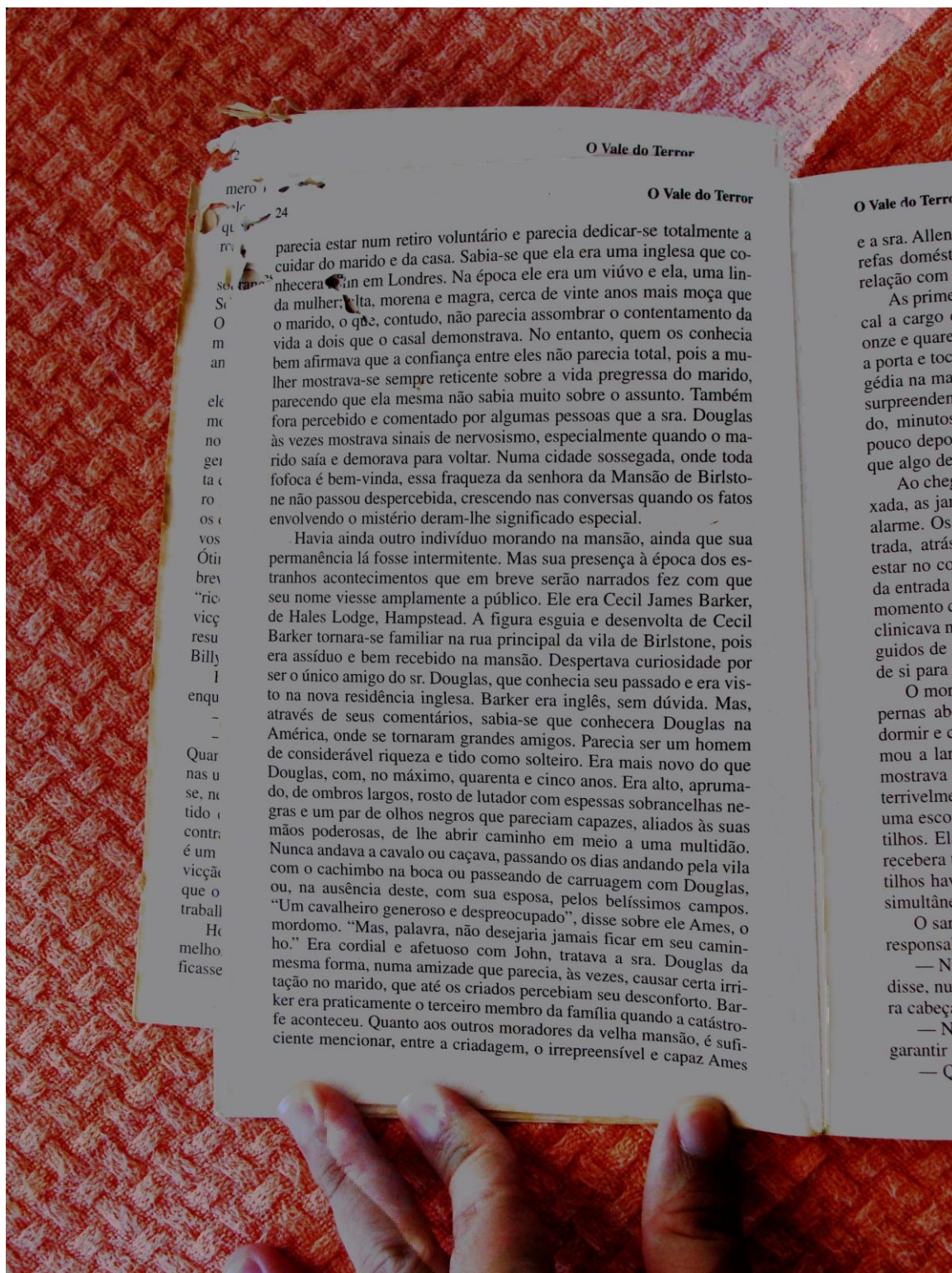


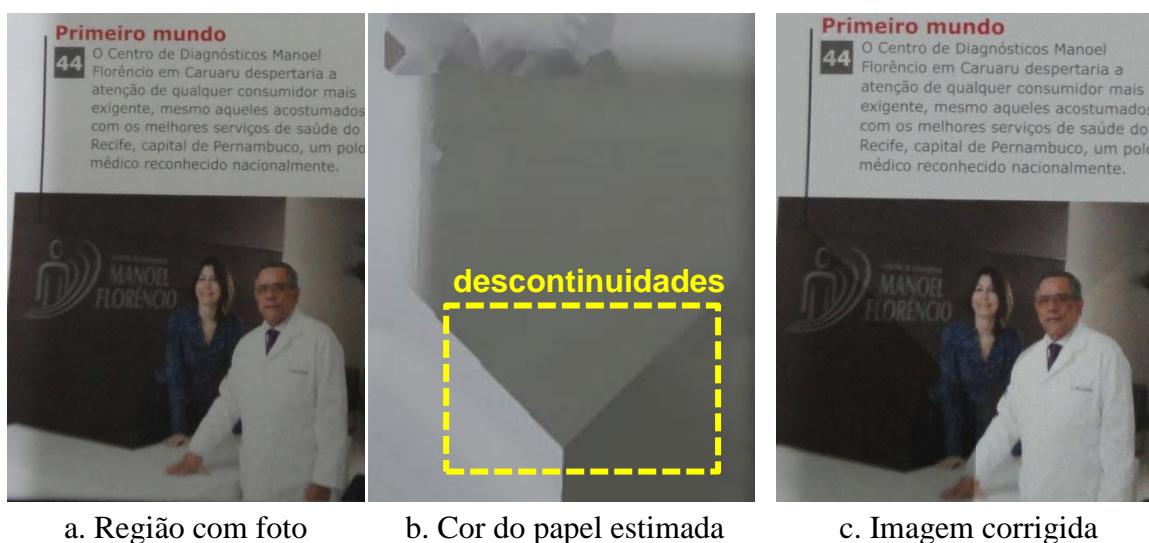
Figura 5.6 Fotografia de página do livro “Sherlock Homes: Vale do Terror”



**Figura 5.7** Processamento da segunda proposta de remoção de sombreamento sobre a Figura 5.6

## 5.4 Terceiro Algoritmo Proposto

O algoritmo anterior não pode ser aplicado quando o documento possui grandes áreas em que a cor do papel precisa ser estimada. A descontinuidade é proveniente da estimativa do valor do papel nessas regiões, como mostrado na Figura 5.8, com imagem distorcida em (a), a estimativa de iluminação em (b) e a correção pela segunda proposta em (c). As descontinuidades também são observadas na região externa ao livro da imagem corrigida, como pode ser observado na Figura 5.7.



**Figura 5.8 Estimativa de sombreamento da segunda proposta**

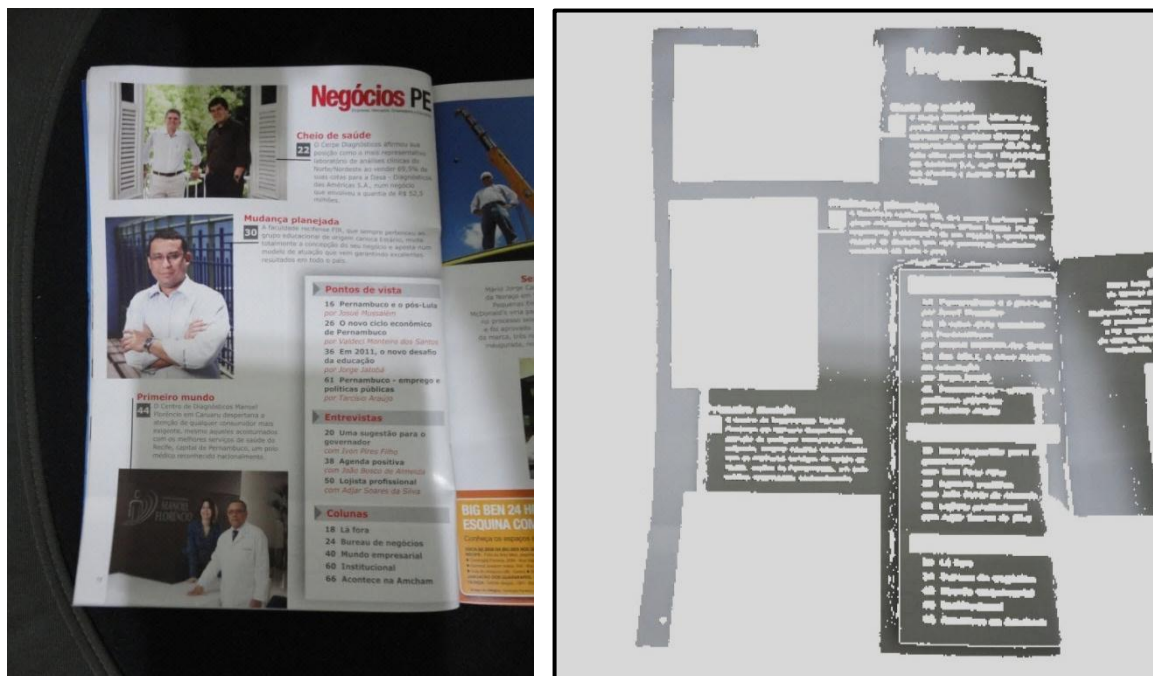
Tendo em vista este problema, a última proposta visa modificar o algoritmo anterior para que possa remover sombreamento de documentos com figuras, desenhos ou gráficos. Para atingir tal objetivo, remodela-se as duas últimas modificações da proposta anterior, ou seja:

- o procedimento de agrupamento de blocos;
- a estimativa das cores dos blocos que não fazem parte do fundo.

Para o primeiro, substitui-se a busca de blocos uniformes vizinhos em várias direções pela busca de todos k-vizinhos usando o algoritmo proposto por (CONNOR e KUMAR, 2010). Segue-se com a aplicação da regra de similaridade entre os blocos para todos os k-vizinhos.

Para o segundo ponto, ao invés da dilatação iterativa, utiliza-se a interpolação do vizinho natural. Segundo Amidror (AMIDROR, 2002), esse é o melhor método para estimar valores de pontos desconhecidos onde a densidade varia bruscamente. Este é o

caso dos documentos tratados nesta seção, pois os blocos do papel possuem uma alta densidade nas regiões de texto e praticamente inexistente nas ilustrações, como pode ser observado na Figura 5.9.



a. Imagem fotografada

b. Regiões identificadas como pertencentes ao papel

**Figura 5.9 Identificação da região pertencente ao papel**

Apesar dessas mudanças parecerem simples, vários procedimentos foram desenvolvidos para viabilizar computacionalmente o novo algoritmo. O mais importante destes foi a definição de um mecanismo de iteração de pixels da imagem em ordem-z. A ordem-z mapeia um ponto em duas ou mais dimensões em um código-z, que é definido pelo entrelaçamento dos *bits* de cada coordenada. A Figura 5.10 mostra o entrelaçamento do ponto (5,3) em coordenadas cartesianas binárias para o código-z.



**Figura 5.10 Entrelaçamento de bits na obtenção do código**

O incremento do código-z produz sobre o espaço cartesiano a forma de Z em várias dimensões. Isto pode ser visto na Figura 5.11. Com este código é possível traduzir a busca ortogonal em 2 dimensões em um conjunto de intervalos de códigos-z (parte vermelha da Figura 5.11). Nessa mesma figura, a parte preta indica os segmentos fora do retângulo de busca e do intervalo-z; a parte cinza indica a região compreendida entre os códigos-z das bordas, mas fora do retângulo de busca. Todos os mecanismos eficientes na literatura técnica para a visita de pontos na ordem-z dentro de um retângulo não informam detalhes suficientes para sua implementação (SKOPAL *et al.*, 2006). No artigo (OLIVEIRA *et al.*, 2011b), que explica esta proposta, são apresentados algoritmos que possibilitam esta busca. A utilização de códigos-z possuem outras utilidades; algumas delas estão descritas em (HAR-PELED, 2011).



**Figura 5.11** Iteração na ordem-z sobre o espaço bidimensional (EPPSTEIN, 2011)

Outros pontos importantes relativos à iteração dos *pixels* em ordem-z para o método proposto são:

- Iterar deste modo tira proveito do conceito de localidade da memória.
- O pré-processamento do algoritmo de busca por todos k-vizinhos, ordena os *pixels* em ordem-z (CONNOR e KUMAR, 2010). Como os blocos são iterados nesta ordem, não há a necessidade deste ordenamento.



- A interpolação do vizinho natural é obtida através da triangulação de Delaunay, esta é construída com complexidade  $O(n)$  se  $n$  pontos forem inseridos em ordem-z (BUCHIN e MULZER, 2009). O método mais usual é a inclusão aleatória dos pontos (BERG *et al.*, 2008) que possui complexidade  $O(n \log n)$ .

Outros procedimentos do método proposto são explicados com mais detalhes em (OLIVEIRA *et al.*, 2011b). Para validar o desempenho do algoritmo, 11 imagens escaneadas com baixo nível de distorção foram obtidas e combinadas com 3 padrões de iluminação gerados por computador, totalizando 33 imagens com iluminação irregular. Para cada imagem distorcida foi obtida uma de referência sem este tipo de distorção. A Figura 5.12 ilustra esse procedimento. A Figura 5.13 mostra como as medidas SSIM entre as imagens processadas e de referência são obtidas e comparadas.



Figura 5.12 Criação da base de dados com iluminação gerada por computador

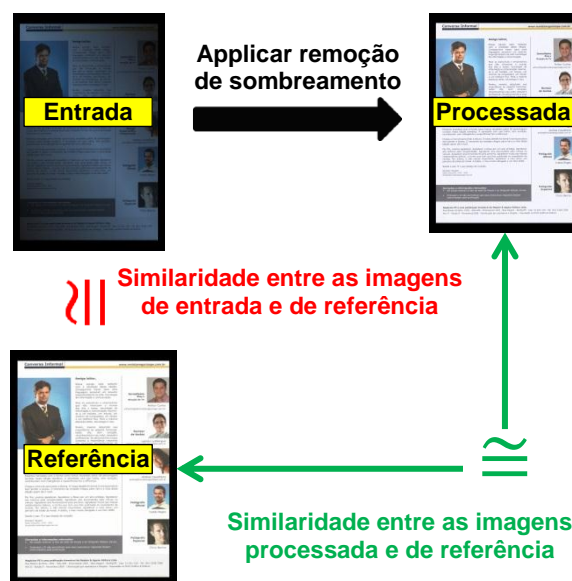


Figura 5.13 Obtenção de medidas de similaridades




A Tabela 5.1 mostra a média e desvio padrão das similaridades SSIM, agrupadas por modelo de iluminação, entre a imagem de referência com a de Entrada (segunda linha) e processada (terceira linha), sendo o percentual de ganho definido pela equação (5.1) presente na última linha da tabela. O ganho foi definido de maneira análoga na validação do algoritmo de remoção de embaçamento pela equação (4.3).

$$\text{Ganho} = \frac{\text{Similaridade entre imagens corrigidas e de referência}}{\text{Similaridade entre imagens distorcidas e de referência}} - 1 \quad (5.1)$$

Na Tabela 5.1 é possível observar o ótimo desempenho do algoritmo pois as similaridades SSIM para cada modelo possuem:

- médias elevadas e com desvios padrões baixos;
- alto valor de ganho, sendo o maior ganho médio de 68,0%.

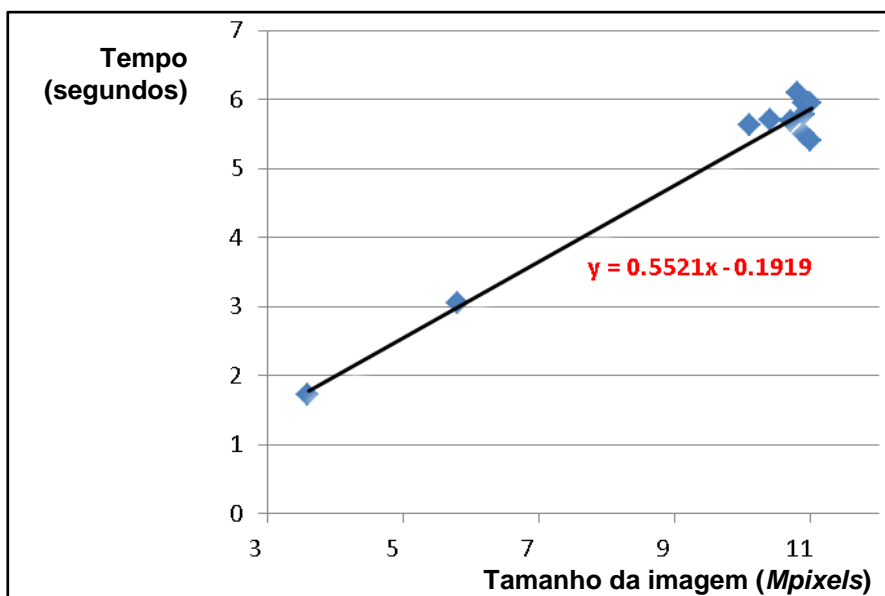
**Tabela 5.1 Qualidade (SSIM) do algoritmo proposto**

Modelo de iluminação				
SSIM Relativa à referência	Entrada (média $\pm \sigma$ )	0,572 $\pm$ 0,048	0,814 $\pm$ 0,018	0,951 $\pm$ 0,006
	Processada (média $\pm \sigma$ )	0,952 $\pm$ 0,014	0,970 $\pm$ 0,011	0,981 $\pm$ 0,004
	Ganho (média $\pm \sigma$ )	68,0% $\pm$ 13,5%	19,5% $\pm$ 3,1%	3,2% $\pm$ 0,6%

A Tabela 5.2 mostra o tempo médio de processamento do algoritmo agrupado por imagem sem distorção, que varia linearmente de acordo com o tamanho da imagem. A Figura 5.14 mostra o gráfico com pontos definidos por estas duas medidas. Uma regressão linear foi realizada, o que reforça a idéia da relação linear. No entanto, a comprovação merece um estudo mais aprofundado para verificar a relação entre diferentes estruturas de documentos (e.g. documento com uma ou mais figuras) e o seu tempo de processamento pelo algoritmo proposto. Esta sugestão é oportuna, pois existe uma linha de pesquisa que trata da reestruturação da triangulação de *Delaunay* quando existe grande variação na densidade dos pontos (HAR-PELED, 2011); o desempenho de alguns tipos de algoritmos sobre uma estrutura neste cenário pode ser degradado.

**Tabela 5.2 Tempo (em segundos) do processamento do algoritmo**

Imagem sem distorção	Tamanho médio (Mpixels)	Tempo (em segundos)	
		Média	Desv. Padrão
beige_pag014_livememory	5,8	3,042	0,442
white_digitalizar0001	10,1	5,637	0,781
white_digitalizar0003	10,9	5,777	0,495
white_digitalizar0004	10,8	6,094	0,589
white_digitalizar0005	11,0	5,398	0,426
white_digitalizar0006	10,9	5,949	0,580
white_digitalizar0007	10,4	5,702	0,398
white_digitalizar0009	10,7	5,684	0,530
white_digitalizar0013	11,0	5,954	0,528
white_digitalizar0017	10,9	5,502	0,494
white_pag625_livememory	3,6	1,726	0,166

**Figura 5.14 Relação entre o tamanho da imagem e o tempo de processamento**

A Figura 5.17 mostra um exemplo de resultado de processamento sobre a Figura 5.16 para validar a terceira proposta, com os valores do índice SSIM entre a imagem corrigida e de referência (presente na Figura 5.15). A Figura 5.18 mostra o resultado da remoção de sombreamento sobre a Figura 5.9. As Figuras 5.20 e 5.22 mostram resultados de processamento em documentos fotografados com a câmera Sony Cybershot DSC-W350 como pode ser observado nas Figuras 5.19 e 5.21, respectivamente. Em todos os casos, observa-se que há uma grande variação de iluminação e o processamento praticamente removeu essa distorção.

## Enxergando longe

Delta Café na edição 8

**N**a edição que circulou em outubro de 2008, a Negócios PE mostrava os empreendedores João Barbosa e Eduardo Gusmão difundindo a mais importante marca de café de Portugal, a Delta, que chegou ao Brasil por Recife, investindo no canal franquia e fechando aquele ano com um faturamento superior a R\$ 2 milhões, lucratividade líquida média de 15%, e a presença da marca Delta em mais de 100 estabelecimentos da Região Nordeste, entre hotéis e restaurantes.

Hoje o projeto de franquias evoluiu para Deltaexpresso Coffee Convenience Store, uma franquia que tem quatro negócios no mesmo espaço físico – cafeteria, empório de cafés, loja Delta Q e espaço para cursos e degustação de café. Atualmente são nove operações em funcionamento e duas lojas que devem abrir até fevereiro, além dos cerca de 150 estabelecimentos que comercializam a marca Delta Café. Vivendo um momento excelente, a aposta do grupo é no Projeto Deltaexpresso 2014, que prevê a abertura de 150 lojas em todas as capitais brasileiras sedes da Copa do mundo 2014.



Capa edição 8



**"A Negócios PE se transformou num canal de visibilidade e de aproximação entre os empreendedores do Estado. A publicação é bastante ousada em sua opção editorial e, neste contexto de crescimento que vivemos em Pernambuco, torna-se uma importante fonte de informações e um filtro interessante da conjuntura econômica local. Parabéns pelo trabalho e força para continuar perseguindo o sucesso!"**

**João Barbosa**  
Empreendedor da Delta Expresso

30

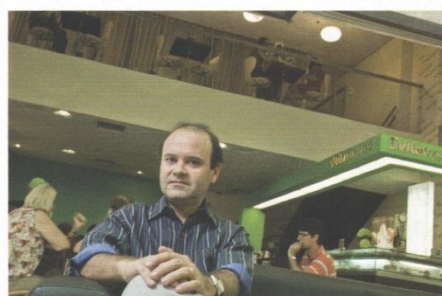


Figura 5.15 Imagem de referência a partir de uma página da revista Negócios PE

## Enxergando longe

### Delta Café na edição 8

**N**a edição que circulou em outubro de 2008, a Negócios PE mostrava os empreendedores João Barbosa e Eduardo Gusmão difundindo a mais importante marca de café de Portugal, a Delta, que chegou ao Brasil por Recife, investindo no canal franquia e fechando aquele ano com um faturamento superior a R\$ 2 milhões, lucratividade líquida média de 15%, e a presença da marca Delta em mais de 100 estabelecimentos da Região Nordeste, entre hotéis e restaurantes.

Hoje o projeto de franquias evoluiu para Deltaexpresso Coffee Convenience Store, uma franquia que tem quatro negócios no mesmo espaço físico – cafeteria, empório de cafés, loja Delta Q e espaço para cursos e degustação de café. Atualmente são nove operações em funcionamento e duas lojas que devem abrir até fevereiro, além dos cerca de 150 estabelecimentos que comercializam a marca Delta Café. Vivendo um momento excelente, a aposta do grupo é no Projeto Deltaexpresso 2014, que prevê a abertura de 150 lojas em todas as capitais brasileiras sedes da Copa do mundo 2014.



Capa edição 8



"A Negócios PE se transformou num canal de visibilidade e de aproximação entre os empreendedores do Estado. A publicação é bastante ousada em sua opção editorial e, neste contexto de crescimento que vivemos em Pernambuco, torna-se uma importante fonte de informações e um filtro interessante da conjuntura econômica local. Parabéns pelo trabalho e força para continuar perseguindo o sucesso!"

**João Barbosa**  
Empreendedor da Delta Expresso

30

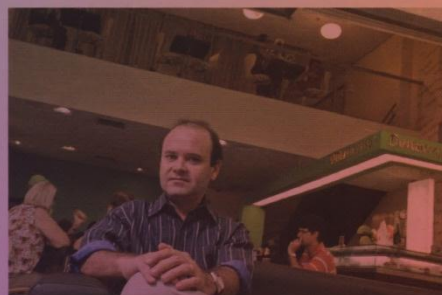


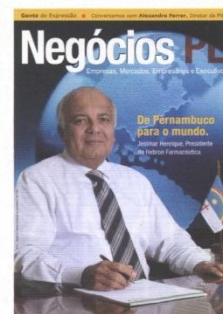
Figura 5.16 Imagem com distorção relativa a Figura 5.15

## Enxergando longe

Delta Café na edição 8

**N**a edição que circulou em outubro de 2008, a Negócios PE mostrava os empreendedores João Barbosa e Eduardo Gusmão difundindo a mais importante marca de café de Portugal, a Delta, que chegou ao Brasil por Recife, investindo no canal franquia e fechando aquele ano com um faturamento superior a R\$ 2 milhões, lucratividade líquida média de 15%, e a presença da marca Delta em mais de 100 estabelecimentos da Região Nordeste, entre hotéis e restaurantes.

Hoje o projeto de franquias evoluiu para Deltaexpresso Coffee Convenience Store, uma franquia que tem quatro negócios no mesmo espaço físico – cafeteria, empório de cafés, loja Delta Q e espaço para cursos e degustação de café. Atualmente são nove operações em funcionamento e duas lojas que devem abrir até fevereiro, além dos cerca de 150 estabelecimentos que comercializam a marca Delta Café. Vivendo um momento excelente, a aposta do grupo é no Projeto Deltaexpresso 2014, que prevê a abertura de 150 lojas em todas as capitais brasileiras sedes da Copa do mundo 2014.



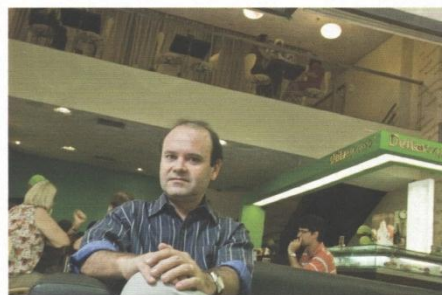
Capa edição 8



**"A Negócios PE se transformou num canal de visibilidade e de aproximação entre os empreendedores do Estado. A publicação é bastante ousada em sua opção editorial e, neste contexto de crescimento que vivemos em Pernambuco, torna-se uma importante fonte de informações e um filtro interessante da conjuntura econômica local. Parabéns pelo trabalho e força para continuar perseguindo o sucesso!"**

**João Barbosa**  
Empreendedor da Delta Expresso

30



**Figura 5.17** Processamento pela terceira proposta na Figura 5.16 com similaridades SSIM relativas a imagem de referência da Figura 5.15 iguais a 96,90% (Verm.), 97,57% (Verde), 95,68% (Azul) e 97,96% (Cinza)



Figura 5.18 Processamento da terceira proposta de remoção de sombreamento sobre a Figura 5.9



Figura 5.19 Página de uma edição da revista Negócios PE



**NEGÓCIOS**  
Empresas, Mercados, Empreendedores e Negócios

**saúde**  
agnósticos afirmou sua  
mo o mais representati  
de análises clínicas de  
deste ao vender 69,5%  
para a Dasa - Diagnós  
cas S.A., num negócio  
veu a quantia de R\$ 52,3

que sempre pertenceu  
em carioca Estácio, mud  
o seu negócio e aposta  
em garantindo excelente



**Matéria de capa**  
**54** Ele tem 48 anos, quatro  
stents no peito e uma  
história que serve de  
exemplo e reflexão  
para quem pensa em  
empreender. Até chegar ao  
bem-sucedido momento  
que vivencia como  
empresário, comandando  
a Bezerra Guindastes,  
empresa que está sentindo  
no bolso a intensificação  
da atividade econômica  
estruturadora de  
Pernambuco e faturou R\$  
30 milhões em 2010 .

**tos de vista**  
Pernambuco e o pós-  
*osué Mussalém*  
O novo ciclo econômico  
ernambuco  
*Valdeci Monteiro dos S*  
Em 2011, o novo des  
ducação  
*Jorge Jatobá*  
Pernambuco - empre  
ticas públicas  
*Tarcísio Araújo*

**trevistas**  
Uma sugestão para o  
vernador  
*Ivon Pires Filho*  
Agenda positiva  
*João Bosco de Almeida*  
Lojista profissional  
*Adjar Soares da Silva*

**olunas**  
Lá fora  
Bureau de negócios  
Mundo empresarial  
stitucional Amchar

**Senhor dos hambúrgueres**  
Mário Jorge Carvalheira dirigia em 1991 a filial da Noração em São Paulo quando leu na revista Pequenas Empresas Grandes Negócios que o McDonald's viria para Pernambuco. Ele se inscreveu no processo seletivo para escolha do franqueado e foi aprovado. Hoje comanda quatro operações da marca, três no Shopping Recife e uma, recém-inaugurada, no Shopping Difusora, em Caruaru.



**BIG BEN 24 HORAS: DOMINGOS FERREIRA, ESQUINA COM BENVINDA DE FARIAS EM BOA VIAGEM.**  
Começa os espaços exclusivos da Big Ben, Amigo do Alérgico\* e Viva Bem Diabético.\*\*

**ORGANIZAÇÃO DO GUARANAPÉ:** Pernambuco: Avenida de Melo, 4186 - Paulista - CEP: 51030-100 - Recife - PE  
Belo Horizonte: Rua da Bahia, 1727 - Casa Caída - CEP: 31030-000 - Belo Horizonte - MG  
Belo Horizonte: Rua da Bahia, 1727 - Casa Caída - CEP: 31030-000 - Belo Horizonte - MG  
Belo Horizonte: Rua da Bahia, 1727 - Casa Caída - CEP: 31030-000 - Belo Horizonte - MG



Figura 5.20 Processamento da terceira proposta de remoção de sombreamento sobre a Figura 5.19



Figura 5.21 Duas páginas de uma edição da revista Negócios PE

## Enxergando longe

Delta Café na edição 8

**N**a edição 8, em outubro de 2014, a Negócios PE mostrava os empreendedores João Barbosa e Estelaro Galvão, afiliados a mais importante franquia de café de Portugal, a Delta, que chegou ao Brasil por Recife, investindo no comércio líquido média de 15%, e a abertura de mais de 100 estabelecimentos da região Nordeste, entre hotéis e restaurantes.

Hoje o projeto de franquias evoluiu para Deltaexpresso Coffee Convenience Store, uma franquia que tem quatro negócios no mesmo espaço físico - cafeteria, empório de café, loja Delta Q e ponto de vendas para cursos e distribuição de café. Atualmente são cerca de 150 estabelecimentos que comercializam a marca Delta Café. Vivendo um momento recorde, o grupo é no Projeto Deltaexpresso 2014, que prevê a abertura de 120 lojas em todas as capitais brasileiras sedes da Copa do mundo 2014.



**PARABÉNS Revista Negócios PE!**  
3 anos informando melhor, o Mercado Empresarial Pernambucano.

Av. Rio Branco, 243 - 2º andar - Bairro do Recife - Recife Antigo - CEP 50030-310  
Fone: 81 4009.7181 - [www.agemar.com.br](http://www.agemar.com.br) - [comercial@agemar.com.br](mailto:comercial@agemar.com.br)

# Negócios PE em



## é com a Agemar

A melhor solução para sua empresa!

Uma das principais operadoras portuárias e de logística do Recife - Pernambuco - Brasil, a Agemar é a opção certa para armazenar, movimentar e manusear sua carga com agilidade e segurança. Disponíveis também, de locação de containers adaptados, venda e locação de equipamentos, com ou sem operador.

- Locação de Containers;
- Locação de Equipamentos;
- Operações Portuárias;
- Operação de postos de combustíveis em Recife e Fernando de Noronha;
- Logística e Armazéns Gerais;
- Transporte Marítimo.




Figura 5.22 Processamento da terceira proposta de remoção de sombreamento sobre a Figura 5.21

## 6 Conclusões

A digitalização de documentos por *scanners* e câmeras digitais portáteis é cada vez mais frequente. Isso ocorre devido à redução do custo dos mesmos e ao fácil compartilhamento de imagens pela Internet. Esta tese apresentou vários algoritmos para melhorar a qualidade de imagens de documentos obtidas por esses equipamentos.

Apresentaram-se algoritmos para melhoria de imagens de documentos digitalizados com os seguintes problemas: distorção geométrica, iluminação irregular e embaçamento. Observou-se na introdução desta tese que as distorções de imagens fotografadas são mais evidentes e difíceis de tratar que nas imagens digitalizadas por *scanners*.

Para corrigir a distorção geométrica em imagens digitalizadas por *scanners*, foi apresentada uma proposta que usa uma imagem de calibração para inferir a distorção geométrica do documento a partir do sombreamento da imagem. Outra proposta foi desenvolvida com o fim de corrigir a distorção geométrica quando a junção de páginas é ortogonal ou paralela à direção de varredura, sendo tal abordagem inédita na literatura técnica. Para tal solução, fez-se necessário o desenvolvimento de algoritmos de segmentação de linhas de texto. As duas propostas implementadas demonstraram desempenho superior às publicadas até o presente momento tendo como conjunto de testes uma base de imagens disponível publicamente.

No caso da iluminação irregular presente em documentos fotografados, as três propostas apresentadas nesta tese obtiveram resultados satisfatórios. A última com alta taxa de desempenho em documentos mais complexos que os apresentados por trabalhos similares presentes na literatura técnica.

Para corrigir o embaçamento (*blur*), foi proposto nesta tese um algoritmo simples que pode ser implementado diretamente em imagens em tons de cinza. O mesmo se adapta bem quando a imagem possui embaçamento não uniforme.

Um ponto importante de vantagem dos algoritmos aqui apresentados é o tempo de processamento ser menor que dos outros métodos disponíveis, possibilitando que os usuários executem os algoritmos em até 12s.

## 6.1 Trabalhos futuros

Diversas melhorias podem ser aplicadas na pesquisa apresentada nesta tese. Com relação à segmentação, sugere-se como trabalho futuro o desenvolvimento de um novo algoritmo com base em imagens em tons de cinza ou coloridas, pois a binarização muitas vezes sobre/sub segmenta a imagem, degradando o desempenho de algoritmos que a utilizam.

Sobre a correção geométrica, pode-se ter como objetivo o desenvolvimento de um novo algoritmo para documentos fotografados utilizando as linhas de texto como base para inferir o modelo tridimensional do documento. Recomenda-se, também, utilizar esses algoritmos para pré-processar antes de realizar o reconhecimento óptico de caracteres (OCR).

A correção de iluminação irregular pode ser aprimorada com o desenvolvimento de um *software* em que usuários sem conhecimentos técnicos possam utilizar. Nesse sistema seria possível modificar os parâmetros do algoritmo, remover/incluir blocos uniformes e até modificar o sombreamento inferido. Atualmente, o usuário executa a correção pela linha de comando do Windows. Propõe-se também o desenvolvimento de um mecanismo de binarização em documentos que contenham ilustrações utilizando alguns princípios elucidados nas remoções do sombreamento. Sugere-se também um estudo sobre a comparação dos tamanhos de arquivos antes e após à remoção de sombreamento sobre imagens de quadros didáticos em vários formatos (e.g. PNG, JPEG, GIF) e quantidades de cores; esta sugestão é oportuna pois as imagens das lousas possuem baixa variação de cor após a correção.

O embaçamento pode ser aperfeiçoado com o desenvolvimento de um novo esquema para imagens coloridas. A sua aplicação não é simples, pois cada ponto da imagem apresenta uma cor com três valores correlacionados; analisar os canais RGB da imagem de forma independente pode gerar imagens com “aberrações” visuais.

Por fim, considerando que um documento apresenta na mesma imagem diversas distorções tratadas aqui, é sugerido um estudo sobre qual a melhor ordem de aplicação da remoção dessas distorções. O orientador do autor desta tese e colaboradores identificam se uma determinada imagem possui ou não certos defeitos (LINS *et al.*, 2010d), mas não indicam qual a melhor ordem de aplicação dos algoritmos que removem os defeitos. Este estudo é sugerido como uma nova linha de pesquisa.

## REFERÊNCIAS

- (AMIDROR, 2002) I. Amidror. Scattered data interpolation methods for electronic imaging systems: A survey. *J. Electron. Imaging* 11 (2): 157–176.
- (ÁVILA e LINS, 2005) B. T. Ávila e R. D. Lins. A fast orientation and skew detection algorithm for monochromatic document images. *Proceedings of the ACM Symposium on Document Engineering*, pp. 118-126, 2005.
- (BERG *et al.*, 2008) M. de Berg, O. Cheong, M. van Kreveld e M. Overmars. *Computational Geometry: Algorithms and Applications*, 3rd Edition, pages 99-101. Springer, 2008.
- (BLAKE, 1985) A. Blake. Boundary Conditions for Lightness Computation in Mondrian World. *Computer Vision, Graphics and Image Processing* 32: 314-327.
- (BRINGHURST, 2004) R. Bringhurst. *The Elements of Typographic Style*. 3rd edition, Hartley & Marks Publishers, 2004.
- (BROWN *et al.*, 2007) M. S. Brown, M. Sun, R. Yang, L. Yun e W. B. Seales. Restoring 2D Content from Distorted Documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, no. 11. pp. 1904-1916.
- (BUCHIN e MULZER, 2009) K. Buchin e W. Mulzer. Delaunay Triangulations in  $O(\text{sort}(n))$  Time and More. *50th Annual IEEE Symposium on Foundations of Computer Science*, pp. 139-148, Washington DC, EUA, 2009.
- (BUKHARI *et al.*, 2008) S. S. Bukhari, F. Shafait e T. M. Breuel. Segmentation of curled textlines using active contours. In *Proceedings 8th IAPR Workshop on Document Analysis Systems*, pp. 270–277, Nara, Japão, 2008.
- (BUKHARI *et al.*, 2009a) S. S. Bukhari, F. Shafait e T. M. Breuel. Coupled snakelet model for curled textline segmentation of camera-captured document images. In *Proceedings 10th International Conference on Document Analysis and Recognition*, pp. 61–65, Barcelona, Espanha, 2009.
- (BUKHARI *et al.*, 2009b) S. S. Bukhari, F. Shafait e T. M. Breuel. Ridges based curled textline region detection from grayscale camera-captured document images. In *Proc. The 13th International Conference on Computer Analysis of Images and Patterns*, LNCS 5702, pp. 173–180, Springer Verlag, 2009.
- (BUKHARI *et al.*, 2009c) S. S. Bukhari, F. Shafait e T. M. Breuel. Textline information extraction from grayscale camera-captured document images. In *Proc. The 13th International Conference on Image Processing*, Cairo, Egito, 2009.

- (BUKHARI *et al.*, 2009d) S. S. Bukhari, F. Shafait e T. M. Breuel. Dewarping of Document Images using Coupled-Snakes. CBDAR 2009, pp. 17-24, Barcelona, Espanha, 2009.
- (BUKHARI *et al.*, 2010a) S. S. Bukhari, F. Shafait e T. M. Breuel. Performance Evaluation of curled textline segmentation algorithms on CBDAR 2007 dewarping contest dataset. 17th Int. Conf. on Image Processing, ICIP'10. Hong Kong, China, 2010.
- (BUKHARI *et al.*, 2010b) S. S. Bukhari, F. Shafait e T. M. Breuel. Performance Evaluation of Curled Textlines Segmentation Algorithms, 9th IAPR Workshop on Document Analysis Systems, 2010.
- (BUKHARI *et al.*, 2011) S. S. Bukhari, F. Shafait e T. M. Breuel. Text-Line Extraction using a Convolution of Isotropic Gaussian Filter with a Set of Line Filters. 11th International Conference on Document Analysis and Recognition. Beijing, China, 2011.
- (CHANDRASHEKARA *et al.*, 2002) R. Chandrashekara, R. H. Mohiaddin e D. Rueckert. Analysis of myocardial motion in tagged MR images using nonrigid image registration. *Progress in biomedical optics and imaging*, 3(2):1168–1179, 2002.
- (CIE, 2001) Commission Internationale de l'Éclairage (CIE). Improvement to industrial colour-difference evaluation. CIE Publication No. Central Bureau of the CIE, Vienna, 2001.
- (CONNOR e KUMAR, 2010) M. Connor e P. Kumar. Fast construction of k-nearest neighbor graphs for point clouds. *IEEE Transactions on Visualization and Computer Graphics* 99 (PrePrints), pp. 599-608, 2010. Código fonte disponível em <http://sites.google.com/a/compgeom.com/stann/>.
- (DEMOMENT, 1989) G. Demoment. Image reconstruction and restoration: Overview of common estimation structures and problems. *IEEE Transactions on Acoustics, Speech, & Signal Processing*, 37(12), 2024–2036, 1989.
- (DUROU *et al.*, 2007) J. D. Durou, M. Falcone e M. Sagona. Numerical Methods for Shape-from-shading: A New Survey with Benchmarks. *Computer Vision and Image Understanding*, 2007.
- (EPPSTEIN, 2011) D. Eppstein. Four levels of the Z curve, showing the square that is eventually filled by the curve. Disponível em [http://en.wikipedia.org/wiki/File:Four-level\\_Z.svg](http://en.wikipedia.org/wiki/File:Four-level_Z.svg).
- (FAN, 2007) J. Fan. Enhancement of Camera-captured Document Images with Watershed Segmentation, CBDAR 2007, pp. 87-93, Curitiba, Brasil, 2007.

- (FAN, 2009) J. Fan. Robust Color Image Enhancement of Digitized Books, Proceedings of International Conference on Document Analysis and Recognition, pp. 561-565, IEEE Press, Barcelona, Espanha, 2009.
- (FRENCH, 2006) N. French. InDesign Type: Professional Typography with Adobe InDesign CS2. pp. 219, Adobe Press, 2006.
- (FU *et al.*, 2007) B. Fu, M. Wu, R. Li, W. Li e Z. Xu. A model-based book de-warping method using text line detection. In 2nd Int. Workshop on Camera-Based Document Analysis and Recognition, Curitiba, Brasil, 2007.
- (FUNT *et al.*, 1992) B. V. Funt, M. S. Drew e M. Brockington. Recovering Shading from Color Images. Proc. European Conf. Computer Vision. LNCS 588, pp. 124-132, Santa Margherita Ligure, Itália, 1992.
- (GATOS *et al.*, 2004) B. Gatos, I. Pratikakis e S.J. Perantonis. An Adaptive Binarization Technique for Low Quality Historical Documents. DAS VI, LNCS 3163/2004, pp. 102-113.
- (GATOS *et al.*, 2006) B. Gatos, I. Pratikakis e S.J. Perantonis. Adaptive degraded document image binarization. Pattern Recognition, Vol. 39 (3), Março 2006, pp. 317-327.
- (GONZALEZ e WOODS, 2008) R. C. Gonzalez e R. E. Woods. Digital Image Processing. 3rd ed. Prentice Hall, 2008.
- (HAR-PELED, 2011) S. Har-Peled. Data structures for geometric approximation. American Mathematical Society, 2011.
- (HORN, 1970) B. K. P Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. PhD Thesis, MIT, 1970.
- (HORN, 1974) B. K. P Horn. Determining Lightness from an Image. Computer Vision, Graphics and Image Processing 3: 277-299, 1974.
- (JOSHI, 2008) N. S. Joshi. Enhancing photographs using content-specific image priors. Phd thesis, University of California, San Diego, EUA, 2008.
- (KASS *et al.*, 1988) M. Kass, A. Witkin e D. Terzopoulos. Snakes: Active contour models. International Journal of Computer Vision. IEEE Press, pp. 312-331, 1987.
- (LAND e MCCANN, 1971) E. H. Land e J. J. McCann. Lightness and Retinex Theory. J. Optical Soc. Am. 61, pp. 1-11, 1971.



- (LEWAND, 2000) R. Lewand. Cryptological Mathematics. The Mathematical Association of America, 2000. ISBN 978-0883857199.
- (LINS *et al.*, 2007) R. D. Lins, A. R. G. Silva e G. F. P. Silva. Enhancing Document Images Acquired Using Portable Digital Cameras. Em: ICIAR 2007, LNCS, Springer Verlag, pp. 1229-1241, 2007.
- (LINS, 2009) R. D. Lins. A Taxonomy for Noise Detection in Images of Paper Documents - The Physical Noises. ICIAR 2009. LNCS v. 5627. p. 844-854, Springer Verlag, 2009.
- (LINS *et al.*, 2009) R. D. Lins, G. Torreão e G. P. F. Silva. Content Recognition and Indexing in the LiveMemory Platform. GREC 2009, pp. 220-230, Springer Verlag, 2009.
- (LINS *et al.*, 2010a) R. D. Lins, D. M. Oliveira, G. Torreão, J. Fan e M. Thielo. A Dewarping Algorithm to Compensate Volume Binding Distortion in Scanned Documents. ACM SAC 2010, 2010.
- (LINS *et al.*, 2010b) R. D. Lins, D. M. Oliveira, G. Torreão, J. Fan e M. Thielo. Correcting Book Binding Distortion in Scanned Documents. Proceedings of ICIAR 2010. LNCS 6112, pp. 355-365, Springer Verlag, 2010.
- (LINS *et al.*, 2010d) R. D Lins, G. F. P. Silva, S. Banergee, A. Kuchibhotla e M. Thielo. Automatically Detecting and Classifying Noises in Document Images, ACM-SAC 2010, ACM Press, v.1. p.33 – 39, 2010.
- (LUCY, 1974) L. B. Lucy. An iterative technique for the rectification of observed distributions. *Astronomical Journal* 79 (6), pp. 745–754, 1974. doi:10.1086/111605.
- (MAKHOUL *et al.*, 2010) J. Makhoul, F. Kubala, R. Schwartz e R. Weischedel. Performance measures for information extraction. Proc. DARPA Workshop on Broadcast News Understanding. San Mateo, CA: Morgan Kaufmann, Mar., pp. 249–252, 1999.
- (MASALOVITCH e MESTETSKIY, 2007) A. Masalovitch e L. Mestetskiy. Usage of continuous skeletal image representation for document images de-warping. Proceedings of International Workshop on Camera-Based Document Analysis and Recognition, Curitiba, 2007, pp. 45-53.
- (NAJAIM & AGUIAR Ltd, 2011a) Najaim & Aguiar Ltd. Negócios PE – Empresas, Mercado, Empresários e Executivos. 19th edition. Najaim & Aguiar Ltd, Recife, Brasil, 2011.

- (NAJAIM & AGUIAR Ltd, 2011b) Najaim & Aguiar Ltd. Negócios PE – Empresas, Mercado, Empresários e Executivos. 18th edition. Najaim & Aguiar Ltd, Recife, Brasil, 2011.
- (NAYLOR, 2010) M. Naylor. Typographic line terms. Disponível em [http://en.wikipedia.org/wiki/File:Typography\\_Line\\_Terms.svg](http://en.wikipedia.org/wiki/File:Typography_Line_Terms.svg).
- (NEELAMANI *et al.*, 1989) R. Neelamani, H. Choi e R. G. Baraniuk. Wavelet-based deconvolution for ill-conditioned systems. Proc. of IEEE ICASSP, Vol. 6, pp. 3241–3244, 1999.
- (O’GORMAN, 1993) L. O’Gorman. The Document Spectrum for Page Layout Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 15(11), p. 1162-1173, 1993;
- (OH *et al.*, 2001) B. M. Oh, M. Chen, J. Dorsey e F. Durand. Image-Based Modeling and Photo Editing. Proc. Int’l Conf. Computer Graphics and Interactive Techniques, pp. 433-442, 2001.
- (OLIVEIRA e LINS, 2007) D. M. Oliveira e R. D. Lins. Tableau – Processing Teaching-board Images Acquired with Portable Digital Cameras. Em: Proceedings of Second International Workshop on Camera-Based Document Analysis and Recognition, p.79-86, IAPR, 2007.
- (OLIVEIRA e LINS, 2008) D. M. Oliveira e R. D. Lins. Improving the Border Detection and Image Enhancement Algorithms in Tableau. ICIAR 2008, LNCS 5112, pp. 1111-1121, Springer Verlag, 2008.
- (OLIVEIRA e LINS, 2009) D. M. Oliveira e R. D. Lins. A New Method for Shading Removal and Binarization of Documents Acquired with Portable Digital Cameras. Em: Proceedings of Third International Workshop on Camera-Based Document Analysis and Recognition, pp.3-10, 2009.
- (OLIVEIRA e LINS, 2010) D. M. Oliveira e R. D. Lins. Generalizing Tableau to Any Color of Teaching Boards. In Proceedings of ICPR 2010. pp.2411-2414, 2010.
- (OLIVEIRA e LINS, 2011) D. M. Oliveira e R. D. Lins. Shading Removal of Camera Documents with Natural Neighbor Interpolation. (Submetido em 10 de Junho de 2011 ao IJDAR).
- (OLIVEIRA *et al.*, 2010a) D. M. Oliveira, R. D. Lins, G. Torreão, J. Fan e M. Thielo. A New Method for Text-line Segmentation for Warped Documents. Proceedings of ICIAR 2010. LNCS 6112, pp. 398-408, Springer Verlag, 2010.

- (OLIVEIRA *et al.*, 2010b) D. M. Oliveira, R. D. Lins, G. Torreão, J. Fan and M. Thielo. A New Deblurring Algorithm for Textual Document Images. Proceedings of the International Telecommunications Symposium, Manaus, Brasil, 2010.
- (OLIVEIRA *et al.*, 2011a) D. M. Oliveira, R. D. Lins, G. Torreão, J. Fan and M. Thielo. A New Algorithm for Segmenting Warped Text-lines in Document Images. ACM SAC 2011, Taichung, Taiwan, 2011.
- (OLIVEIRA *et al.*, 2011b) D. M. Oliveira, R. D. Lins, G. Torreão, J. Fan and M. Thielo. An Efficient Algorithm for Segmenting Warped Text-lines in Document Images. (Submetido em 6 de Julho de 2011 ao IJDAR).
- (OLIVEIRA *et al.*, 2012) D. M. Oliveira, R. D. Lins e G. S. Torreão. Characters' Shape Analysis of Printed Documents. (Submetido para ser publicado no ICPR 2012).
- (OLIVEIRA *et al.*, 2011c) D. M. Oliveira, R. D. Lins e G. S. Torreão. Deblurring Textual Document Images. (Submetido para ser publicado no LNCS do GREC 2011).
- (OLSON e DELEN, 2008) D. L. Olson e D. Delen. Advanced Data Mining Techniques. Springer Verlag. 1ª edição, Fevereiro de 2008, pp. 138.
- (PIERCE, 1980) J. R. Pierce. An Introduction to Information Theory: Symbols, Signals and Noise. Dover Publications, Inc., New York second edition, p. 75, 1980.
- (PINTUS *et al.*, 2009) R. Pintus, T.Malzbender, O.Wang, R.Bergman, H.Nachlieli e G. Ruckenstein. Photo Repair and 3D Structure from Flatbed *Scanners*. Computer Vision Theory and Applications, pp. 40-50, 2009.
- (RAMSAK, 2002) F. Ramsak. Towards a general-purpose, multidimensional index: Integration, Optimization e Enhancement of UB-Trees. Munich University of Technology. PhD Thesis. Munique, 2002.
- (RICHARDSON, 1972) W. H. Richardson. Bayesian-Based Iterative Method of Image Restoration. JOSA 62 (1), pp. 55–59, 1972. doi:10.1364/JOSA.62.000055.
- (RILEY, 1987) M. D. Riley. Time-frequency representation for speech signals. PhD Thesis, MIT, 1987.
- (ROTH *et al.*, 2005) S. Roth e M. J. Black. Fields of experts: a framework for learning image priors. CVPR, Vol. 2, pp. 860–867, 2005.
- (RUDIN *et al.*, 1992) L. I. Rudin, S. Osher e E. Fatemi. Nonlinear total variation based noise removal algorithms. Physica D, 60, 259–268, 1992.

- (SAUVOLA e PIETIKAINEN, 2000) J. Sauvola e M. Pietikainen. Adaptive document image binarization. *Pattern Recognition* 33 (2), pp. 225–236, 2000.
- (SCHUBERT, 2000) R. Schubert. Using a Flatbed *Scanner* as a Stereoscopic Near-Field Camera, *IEEE Computer Graphics and Applications*, pp. 38-45, 2000.
- (SHAFAIT e BREUEL, 2007) F. Shafait and T. M. Breuel. Document Image Dewarping Contest. 2nd Int. Workshop on Camera-Based Document Analysis and Recognition, CBDAR 2007, pp 181-188, Curitiba, Brasil, 2007.
- (SHAFAIT *et al.*, 2008) F. Shafait, D. Keysers e T. M. Breuel. Performance evaluation and benchmarking of six page segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):941–954, 2008.
- (SILVA e LINS, 2005) A. R. G. Silva e R. D. Lins. Background Removal of Document Images Acquired Using Portable Digital Cameras. LNCS 3656, pp. 278-285, Toronto, Canadá, 2005.
- (SILVA e LINS, 2007) G. F. P. Silva e R. D. Lins. PhotoDoc: A Toolbox for Processing Document Images Acquired Using Portable Digital Cameras. CBDAR 2007, pp. 107-115, Curitiba, Brasil, 2007.
- (SILVA *et al.*, 2010) G. F. P. Silva, R. D. Lins e J. M. M. da Silva. HistDoc - A Toolbox for Processing Images of Historical Documents. In *International Conference on Image Analysis and Recognition 2*, pp. 409-419, Póvoa de Varzim, Portugal, 2010.
- (SILVA, 2006) A. R. G. Silva. Análise e Melhoria da Qualidade de Documentos Fotografados. Dissertação de Mestrado. Universidade Federal de Pernambuco, Departamento de Eletrônica e Sistemas, Pernambuco, Recife, 2006.
- (SKOPAL *et al.*, 2006) T. Skopal, M. Kratky, V. Snasel e J. Pokorny. A New Range Query Algorithm for the Universal B-trees. *Information Systems* 31(6), pp. 489-511, Oxford, Reino Unido, 2006.
- (STAMATOPOULOS *et al.*, 2008) N. Stamatopoulos, B. Gatos, I. Pratikakis e S. J. Perantonis. A two-step de-warping of camera document images. *Proceedings 8th IAPR Workshop on Document Analysis Systems*, pp. 209–216, Nara, Japão, 2008.
- (SUZUKI *et al.*, 2007) Y. Suzuki, A. Yamashita e T. Kaneko. Correction of Geometric and Photometric Distortion of Document Images Using a Stereo Camera System. *Proceedings of the IAPR Conference on Machine Vision Applications (IAPR MVA 2007)*, pp. 215-218, Tóquio, Japão, 2007. ISBN 978-4-901122-07-8.

- (SZELISKI, 2010) R. Szeliski. Computer Vision: Algorithms and Applications. Springer Verlag, 1ª edição. Rascunho disponível em <http://research.microsoft.com/en-us/um/people/szeliski/Book/>.
- (TAN *et al.*, 2006) C. L. Tan, L. Zhang, Z. Zhang e T. Xia. Restoring Warped Document Images through 3D Shape Modeling, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 28(2), pp. 195-208, 2006.
- (UKIDA e KONISHI, 2001) H. Ukida e K. Konishi. 3D Shape Reconstruction Using Three Light Sources in Image Scanner. IEICE Trans. on Inf. & Syst., Vol.E84-D, No. 12, pp.1713-1721, 2001.
- (UKIDA *et al.*, 2007) H. Ukida, Y. Tanimoto, T. Sano e H. Yamamoto. 3D Shape Reconstructions Using Image Scanner under Various Number of Illuminations. IEEE IST 2007, pp. 1-6, Krakow, Polônia, 2007.
- (ULGES *et al.*, 2005) A. Ulges, C. H. Lampert e T. M. Breuel. Document image dewarping using robust estimation of curled text lines. Proc. Eighth Int. Conf. on Document Analysis and Recognition, 2005, pp. 1001–1005.
- (WANG *et al.*, 2004) Z. Wang, A. C. Bovik, H. R. Sheikh e E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing 13(4), pp. 600-612, 2004. Disponível em <https://ece.uwaterloo.ca/~z70wang/research/ssim/>.
- (WOLFRAM, 2009) Wolfram Research. Least Squares Fitting--Exponential. Acessado no dia 14 de Outubro de 2009. Disponível em <http://mathworld.wolfram.com/LeastSquaresFittingExponential.html>.
- (ZHANG *et al.*, 1999) R. Zhang, P. S. Tsai, J. E. Cryer e M. Shah. Shape from shading: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(8), 1999, pp. 690–706, 1999.
- (ZHANG *et al.*, 2007) L. Zhang, A. M. Yip e C. L. Tan. Photometric and geometric restoration of document images using inpainting and shape-from-shading. 22nd Conference on Artificial Intelligence: 1121-1126, Vancouver, Canadá, 2007.
- (IMAGEJ, 2011) ImageJ. Disponível em <http://rsbweb.nih.gov/ij/>. Acessado em 14 de abril de 2012.

## APÊNDICE A

### A.1 Publicações sobre segmentação de linhas de texto

D. M. Oliveira, R. D. Lins, G. Torreão, J. Fan e M. Thielo. *A New Method for Text-line Segmentation for Warped Documents*. Proceedings of ICIAR 2010. LNCS 6112, pp. 398-408, Springer Verlag, 2010.

D. M. Oliveira, R. D. Lins, G. Torreão, J. Fan and M. Thielo. *A New Algorithm for Segmenting Warped Text-lines in Document Images*. ACM SAC 2011, Taichung, Taiwan, 2011.

D. M. Oliveira e R. D. Lins. *An Efficient Algorithm for Segmenting Warped Text-lines in Document Images*. (Submetido em 6 de Julho de 2011 ao IJDAR).

### A.2 Publicação sobre estudo estatístico sobre elementos tipográficos de texto

D. M. Oliveira, R. D. Lins e G. S. Torreão. *Characters' Shape Analysis of Printed Documents*. (Submetido para ser publicado no ICPR 2012).

### A.3 Publicações sobre correção geométrica

R. D. Lins, D. M. Oliveira, G. Torreão, J. Fan e M. Thielo. *A Dewarping Algorithm to Compensate Volume Binding Distortion in Scanned Documents*. ACM SAC 2010, 2010.

R. D. Lins, D. M. Oliveira, G. Torreão, J. Fan e M. Thielo. *Correcting Book Binding Distortion in Scanned Documents*. Proceedings of ICIAR 2010. LNCS 6112, pp. 355-365, Springer Verlag, 2010.

### A.4 Publicação sobre remoção de embaçamento

D. M. Oliveira, R. D. Lins, G. Torreão, J. Fan and M. Thielo. *A New Deblurring Algorithm for Textual Document Images*. Proceedings of the International Telecommunications Symposium, Manaus, Brasil, 2010.

D. M. Oliveira, R. D. Lins e G. S. Torreão. Deblurring Textual Document Images (Submetido para ser publicado no LNCS do GREC 2011).

## **A.5 Publicações sobre remoção de sombreamento**

D. M. Oliveira e R. D. Lins. *A New Method for Shading Removal and Binarization of Documents Acquired with Portable Digital Cameras*. Em: Proceedings of Third International Workshop on Camera-Based Document Analysis and Recognition, pp.3-10, 2009.

D. M. Oliveira e R. D. Lins. *Generalizing Tableau to Any Color of Teaching Boards*. In Proceedings of ICPR 2010. pp.2411-2414, 2010.

D. M. Oliveira e R. D. Lins. *Shading Removal of Camera Documents with Natural Neighbor Interpolation*. (Submetido em 10 de Junho de 2011 ao IJDAR).

D. M. Oliveira, R. D. Lins, G. Torreão, J. Fan e M. Thielo. *A New Method for Text-line Segmentation for Warped Documents*. Proceedings of ICIAR 2010. LNCS 6112, pp. 398-408, Springer Verlag, 2010.



# A New Method for Text-Line Segmentation for Warped Documents

Daniel M. Oliveira<sup>1</sup>, Rafael D. Lins<sup>1</sup>, Gabriel Torreão<sup>1</sup>,  
Jian Fan<sup>2</sup>, and Marcelo Thielo<sup>3</sup>

<sup>1</sup> Universidade Federal de Pernambuco, Departamento de Eletrônica e Sistemas.  
Av. Prof. Luiz Freire, s/n, Cidade Universitária  
50740-540, Recife, PE, Brasil  
{rdl,daniel.moliveira,gabriel.dsilva}@ufpe.br

<sup>2</sup> Hewlett-Packard Labs, Palo Alto, USA  
jian.fan@hp.com

<sup>3</sup> Hewlett-Packard Labs, Porto Alegre, Brazil  
marcelo.resende.thielo@hp.com

**Abstract.** Bound documents either scanned or captured with digital cameras often present a geometrical warp that makes text-lines curled. The identification of text-lines is one of the steps for document de-warping when only a single image is available. This paper presents a new method for text-line segmentation. It is based on a simple, but effective, skew detector proposed by Ávila-Lins and simplifies the idea of coupled snakes introduced by Bukhari to a moving parallel line regression. The proposed method performed better than the best of the similar algorithms in the literature.

**Keywords:** Text-line segmentation, document de-warping, layout analysis.

## 1 Introduction

The digitalization of bound documents, such as books, either performed by flatbed scanners or digital cameras often yields images that exhibit a geometrical distortion in the region close to the book spine. Such distortion not only makes more difficult the document reading for humans, but also degrades OCR performance. Text-line envelope segmentation is one of the pre-processing steps for many algorithms. The segmentation process can be accompanied of a baseline and/or mean line estimation, Figure 1 illustrates these typographic lines and others.

Bukhari, Shafait, and Breuel [9] introduce the concept of baby snakes for extraction of text-lines. Later on, they use coupled *snakelets* [7] for the same purpose. Finally, in references [8][10] they obtain text-lines by ridges detection on grayscale images. Coupled snakes are used for base/mean line estimation. De-warping is done by calculating y-coordinates using upper/lower neighboring lines followed by a perspective correction.

Stamatopoulos and his colleagues [6] detect text-lines and estimate a document 3D model by approximating the border lines with a polynomial of degree three. Fu *et al*



**Fig. 1.** Font typographic lines following reference [14]

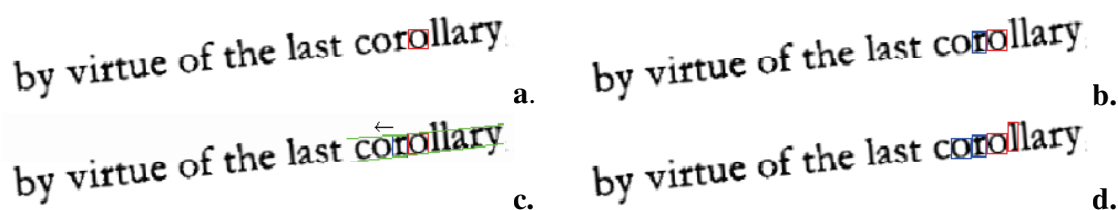
[2] estimate border points fitting them in 3D-cylinder model. Masalovitch and Mes-tetskiy [1] estimate the spaces between lines; a bezier patch is built and followed by de-warping procedure.

This paper proposes new text-line segmentation method. It borrows ideas from a simple but effective skew detector proposed by Ávila-Lins [3] and coupled snakes introduced by Bukhari, Shafait, and Breuel [7] and can be used regardless document orientation. Reference [4] uses the new method presented herein for de-warping scanned documents. It is organized as follows. Section 2 presents details of the new algorithm. Section 3 shows that the proposed algorithm benchmarked on the CBDAR 2007 de-warping contest test-set achieved an accuracy rate of 91.10% with an under segmentation rate of 1.81%, while the performance of algorithm by Bukhari, Shafait, and Breuel [7], the best algorithm in the literature yields 89.65% and 3.30%, respectively. Section 4 concludes this works.

## 2 Segmenting Text-Lines

The proposed algorithm improves the Ávila-Lins skew detection scheme [3] for text-line segmentation which is summarized below as illustrated in Figure 2. Black and white images are assumed as input for the algorithm.

1. Component labeling transforming components as enclosing blocks (Figure 2.a);
2. For each unvisited block B do:
  - a. Locate the nearest unvisited neighbor block N of block B (Figure 2.b);
  - b. Group a text-line starting from blocks B and N (Figure 2.c-d) forming up-  
per/lower or right/left lines;
  - c. Detect the skew angle and landscape/portrait orientation of the document;
  - d. Detect the up-down orientation of the document;
3. Detect total document rotation;



**Fig. 2.** Ávila-Lins skew detection [3]

This (in C language) algorithm is fast. For a 200dpi scanned image, it takes 115ms on a Pentium IV of 2.4GHz and 512MB of RAM with accuracy of 98%. Despite the good results in skew detection, text-line segmentation requires a more robust approach.

## 2.1 Letter/Line Properties

Let  $V$  be the vector formed by the extreme points of a line. The slope of  $V$ , called the  $V$  angle, is considered to be the angle that the text-line forms with the horizontal line. If the absolute value for the x-component is greater than the y-component of this vector, the line is considered to be horizontal, otherwise it is vertical. The Text-line length is set to  $V$  length.

**Table 1.** Letter properties definition

<b>Letter property</b> \ <b>Orientation</b>	<b>Horizontal</b>	<b>Vertical</b>
<i>Letter case height</i>	Block height	Block width
<i>Letter case width</i>	Block width	Block height
<i>Letter case top point</i>	Upper middle point	Left middle point
<i>Letter case bottom point</i>	Lower middle point	Right middle point

One may notice that letter properties are subject to orientation, thus term “block” is used for the “letter case”, the enclosing box relative to image; term “letter” depends on the document orientation as described in Table 1. The discrimination between the character height and width is useful as the width is less stable than the height due to: variable font width values and character merging caused by digitization and/or binarization (e.g. see “precondition” and “width” of Fig. 3).

For the steps presented in the next section, some terms are underlined in a high level language with a more precise definition below; the ratio function is defined by (1):

- Small block – Box where width and height have less than 6 pixels;
- Similar size letters – Letters N and M have similar size if the  $ratio(N_{height}, M_{height}) \geq 0.6$  and  $ratio(N_{width}, M_{width}) \geq 0.1$ .
- Parallel lines with offset – an offset of 40% relative to y-axis intercept (i.e.  $|b_t - b_d|$ ) is added to top/down lines;
- Smaller than window mean widths/heights – Letter N and properties window W with  $ratio(N_{height}, W_{\overline{height}}) < 0.6$  and  $N_{width} \leq W_{\overline{width}}$ .
- Maximum distance between letters – 2.50 times window mean height;
- Search for text-line upwards/downwards – search range are limited to 3 times the height of a letter;

$$ratio(a, b) = \min(a, b) / \max(a, b) . \quad (1)$$

## 2.2 The New Approach

The main idea of the new method proposed here is to group together characters with same properties by “walking” through the document to form a text-line. Instead of coupled snakes [7], moving parallel straight lines are used; Section 2.2.1 explains how to obtain them. As the warp level may distort character sizes, a moving letter window is used while a text-line is formed; A window of length of 7 components was used herein. Listing 1 summarizes the whole procedure.

### Listing 1

1. Label components transforming them as enclosing boxes;
2. Remove small blocks or if a block encloses another totally;
3. For each block  $B$  do (term “block” is used; orientation is not available):
  - a. Locate the nearest neighbor block  $N$  of block  $B$ ;
  - b. If  $N$  and  $B$  have similar sizes, place them in  $Q_{NEIGHBORS}$  priority queue, with  $priority = ratio(N_{width}, B_{width}) + ratio(N_{height}, B_{height})$ ;
4. While  $Q_{NEIGHBORS}$  is not empty
  - a. Pull-out neighbors ( $B$  and  $N$ ) from  $Q_{NEIGHBORS}$
  - b. If any of  $B$  or  $N$  was visited go to step 4
  - c. Create new text-line  $TL$  and add ( $B$  and  $N$ ) neighbors
  - d. Search letters between in  $B$  to  $N$  direction (width/height are orientation dependent):
    - i. Create a moving properties window;
    - ii. Search for a letter using parallel lines;
    - iii. If a letter is found add it to  $TL$  if:
      1. It has similar size when compared to moving mean of widths/heights or if it is smaller than window mean widths/heights;
      2. The box center is between parallel lines with offset;
      3. The distance between the last letter and new one is less than the maximum distance between letters;
    - iv. Add the letter onto neighbor candidate list on  $TL$  if conditions 1-2 above are met and the third is not;
    - v. If the letter was added to  $TL$ , append it to properties window if it is not smaller than window mean widths/heights; this prevents from adding small components (e.g. accents, punctuation marks) to parallel regression;
  - e. Execute previous step for direction  $N$  to  $B$ ;
  - f. Place text-line in  $Q_{TEXTLINE}$  priority queue, with  $priority = textline\ length$
  - g. Mark all letters on the new text-line as visited;
5. Remove text-lines whereas its angle is  $90^\circ$  apart most common text-line angle;
6. While  $Q_{TEXTLINE}$  is not empty (process bigger text-lines first)
  - a. Pull-out text-line  $TL$  from  $Q_{TEXTLINE}$
  - b. If  $TL$  was merged go to step 6
  - c. For each letter on  $TL$  search for text-line upwards and add it to  $UTL$ ;
  - d. For each letter on  $TL$  search for text-line downwards and add it to  $DTL$ ;

- e. If there are any letter in common between UTL and DTL then
    - i. Mark current TL text-line as an invalid text-line;
    - ii. Delete it from text-line list;
    - iii. Go back to step 6;
  - f. Merge two text-lines in UTL if one have a candidate neighbor of the other and vice-versa;
  - g. Add new textlines to  $Q_{TEXTLINE}$  with  $priority = textline\ length$
  - h. Repeat steps 6.f-g for DTL;
7. Remove text-lines where letter count is less the moving window;
  8. Remove text-lines if it contains a letter on 10% of image border (mark them as noise);
  9. For each text-line
    - a. Calculate a simple moving linear regression [12] for top/down points
    - b. Compute corresponding point in the line and its deviation error
  10. Set top or down points as baseline whether the set with less error;

Figure 3 shows an example of the execution of step 4, where letters with boxes belong to the moving window. Figure 4 shows on upper left finding parallel lines and neighbor candidates between lines on upper right, on bottom shows merging result. Figure 5 shows the execution of step 7-8, with top and down sample points in gray and baseline points in black. The Method described herein can also be used to estimate document orientation using text-line angle histogram.

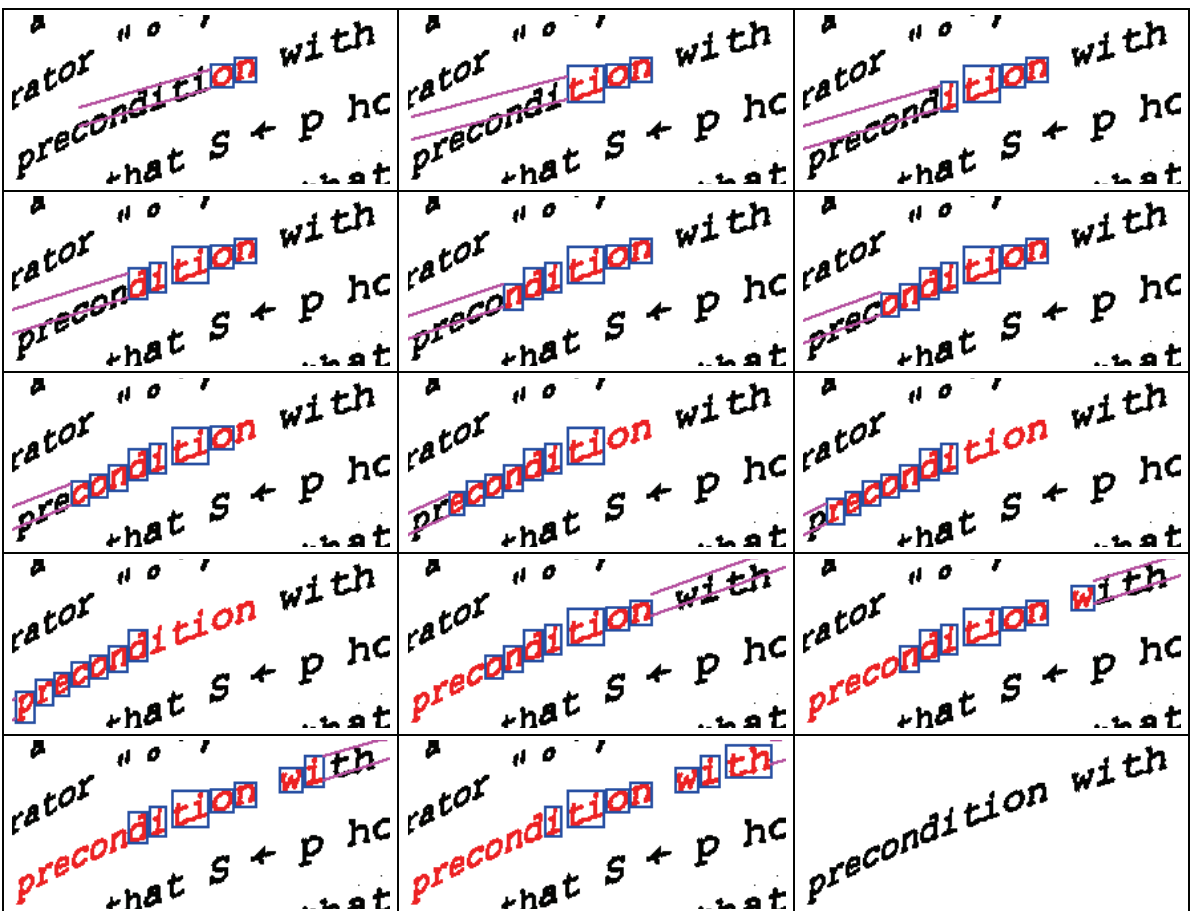


Fig. 3. Example of text-line formation by the new method

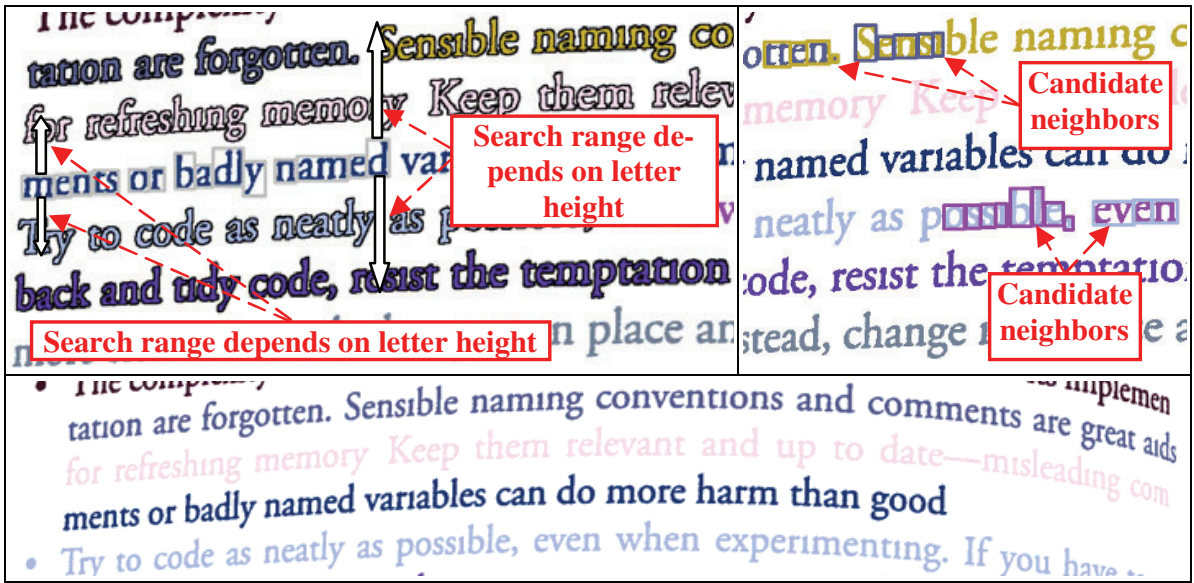


Fig. 4. Text-lines merging procedure: (upper left) parallel lines; (upper right) candidate neighbors; (bottom) merging result

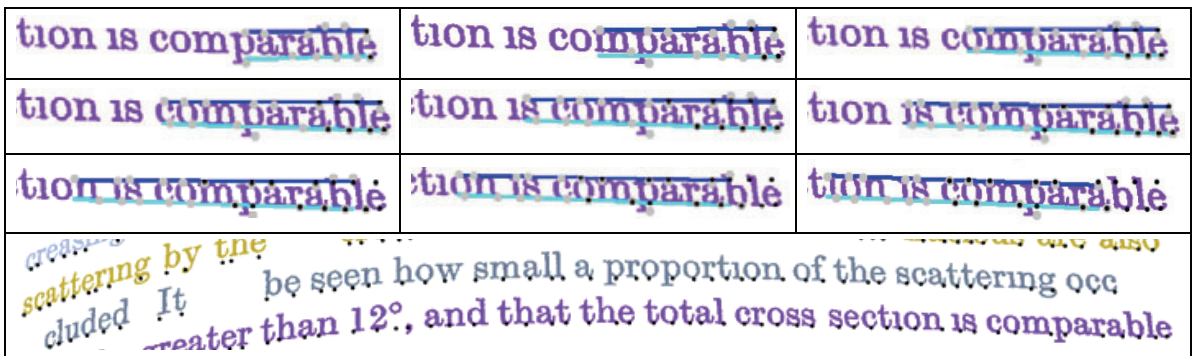


Fig. 5. Baseline estimation with simple moving simple linear regression

### 2.1.1 Parallel Line Regression

The aim of parallel lines regression is to minimize the error function of equation (2). Where  $m$  is the slope which is the same for both lines;  $b_t$  and  $b_d$  parallel lines intercept for top and bottom lines, respectively;  $(x_i, y_i)$  and  $(x_k, y_k)$  are the top and bottom samples points, respectively;  $N$  is the number of pairs of sample points.

$$E(m, b_t, b_d) = \sum_{i=1}^N (m \times x_i + b_t - y_i)^2 + \sum_{k=1}^N (m \times x_k + b_d - y_k)^2 \quad (2)$$

Making  $\partial E / \partial m = \partial E / \partial b_t = \partial E / \partial b_d = 0$ , results in eqs. (3)-(5).

$$m \left( \sum_{i=1}^N x_i^2 + \sum_{k=1}^N x_k^2 \right) + b_t \sum_{i=1}^N x_i + b_d \sum_{k=1}^N x_k = \sum_{i=1}^N y_i x_i + \sum_{k=1}^N y_k x_k \quad (3)$$

$$m \sum_{i=1}^N x_i + b_t N = \sum_{i=1}^N y_i \tag{4}$$

$$m \sum_{k=1}^N x_k + b_d N = \sum_{k=1}^N y_k \tag{5}$$

Using Cramer’s rule,  $m$ ,  $b_u$  and  $b_l$  values are obtained in eqs. (6)-(9).

$$\Delta = \begin{vmatrix} \left( \sum_{i=1}^N x_i^2 + \sum_{k=1}^N x_k^2 \right) & \sum_{i=1}^N x_i & \sum_{k=1}^N x_k \\ \sum_{i=1}^N x_i & N & 0 \\ \sum_{k=1}^N x_k & 0 & N \end{vmatrix} \tag{6}$$

$$m = \frac{\Delta m}{\Delta}; \Delta m = \begin{vmatrix} \left( \sum_{i=1}^N y_i x_i + \sum_{k=1}^N y_k x_k \right) & \sum_{i=1}^N x_i & \sum_{k=1}^N x_k \\ \sum_{i=1}^N y_i & N & 0 \\ \sum_{k=1}^N y_k & 0 & N \end{vmatrix} \tag{7}$$

$$b_t = \frac{\Delta b_t}{\Delta}; \Delta b_t = \begin{vmatrix} \left( \sum_{i=1}^N x_i^2 + \sum_{k=1}^N x_k^2 \right) & \left( \sum_{i=1}^N y_i x_i + \sum_{k=1}^N y_k x_k \right) & \sum_{k=1}^N x_k \\ \sum_{i=1}^N x_i & \sum_{i=1}^N y_i & 0 \\ \sum_{k=1}^N x_k & \sum_{k=1}^N y_k & N \end{vmatrix} \tag{8}$$

$$b_d = \frac{\Delta b_d}{\Delta}; \Delta b_d = \begin{vmatrix} \left( \sum_{i=1}^N x_i^2 + \sum_{k=1}^N x_k^2 \right) & \sum_{i=1}^N x_i & \left( \sum_{i=1}^N y_i x_i + \sum_{k=1}^N y_k x_k \right) \\ \sum_{i=1}^N x_i & N & \sum_{i=1}^N y_i \\ \sum_{k=1}^N x_k & 0 & \sum_{k=1}^N y_k \end{vmatrix} \tag{9}$$

### 3 Results

Reference [11] compares the methods presented in [7][9][10] using CBDAR de-warping dataset [5]. Herein, the same comparison methodology, described in [13], is used. The ground truth and hypothesized (processed) image have each line painted using a different color. Their similarity is compared by a pixel-correspondence graph, where each node represents a text line in both images; the edges are text-line pixels that are shared between them where the weight is the total number of pixels shared. Black and white are not text-line colors; they stand for non-textual (noise) and background pixels, respectively. An incoming edge is significant if  $w_i/P \geq T_r$  and  $w_i \geq T_a$ , where  $w_i$  is the weight of the edge;  $P$  is the total number of node pixels;  $T_r$  and  $T_a$  are the relative and absolute thresholds. The following parameters are computed (copied from [11]).

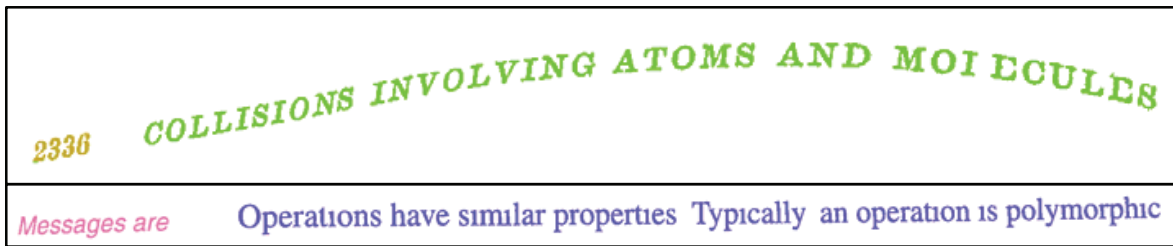
- Number of ground truth lines ( $N_g$ ) – total number ground truth lines in the whole database.
- Total correct segmentation ( $N_{o2o}$ ) – the number of one-to-one matches between the ground-truth components and the segmentation components.
- Total over segmentations ( $N_{oseg}$ ) – the number of significant edges that ground truth lines have, minus the number of ground truth lines.
- Total undersegmentations ( $N_{useg}$ ): the number of significant edges that segmented lines has minus the number of segmented lines.
- Oversegmented components ( $N_{ocomp}$ ): the number of ground truth lines having more than one significant edge.
- Undersegmented components ( $N_{ucomp}$ ): the number of segmented lines having more than one significant edge.
- Missed components ( $N_{mcomp}$ ): the number of ground truth components that matched the background in the hypothesized segmentation.
- False alarms ( $N_{falarm}$ ): the number of components in the hypothesize segmentation that did not match any foreground component in the ground-truth segmentation.
- % correct segmentation ( $P_{o2o} - N_{o2o}/N_g$ )
- % oversegmented text-lines ( $P_{ocomp} - N_{ocomp}/N_g$ )
- % undersegmented text-lines ( $P_{ucomp} - N_{ucomp}/N_g$ )
- % missed text-lines ( $P_{mcomp} - N_{mcomp}/N_g$ )

Table 2 shows results of new algorithm and [7][9][10] (copied from [11] until writing of this article), where G-ridges and B-ridges stands for [10] the segmentation in grayscale and binary images, respectively. The proposed method has the best performance for under segmentation, false positives and correct segmentation figures. No parallel line merging was registered, lines where merged if they were aligned but belong to other column such as in Figure 6. Despite the highest missed components among other algorithms, missed lines are suppressed by matched ones when it is used together with a de-warping method. An example of successful ( $P_{o2o}=100\%$ ) processing can be seen in Figure 7 with noisy pixels in black. The proposed algorithm proved also to be fast, running in 8.75s with Java implementation over Windows Vista Business on a Dell D531 3GB.

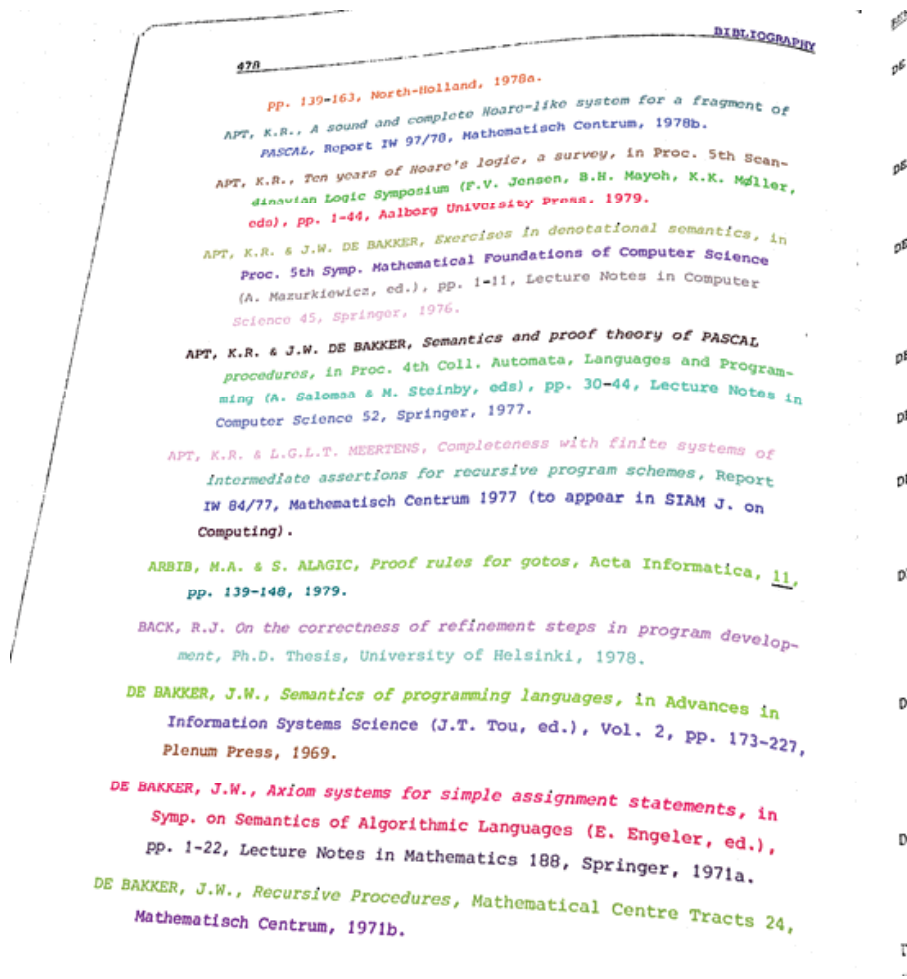


**Table 2.** Algorithms comparison metrics

Algorithm	$N_g$	$N_s$	$P_{o2o}$	$P_{ocomp}$	$P_{ucomp}$	$P_{mcomp}$	$N_{oseg}$	$N_{useg}$	$N_{falarm}$
New	3091	2924	<b>91.10%</b>	21.71%	<b>1.81%</b>	4.43%	682	<b>57</b>	<b>785</b>
B-Snakes	3091	3371	87.58%	5.79%	2.91%	0%	294	117	13199
Ridges (G)	3091	3045	89.10%	3.53%	3.85%	0.91%	115	131	1186
Ridges (B)	3091	3115	89.65%	4.40%	3.30%	0.29%	144	110	2183
C-Snakes	3091	2799	78.26%	<b>1.26%</b>	9.06%	0%	<b>39</b>	359	3251



**Fig. 6.** Line merging examples



**Fig. 7.** Example of successful processing

## 4 Conclusions

A new algorithm for text-line segmentation is presented. It outperforms the other state-of-the-art algorithms with 91.10% of accuracy and 1.81% of under segmentation rates. It can automatically detect text baselines with any orientation, proving also to be fast running in 8.75s with Java implementation for CBDAR 2007 images. The new process was used successfully in correcting the binding distortion in scanned books [4] where the document orientation is arbitrary.

## Acknowledgments

The authors like to thank Syed Bukhari, Thomas Breuel and Faisal Shafait for providing page segmentation performance evaluation program source code and for discussions on the subject.

The research reported herein was sponsored by a MCT-Brazilian Government R&D Grant and CNPq funding.

## References

- [1] Masalovitch, A., Mestetskiy, L.: Usage of continuous skeletal image representation for document images de-warping. In: Proceedings of International Workshop on Camera-Based Document Analysis and Recognition, Curitiba, pp. 45–53 (2007)
- [2] Fu, B., Wu, M., Li, R., Li, W., Xu, Z.: A model-based book de-warping method using text line detection. In: 2nd Int. Workshop on Camera-Based Document Analysis and Recognition, Curitiba, Brazil (September 2007)
- [3] Ávila, B.T., Lins, R.D.: A fast orientation and skew detection algorithm for monochromatic document images. In: Proceedings of the ACM Symposium on Document Engineering, Bristol, UK, pp. 118–126 (2005)
- [4] Lins, R.D., Oliveira, D.M., Torreão, G., Fan, J., Thielo, M.: Correcting Book Binding Distortion in Scanned Documents. In: Campilho, A., Kamel, M. (eds.) ICIAR 2010, Part II. LNCS, vol. 6112, pp. 355–365. Springer, Heidelberg (2010)
- [5] Shafait, F., Breuel, T.M.: Document Image De-warping Contest. In: 2nd Int. Workshop on Camera-Based Document Analysis and Recognition, CBDAR 2007, Brazil, September 2007, pp. 181–188 (2007)
- [6] Stamatopoulos, N., Gatos, B., Pratikakis, I., Perantonis, S.J.: A two-step de-warping of camera document images. In: Proceedings 8th IAPR Workshop on Document Analysis Systems, Nara, Japan, pp. 209–216 (2008)
- [7] Bukhari, S.S., Shafait, F., Breuel, T.M.: Coupled snakelet model for curled textline segmentation of camera-captured document images. In: Proceedings 10th International Conference on Document Analysis and Recognition, Barcelona, Spain, pp. 61–65 (2009)
- [8] Bukhari, S.S., Shafait, F., Breuel, T.M.: Ridges based curled textline region detection from grayscale camera-captured document images. In: Jiang, X., Petkov, N. (eds.) Computer Analysis of Images and Patterns. LNCS, vol. 5702, pp. 173–180. Springer, Heidelberg (2009)
- [9] Bukhari, S.S., Shafait, F., Breuel, T.M.: Segmentation of curled textlines using active contours. In: Proceedings 8th IAPR Workshop on Document Analysis Systems, Nara, Japan, pp. 270–277 (2008)

- [10] Bukhari, S.S., Shafait, F., Breuel, T.M.: Textline information extraction from grayscale camera-captured document images. In: Proc. The 13th International Conference on Image Processing, Cairo, Egypt (2009)
- [11] Bukhari, S.S.: Technical Report: Performance Evaluation and Benchmarking of Three Curled Textline Segmentation Algorithms. IUPR Technical Report, Kaiserslautern (2010)
- [12] Wolfram Research. Least Squares Fitting,  
<http://mathworld.wolfram.com/LeastSquaresFitting.html>  
(accessed January 15, 2010)
- [13] Shafait, F., Keysers, D., Breuel, T.M.: Performance evaluation and benchmarking of six page segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(6), 941–954 (2008)
- [14] Naylor, M.: Typographic line terms,  
[http://en.wikipedia.org/wiki/File:Typography\\_Line\\_Terms.svg](http://en.wikipedia.org/wiki/File:Typography_Line_Terms.svg)

D. M. Oliveira, R. D. Lins, G. Torreão, J. Fan and M. Thielo. *A New Algorithm for Segmenting Warped Text-lines in Document Images*. ACM SAC 2011, Taichung, Taiwan, 2011.

# A New Algorithm for Segmenting Warped Text-lines in Document Images

Daniel Marques Oliveira  
Rafael Dueire Lins

Gabriel Torreão  
Universidade Federal de Pernambuco  
Recife, Pernambuco, Brazil  
+55 81 2126-7794  
{daniel.moliveira, rdl}@ufpe.br

Jian Fan

HP Labs  
Palo Alto, USA  
+1 650-857-2554  
jian.fan@hp.com

Marcelo Thielo

HP R&D Brazil  
Porto Alegre, Brazil  
+55 51 2121-3583  
marcelo.resende.thielo@hp.com

## ABSTRACT

The digitalization of bound documents either using flatbed scanners or digital cameras often yield images with non-straight text-lines due to a geometrical warp. This paper presents a new algorithm for text-line segmentation for documents captured by digital cameras or scanners. The proposed method reached 97.84% correct segmentation, while the best results offered by its predecessors in the literature yields 95.21%.

## Categories and Subject Descriptors

I.4.9 [Image Processing and Computer Vision]: Applications.

## General Terms

Algorithms, Segmentation.

## Keywords

Text-line segmentation; image processing; documents.

## 1. Introduction

The digitalization of hard bound documents often exhibits a geometrical distortion in the region close to the book spine, whenever performed by flatbed scanners or digital cameras. Such warp degrades document readability, both by humans and OCRs. Thus, text-line de-warping is an important step in document processing. Envelope segmentation is one of the pre-processing steps for many text-line de-warping algorithms. The segmentation process can be accompanied of a base and/or mean line estimation, Figure 1 shows some typographic elements.



Figure 1. Font typographic lines following Naylor [19]

The concept of coupled snakes [20][21] was used by Bukhari, Shafait, and Breuel [12] to segment text-lines. In references

[13][15] they obtain text-lines by ridge detection in grayscale images with baseline estimation using coupled snakes.

Stamatopoulos *et al* [11] approximate the border lines by a polynomial of degree three to generate a 3D-model, while Fu and his colleagues [2] estimate border points fitting them into a 3D-cylinder model. Masalovitch and Mestetskiy [1] use Voronoi diagrams between letter cases to estimate the spaces between lines; a Bézier path is built by the de-warping procedure.

The new algorithm presented herein is an extension of the method in reference [9], which introduces a new text-line segmentation method, used in reference [8] for scanned document de-warping. The algorithm is based on ideas from skew detectors [6][7] and coupled snakes introduced by Bukhari, Shafait, and Breuel [12]; the new method can be used regardless of document orientation. The recent paper in reference [16] presents a performance comparison of the best text-line segmentation methods in the literature pointing out 95.21% as the best result achieved. The performance of the algorithm presented herein yields 97.84% correct segmentation benchmarked on the CBDAR 2007 test-set [10].

## 2. Background

The proposed algorithm for text-line segmentation was inspired by the skew detection scheme [7] as it takes a monochromatic image as input and performs a component labeling of all black blocks in the image. Blocks are merged in a bottom-up procedure depending on their relative distances forming lines. The skew and orientation detection algorithm moves towards analyzing the features of the envelopes of such lines and the up and downwards strokes to detect document orientation and study all the envelope lines to assert the document skew.

The concept of coupled snakes was introduced by [20][21] and was used for medical imaging. Bukhari and his colleagues [12] take feature points of each character to segment textlines. They assume that text lines are horizontal and upwards. Figure 2 shows an example by starting from a single letter and expanding to nearby letters.

The new textline segmentation algorithm is also based on O’Gorman Docstrum algorithm [6], which is a skew detection algorithm based on k-nearest neighbor of all connected components in a document. Figure 3 shows the k-nearest neighbor, which may be computed with loglinear time complexity [4].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC’11, March 21-25, 2011, TaiChung, Taiwan.

Copyright 2011 ACM 978-1-4503-0113-8/11/03...\$10.00.

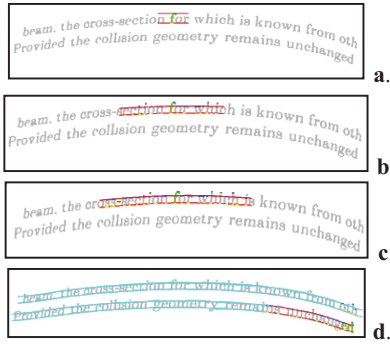


Figure 2. Coupled snakes line expansion [12].



Figure 3. O'Gorman Docstrum [6]

One important issue is finding the typesetting parameters of the text. Reference [9] claims that the height of fonts is less sensitive to rotation than the width. To prove that, the list of characters was analyzed in font faces: Arial, Courier and Times.

0/!\{\}~?0123456789ABCDEFGHIJKLMNPOQRSTUVWXYZ  
XYZ abcdefghijklmnopqrstuvwxyz

Each one was painted on a blank image with 70pt using ImageJ [22] mixing the following characteristics: with/without italic; with/without bold; rotated by 15° and 30°, and without rotation.

Figure 4 shows some example of these characters. In the case where the letter has more than one connected component (e.g. lowercase “i”), the largest component was considered.



Figure 4. “B”: Courier (a); Arial Bold Italic (b); Times 30° (c)

Table 1 presents some statistics of the connected component dimensions of the characters for plain style. Where  $R_w$  and  $R_h$  are as in equations (1) and (2), respectively, if values are close to extreme dimension are more similar;  $\bar{w}$  and  $\bar{h}$  are mean of the width and the height, respectively;  $\sigma_w$  and  $\sigma_h$  are width and height standard deviation percentages relative to  $\bar{w}$  and  $\bar{h}$ , respectively.

$$R_w = \frac{\min width}{\max width} \quad (1) \quad R_h = \frac{\min height}{\max height} \quad (2)$$

Table 1. Analysis of the variation of font width and height

Font	$\theta$	$R_w$	$R_h$	$\bar{w}$	$\bar{h}$	$\sigma_w$	$\sigma_h$
Arial	0°	0.073	0.554	32.3	48.5	37.6%	14.6%
Arial	15°	0.217	0.576	39.5	52.4	26.9%	13.7%
Arial	30°	0.324	0.486	45.4	52.4	21.6%	16.0%
Courier	0°	0.071	0.528	30.7	40.3	25.7%	15.3%
Courier	15°	0.240	0.547	35.9	44.7	20.3%	12.5%
Courier	30°	0.339	0.491	41.1	45.6	17.5%	13.3%
Times	0°	<b>0.047</b>	0.508	33.0	45.9	37.8%	17.1%
Times	15°	0.192	0.522	39.1	49.9	29.4%	16.0%
Times	30°	0.269	<b>0.472</b>	44.8	50.0	23.7%	18.8%

Looking at Table 1 one may observe that the height varies less than the width as it presents a smaller standard deviation. One may observe that  $R_h$  can be about 10 times greater than  $R_w$ . This fact is also found in italic and bold styles.

### 3. The New Segmentation Algorithm

The new textline segmentation algorithm takes as input a monochromatic image and performs the following steps:

1. Component Labeling;
2. Add components to multidimensional tree MDTREE;
3. Perform all k-nearest neighbors on MDTREE;
4. Create orientation neighbor priority queue;
5. For each neighbor on priority queue, group a text-line starting from them;
6. Identify text columns;
7. Merge text-lines that are in same text columns.

A multidimensional tree is used to perform k-nearest neighbors and range search efficiently. KD-tree and UB-trees are able to provide that [3][4][5]. The computation of all k-nearest neighbors for document analysis was introduced by Docstrum [6].

Reference [7] groups characters with similar properties walking through the document to find the upper and lower envelopes of text-lines. It uses independent linear regression for the detection, which may not work well when the starting point is a character whose case is different from the majority of those in the line, as may be observed in the example presented in Figure 5.a for finding envelopes of word “None”.



Figure 5. Neighbor issues: Different case sizes (a); ascend/descending stroke regression (b)

Such scenario is avoided by creating two priority queues: one for horizontal and other for vertical orientations. A component and its nearest neighbor will be added into a queue if their dimension are compatible and they are close enough (see Listing 1 for a precise definition) with priority is defined by equation 3, where the ratio function is defined by equation 4. Assuming two neighboring blocks B and N, the priority is higher as:

- blocks dimensions of B and N are closer (inter block dimension similarity);
- and their dimension are similar (“squareness” of each block).

$$\text{priority}(B, N) = \text{ratio}(W_B, W_N) + \text{ratio}(H_B, H_N) + \frac{\text{ratio}(W_B, H_B) + \text{ratio}(W_N, H_N)}{2} \quad (3)$$

$$\text{ratio}(a, b) = \min(a, b) / \max(a, b) \quad (4)$$

The approach followed in reference [9], considers only the inter-block dimension similarity. However, there are some cases where the dimensions are close but the regression path does not replicate the mean and base lines (e.g. ascending followed by a descending stroke sequence as in Figure 5.b). Thus, it is better to have as an initial block symbols without ascending/descending strokes, such as letters “a”, “c”, etc). In this case, the aspect ratio of the letter case should have values close to 1:1, i.e. a square.

No pre-assumption is made regarding document orientation. The creation of horizontal and vertical priority queues helps to give better starting blocks for text-line formation regardless document orientation. As the text-line is assembled, blocks with the highest and lowest coordinates are recorded. Text-line orientation is obtained by listings 2 and 3. UV coordinates are defined according to them; if the orientation is horizontal the (u,v) values are the same as (x,y); if it is vertical (u,v) values are switched to (y,x). Figure 6 illustrates (u,v) points of a letter. Figure 7 shows one example of extreme u-coordinate points and the vector obtained by it in polar coordinates.

**Listing 1**

```

Input: Blocks / Output: if they are compatible
FUNCTION IS_COMPATIBLE(Ci,Cj)

Let orientation be defined by
  GET_ORIENTATION(|Ci.x-Cj.x|,|Ci.y-Cj.y|);
Let maxdim =
  max(Ci.width,Ci.height,Cj.width,Cj.height);
RETURN
  ratio(Ci.uvwidth,Cj.uvwidth) ≥ MIN_RATIO and
  ratio(Ci.uvheight,Cj.uvheight) ≥ MIN_RATIO and
  Distance between rectangles defined by Ci and
  Cj is less than SFACORXmaxdim;

END FUNCTION

```

**Listing 2**

```

Input: Abs. difference between coordinates
Output: Text-line orientation
FUNCTION GET_ORIENTATION(Δx,Δy)
  IF Δx > Δy THEN
    RETURN HORIZONTAL;
  ELSE
    RETURN VERTICAL;
  END IF
END FUNCTION

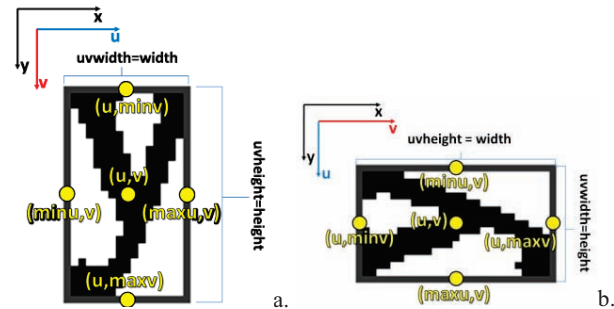
```

**Listing 3**

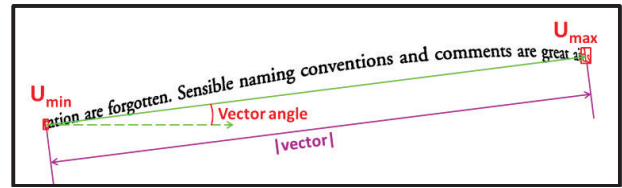
```

Input: T - text-line / Output: Line orientation
FUNCTION GET_TL_ORIENTATION(T)
//T.minX is the block with lowest x value
//T.minY is the block with lowest y value
Δx = max(|T.minY.x - T.maxY.x|,
  |T.minX.x - T.maxX.x|);
Δy = max(|T.minY.y - T.maxY.y|,
  |T.minX.y - T.maxX.y|);
RETURN GET_ORIENTATION(Δx,Δy);
END FUNCTION

```



**Figure 6.** (u,v) coordinates for orientation = horizontal (a); (u,v) coordinates for orientation = vertical (b);



**Figure 7.** Text-line properties

As the queue is processed, the text-lines are formed by parallel line regression using (u,v) coordinates. Text-line may be curled, but looking into a small vicinity of the characters the mean and base lines are approximately linear. The text-line is expanded by doing a regression with the  $U_{min}$  and  $U_{max}$  neighborhoods, alternately, as presented in Listing 4. In both expansions, a block is added when a k-neighbor of  $U_{min}$  or  $U_{max}$  with its center (u,v):

- is between the parallel lines defined by the corresponding vicinity;
- block dimensions and typographic distance are below a threshold using x-height as basis.

Listing 5 provides the pseudo code with previous restrictions. Figures 8 and 9 show examples for adding blocks in  $U_{min}$  and  $U_{max}$  directions, respectively. Figure 10 and 11 illustrate how to compute the typographic distance and x-height, respectively.

**Listing 4**

```

Input: Neighbors / Output: Textline
PROCEDURE EXTRACT_TEXTLINE(Ci,Cj)
T = {Ci, Cj};
WHILE a new letter is found for T DO
  FIND_LETTER(T, Umin); FIND_LETTER(T, Umax);
END WHILE
RETURN T;
END PROCEDURE

```

**Listing 5**

```

Input: Extreme U letter / Output: T updated
PROCEDURE FIND_LETTER(T, Ck)

```

```

PR = parallel regression around Ck
xheight = PR.cmax - PR.cmin;

```

```

IF there exists a neighbor block Nj of Ck such
  Nj.uvwidth ≤ xheight*MAX_W and
  MIN_H*xheight ≤ Nj.uvheight ≤ xheight*MAX_H and
  mdist(Nj.u - Ck.u, PR.m) ≤ xheight*MAX_MDIST
  and (PR.m×Nj.u + PR.cmin - PR.OFFSET) ≤ Nj.v
  and Nj.v ≤ (PR.m×Nj.u + PR.cmax + PR.OFFSET)
THEN
  T = T + {Ck};
END IF
END PROCEDURE

```

The text-line formation process may be split due to large inter-words distance around punctuation marks and in justified texts. In these cases, the k-nearest neighbor may not be able to connect those characters. A text column is identified by looking for parallel lines up and downwards in the v direction using the k-nearest neighbor that does not belong to the text-line. Listing 6 presents the pseudo code for that. A text column is transitively formed: two text-lines belong to the same column if they are parallel to each other; Listing 7 presents the pseudo code for that. Figure 12 exemplifies the preliminary segmentation followed by column extraction using this idea.

**Listing 6**

```

PROCEDURE EXTRACT_PARALLELS(T)

//T.minu and T.maxu are extreme u blocks
minu = T.min.u;
maxu = T.max.u;

FOR EACH Lt in T.letters DO
  FOR EACH Ln in Lt.kneighbors
    IF Ln belongs to a text-line
      and minu ≤ Ln.u ≤ maxu THEN
      T.parallels = T.parallels +
        {Ln text-line};
    END IF
  END FOR
END FOR
END PROCEDURE

```

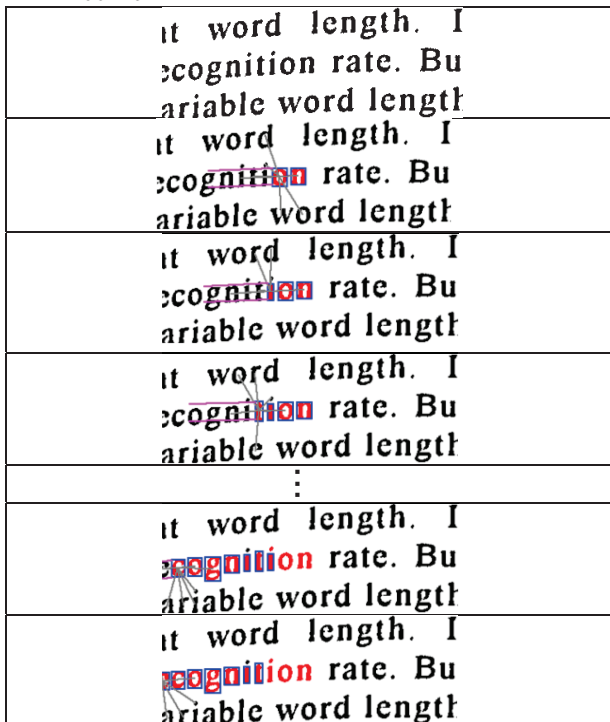


Figure 8. Finding new blocks in  $U_{\min}$  direction

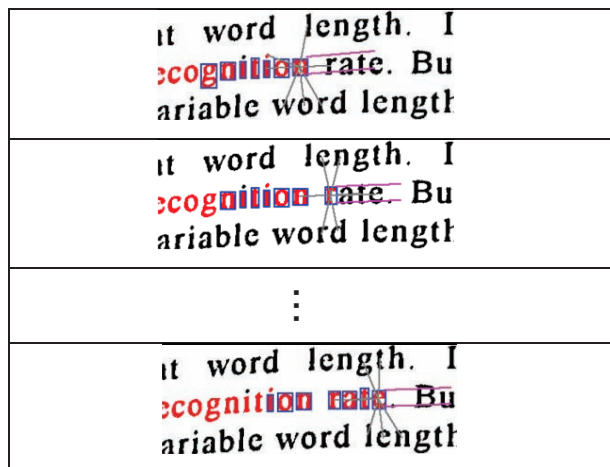


Figure 9. Finding new blocks on  $U_{\max}$  direction

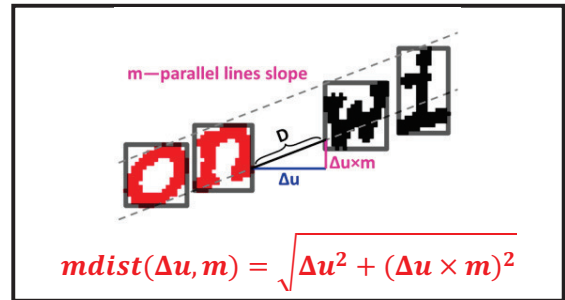


Figure 10. Typographic distance definition



Figure 11. X-height calculation

**Listing 7**

```

PROCEDURE EXTRACT_COLUMN(T)

COLUMN.textlines = empty set;
TLQUEUE = {T};

WHILE TLQUEUE is not empty DO
  Pop Ti from TLQUEUE;
  COLUMN.textlines = COLUMN.textlines + {Ti};
  FOR EACH Tj in Ti.parallels DO
    IF Ti ∈ Tj.parallels and
      Tj ∉ COLUMN.textlines THEN
      TLQUEUE = TLQUEUE + {Ti};
    END IF
  END FOR
END WHILE

END PROCEDURE

```

For every text-line, a range search is done on the MDTREE for blocks that are between parallel lines in the vicinities of  $U_{\min}$  extremity and of  $U_{\max}$ . Among these if there is a block that:

- belongs to another text-line;
- is in same text column;
- is not parallel to the current text-line.

Textline merging is repeated until no merging is possible; Listings 8 and 9 presents the pseudo code of this process. Figure 13 illustrates this process. One may observe that the range search was rectified, but blocks outside parallel lines are not processed.

**Listing 8**

```

PROCEDURE MERGE_TEXTLINES(C)

FOR EACH Ti in C.textlines DO

  WHILE merging is possible DO

    //Search towards minu block
    Create parallel regression PR with blocks in
      Ti.blocks with W_SIZE-lowest u coords;
    searchRange = PR.xheight*MDIST_MERGE;

    searchResult =
      Search on MDTREE blocks inside PR with
        u=[Ti.minu-searchRange,Ti.minu];

    MERGE_NEIGHBOR_TL(Ti,searchResult);
  END WHILE
END FOR

```



```
//Search towards maxu block
Create parallel regression PR with blocks in
Ti.blocks with W_SIZE-highest u coords;
searchRange = PR.xheight*MDIST_MERGE;

searchResult =
Search on MDTREE blocks inside PR with
u=[Ti.maxu,Ti.maxu+searchRange];

MERGE_NEIGHBOR_TL(Ti,searchResult);

END WHILE
END FOR
END PROCEDURE
```

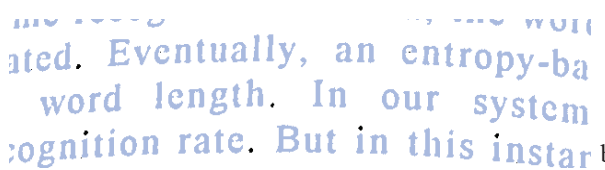


Figure 12. Text column extraction: preliminary segmentation with neighbors (a); extracted block (b)

```
Listing 9
PROCEDURE MERGE_NEIGHBOR_TL(Ti,blocks)

FOR each block Bj in blocks DO
IF Bj belongs to a text-line Tj and
Ti ∉ Tj.parallels and Tj ∉ Ti.parallels THEN
Merge Ti and Tj;
exit procedure;
END IF
END FOR
END PROCEDURE
```

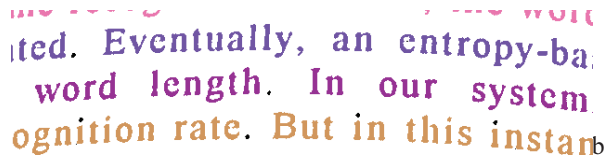
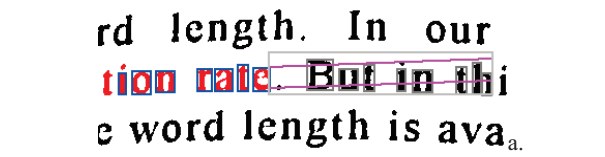


Figure 13. Merging: searched blocks (a); text-lines merged (b)

For a better understanding of how the algorithm works, Listing 10 presents the pseudo code of it. For the sake of space, the algorithms parameters are explained in another publication. Table 2 shows the recommended values; table 3 shows the derived parameters.

Table 2. Algorithms recommended parameters

SFACTOR	0.52	K_NN	8	MAX_W	9.0
MAX_H	2.63	MIN_H	0.70	W_SIZE	7
MAX_MDIST	1.97	MDIST_MERGE	5.26	PR_OFFSET	0.2899

Table 3. Algorithms derived parameters

MIN_RATIO_W	1/MAX_W	MIN_RATIO_H	max(MIN_H, 1/MAX_H)
-------------	---------	-------------	---------------------

Listing 10

```
Input: Binary image
Output: Text columns
PROCEDURE IDENTIFY_TEXTLINES()

Extract connect components of binary image
Create multidimensional tree MDTREE with
connected components
Create K-NNG of MDTREE with k set to K_NN
Create orientation priority queue NQUEUE

FOR EACH connected comp. Ci on MDTREE DO

Let Cj be the nearest neighbor of Ci;

IF IS_COMPATIBLE(Ci, Cj) THEN
//NQUEUE priority defined by eq. (3)
Add (Cj, Ci) to NQUEUE;
END IF

END FOR

CREATE text-lines set TLS

WHILE NQUEUE is not empty DO
Pop (Ci, Cj) elements on top of NQUEUE
IF Ci and Cj don't belong to a text-line THEN
T = EXTRACT_TEXTLINE(Ci,Cj);
TLS = TLS + {T};
END IF
WHILE

FOR EACH text-line T on textLines DO
EXTRACT_PARALLELS(T);
END FOR

CREATE columns set COLS

FOR EACH text-line T on textLines DO
IF T does not belong to a column THEN
C = EXTRACT_COLUMN(T);
COLS = COLS + {C};
MERGE_TEXTLINES(C);
END IF
END FOR

END PROCEDURE
```

#### 4. Results

Reference [16][17] compares the methods presented in [12][14][15] and another unpublished algorithm by the same authors using CBDAR de-warping dataset [10]. The same comparison methodology described in reference [18] is used here. The text-line information is stored in a color image, in which each line is painted with a different color. Ground truth and processed image are compared by a pixel-correspondence graph, where each node represents a text-line in both images; an edge between two lines exists if they share a pixel with weight set to the total number of pixels shared. Black and white are not used for text-line colors; they stand for noise and background pixels, respectively. An incoming edge is significant if  $w_i/P \geq Tr$  and  $w_i \geq Ta$ , where  $w_i$  is the weight of the edge;  $P$  is the total number of node pixels;  $Tr$  and  $Ta$  are the relative and absolute

thresholds. The following parameters are computed (following [16]). Figure 14 exemplifies text-line segmentation errors.

- Number of ground truth lines ( $N_g$ ) – total number ground truth lines in the whole database.
- Number of ground truth lines ( $N_s$ ) – total number segmented components that are not false positive;
- Total correct segmentation ( $N_{o2o}$ ) – the number of one-to-one matches between the ground-truth components and the segmentation components.
- Total over segmentations ( $N_{oseg}$ ) – the number of significant edges that ground truth lines have, minus the number of ground truth lines.
- Total under segmentations ( $N_{useg}$ ): the number of significant edges that segmented lines has minus the number of segmented lines.
- Over segmented components ( $N_{oc}$ ): the number of ground truth lines having more than one significant edge.
- Under segmented components ( $N_{uc}$ ): the number of segmented lines having more than one significant edge.
- Missed components ( $N_{mc}$ ): the number of ground truth components that matched the background in the hypothesized segmentation.
- False alarms ( $N_{falarm}$ ): the number of components in the hypothesize segmentation that did not match any foreground component in the ground-truth segmentation.
- % correct segmentation ( $P_{o2o}$ ) –  $N_{o2o}/N_g$
- % oversegmented text-lines ( $P_{oc}$ ) –  $N_{oc}/N_g$
- % undersegmented text-lines ( $P_{uc}$ ) –  $N_{uc}/N_g$
- % missed text-lines ( $P_{mc}$ ) –  $N_{mc}/N_g$

**Correct segmentation without noise...**

**Correct segmentation with wrong noise...**

**Over segmentation**

**Under segmentation**

**in different lines**

**123 Under segmentation in the same "line"**

**missed lines happens when it is marked as noise**

Figure 14. Illustration of segmentation metrics with noise in black pixels and correctly detected information in color

Table 4 shows the results of the new algorithm and other published in [17], where *prev.* stands for the result of algorithm [9]; R(G) and R(B) for [15] the segmentation in grayscale and binary images, respectively. Is worth remarking that the first author of this paper detected that 0.5% of the ground truth text-lines were incorrectly annotated by the benchmark developers [17]. Therefore,  $P_{o2o}$ ,  $P_{oc}$  and  $P_{uc}$  percentages of other algorithms may oscillate in about 0.5%.

The proposed method has the best performance for under segmentation and correct segmentation figures, with no parallel line merging registered. Despite the highest missed components among other algorithms, these lines were characterized by small lines as presented on Figure 15, where is presented lines with single blocks and mixture of punctuation and letters.

The number of false alarm increased compared to previous algorithm, on the other hand previous has a higher rate for missed components. The main cause is due the rejection of lines with few blocks by the previous approach while the new does not behave that. Figure 16 presents an example of false alarm of

new approach, where one may observe that this wrong segmentation is not that bad.

Another advance comparing the previous and new regards the decrease of over segmentation components ( $P_{oc}$  and  $N_{oseg}$ ). It was obtained due the inclusion of punctuation marks by the new algorithms.

An example of successful ( $P_{o2o}=100\%$ ) processing can be seen in Figure 17 with noisy pixels in black. The proposed algorithm proved also to be fast running in 8.75s (the references on all the other methods do not provide execution times) with Java implementation over Windows Vista Business on a Dell D531 3GB.

Table 4. Algorithms comparison metrics

Algo.	$N_g$	$N_s$	$P_{o2o}$	$P_{oc}$	$P_{uc}$	$P_{mc}$	$N_{oseg}$	$N_{useg}$	$N_{falarm}$
New	3103	3098	<b>97.8%</b>	1.6%	<b>0.3%</b>	0.8%	53	<b>9</b>	848
Prev.	3091	2924	91.1%	22%	1.8%	4.4%	682	57	<b>785</b>
B-S	3091	3371	87.6%	5.8%	2.9%	<b>0%</b>	294	117	13199
R (G)	3091	3045	89.1%	3.5%	3.8%	0.9%	115	131	1186
R (B)	3091	3115	89.6%	4.4%	3.3%	0.3%	144	110	2183
C-S	3091	2799	78.3%	<b>1.3%</b>	9.1%	<b>0%</b>	<b>39</b>	359	3251
C-S v2	3091	3093	95.2%	1.7%	1.6%	<b>0%</b>	54	51	3277



Figure 15. Line missing examples: lines with punctuation marks and a single character (a); undersegmented binarization (b); page number with a single digit (c)

Figure 16. False alarm sample

## 5. Conclusions

A new algorithm for text-line segmentation was presented. It outperforms the other state-of-the-art algorithms with 97.84% of accuracy. It can automatically detect text baselines of documents with any orientation. The time efficiency of the algorithm is high as its implementation in Java takes in average 8.75s for CBDAR 2007 images [10]. The new process was used successfully in correcting the binding distortion in scanned books [8] where the document orientation is arbitrary.

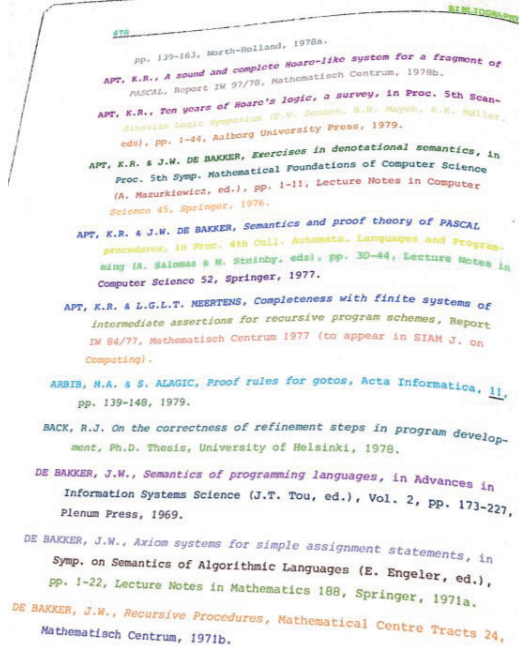


Figure 17. Example of successful processing

## 6. References

- [1] A. Masalovitch and L. Mestetskiy. Usage of continuous skeletal image representation for document images de-warping. Proceedings of International Workshop on Camera-Based Document Analysis and Recognition, Curitiba, 2007, pp. 45-53.
- [2] B. Fu, M. Wu, R. Li, W. Li, and Z. Xu. A model-based book de-warping method using text line detection. In 2nd Int. Workshop on Camera-Based Document Analysis and Recognition, Curitiba, Brazil, Sep. 2007.
- [3] De Berg, M. et al. Computational Geometry: Algorithms and Applications, 3rd Edition, pages 99-101. Springer, 2008.
- [4] Connor, M., Kumar, P.: Fast construction of k-nearest neighbor graphs for point clouds. IEEE Transactions on Visualization and Computer Graphics 99 (PrePrints) (2010).
- [5] F.Ramsak. Towards a general-purpose, multidimensional index: Integration, Optimization, and Enhancement of UB-Trees. Munich University of Technology. PhD Thesis. Munich, 2002.
- [6] L.O’Gorman. The Document Spectrum for Page Layout Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 15(11), p. 1162-1173, 1993;
- [7] B. T. Ávila and R. D. Lins. A fast orientation and skew detection algorithm for monochromatic document images. Proceedings of the ACM Symposium on Document Engineering, pp. 118-126, 2005.
- [8] R.D.Lins, D.M.Oliveira, G. Torreão, J. Fan and M. Thielo. Correcting Book Binding Distortion in Scanned Documents. Proceedings of ICIAR 2010. LNCS 6112, pp. 355-365, Springer Verlag, 2010.
- [9] D.M.Oliveira, R.D.Lins, G. Torreão, J. Fan and M. Thielo. A New Method for Text-line Segmentation for Warped Documents. Proceedings of ICIAR 2010. LNCS 6112, pp. 398-408, Springer Verlag, 2010.
- [10] F. Shafait and T. M. Breuel. “Document Image De-warping Contest”, 2nd Int. Workshop on Camera-Based Document Analysis and Recognition, pp 181-188, Brazil. Sep. 2007.
- [11] N.Stamatopoulos, B. Gatos, I. Pratikakis, S.J. Perantonis. A two-step de-warping of camera document images. Proceedings 8th IAPR Workshop on Document Analysis Systems, pp. 209–216, Nara, Japan, 2008.
- [12] S. S. Bukhari, F. Shafait, and T. M. Breuel. Coupled snakelet model for curled textline segmentation of camera-captured document images. In Proceedings 10th International Conference on Document Analysis and Recognition, pp. 61–65, Barcelona, Spain, 2009.
- [13] S. S. Bukhari, F. Shafait, and T. M. Breuel. Ridges based curled textline region detection from grayscale camera-captured document images. In Proc. The 13th International Conference on Computer Analysis of Images and Patterns, LNCS 5702, pp. 173–180, Springer Verlag, 2009.
- [14] S. S. Bukhari, F. Shafait, and T. M. Breuel. Segmentation of curled textlines using active contours. In Proceedings 8th IAPR Workshop on Document Analysis Systems, pp. 270–277, Nara, Japan, 2008.
- [15] S. S. Bukhari, F. Shafait, and T. M. Breuel. Textline information extraction from grayscale camera-captured document images. In Proc. The 13th International Conference on Image Processing, 2009.
- [16] S. S. Bukhari. Technical Report: Performance Evaluation and Benchmarking of Three Curled Textline Segmentation Algorithms. IUPR Technical Report, Kaiserslautern, 2010.
- [17] S.S. Bukhari, F. Shafait, T. M. Breuel. “Performance Evaluation of Curled Textlines Segmentation Algorithms”, 9th IAPR Workshop on Document Analysis Systems, 2010.
- [18] F. Shafait, D. Keysers, and T. M. Breuel. Performance evaluation and benchmarking of six page segmentation algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(6):941–954, Jun 2008.
- [19] M. Naylor. Typographic line terms. Available at [http://en.wikipedia.org/wiki/File:Typography\\_Line\\_Terms.svg](http://en.wikipedia.org/wiki/File:Typography_Line_Terms.svg).
- [20] B. Hohnhaeuser and G. Hommel. 3D pose estimation using coupled snakes. J. WSCG, 12(1-3):1213–6972, Feb 2003.
- [21] R. Chandrashekar, R. H. mohiaddin, and D. Rueckert. Analysis of myocardial motion in tagged MR images using nonrigid image registration. Progress in biomedical optics and imaging, 3(2):1168–1179, 2002.
- [22] ImageJ. <http://rsb.info.nih.gov/ij/>.

D. M. Oliveira e R. D. Lins. *An Efficient Algorithm for Segmenting Warped Text-lines in Document Images*. (Submetido em 6 de Julho de 2011 ao IJDAR).

# An Efficient Algorithm for Segmenting Warped Text-lines in Document Images

Daniel Marques Oliveira · Rafael Dueire Lins · Gabriel Torreão · Jian Fan · Marcelo Thielo

**Abstract** Warped text-lines often appear whenever one performs the digitalization of bound documents either using flatbed scanners or digital cameras. Compensating such distortion is an important pre-processing step in document transcription via OCR, for instance. This paper presents an efficient algorithm for text-line segmentation for document images. A typographic study and parameter tuning are done yielding into high values for precision, recall and f-measure metrics. The method presented outperforms the competing algorithms using a public available dataset.

## Keywords

*Text-line segmentation, page segmentation, document image processing, camera documents.*

## 1 Background

The digitalization of hard bound documents whenever performed by flatbed scanners or digital cameras often exhibits a warp in the region close to the book spine. If the device is not positioned properly, perspective distortion may be perceivable with high variability of pixel density (i.e. DPI) in the resulting image. All these problems degrade OCR performance and turn reading more difficult by humans. Several de-warping algorithms were proposed to overcome those problems. Most of them are depend on efficient text-line segmentation.

Stamatopoulos and his colleagues [24] fitted points of textual margin into a polynomial of degree three, a three dimensional model is generated with this estimation. Fu *et al* [12] proceed similarly using a 3D cylinder. Masalovitch and Mestetskiy [16] used Voronoi diagrams of characters to form the text-line. Ulges and his colleagues [25] extract text-lines by an analysis of minimum distance and overlap between bounding box of connected components.

Bukhari *et al* [6] executes morphological dilation which serves to initialize the “baby snakes”. Gradient Vector Flow distorts the snakes; these are merged together after few deformation steps. In references [5][7] they obtain text-lines by anisotropic ridges

detection in grayscale images. In reference [4] they identify text-lines with coupled snakes (a medical imaging segmentation method [10][13]). It starts from a single character and expands the text-line using top and bottom snakes that are coupled together. Different weights are set for these snakes making the algorithm orientation dependent. Figure 1 shows an example by starting from a single character and expanding to nearby symbols.

The proposed algorithm works with a multi-dimensional data-structure that performs k-nearest neighbors search. O’Gorman [17] was the first to use this operation for skew detection. Figure 2 shows one example of O’Gorman Docstrum approach.

Instead of starting from a single character, the new approach extracts the text-line beginning with two symbols and spans until blocks cannot be added. A parallel linear regression with upper and lower points of each symbol is estimated by coupling errors of both point sets. The result of this process are two parallel lines with lower computational cost than in [4].

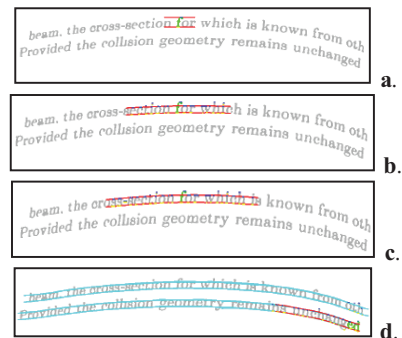


Fig. 1 Coupled snakes line expansion [4].



Fig. 2 O’Gorman Docstrum [17].

The method presented here details and optimizes the algorithm presented in reference [19], in the following ways:

- It introduces a study of the some typographic guidelines on distance between letters, lines and columns used by professional designers.
- The statistical typographical analysis performed is used to set the limits for each parameter value; most of the other methods in the literature are not explicit about parameter limits and how they can be tuned.
- It tests the proposed algorithm for different parameter setups, showing how each of them affects the overall outcome.
- It presents precision and f-measure data for text-line segmentation. Such data are important as they cover false positives text-lines which are not mentioned by competing algorithms.

Daniel Marques Oliveira · Rafael Dueire Lins · Gabriel Torreão  
*Universidade Federal de Pernambuco, Recife, Brazil*  
+55 81 2126-7794  
e-mail {daniel.moliveira, rdl, gabriel.dsilva}@ufpe.br

Jian Fan  
*HP Labs*  
*Palo Alto, USA*  
+1 650-857-2554  
e-mail jian.fan@hp.com

Marcelo Thielo  
*HP R&D Brazil*  
*Porto Alegre, Brazil*  
+55 51 2121-3583  
marcelo.resende.thielo@hp.com

The algorithm described here is used by [14] to dewarp scanned documents. One of its advantages follows:

- It is time-efficient, taking in average 4 seconds of processing time (excluding image loading and saving). Most of the other algorithms in the literature do not provide this information and seem to be far more time-consuming due to their complexity;
- Processing does not access the image once all the needed information is added into a multidimensional data structure;
- Vertical and horizontal lines are segmented in a similar way without distinction;
- The algorithm has freedom of orientation.

The structure of this paper is now presented. Section 2 addresses the statistical analysis of typographical parameters.. Section 3 shows how parallel regression is computed. Section 4 presents the new algorithm. Section 5 exploits the influence of some parameters and compares the performance of the proposed algorithm to other of the best similar purpose algorithms.

## 2 Typographic analysis

Good typographic guidelines were developed over centuries and are partially described in references [3] and [11], for instance. Some of these tips range from character design to spacing between letters and text-lines. Most of times, symbols are structured as in Figure 3. One important measure is the x-height which is defined by the height of lowercase letters that are neither ascenders nor descenders.



**Fig. 3** Font typographic lines following Naylor [27].

Spacing between letters, words and columns are relative to the 1em value [3] [11]. This unit is set according to font size (e.g. 1em is equal to 15pt for a font with size of 15pt). The following section compares the x-height of popular fonts and links it to the textual spacing [3] [11].

### 4.1 Linking the x-height to typographic guidelines

Typographic lines were measured here by placing the different character types in Microsoft Word 2007 with 80pt font size and applying a gray highlight on them. The height of the highlight is considered as 1em.

Figure 4 shows the measure of the typographic lines. The upper offset (U) is the distance between median line and 1em upper limit. The lower offset (L) is defined by the distance between the baseline and 1em lower limit. The x-height (X) is the distance between median line and baseline. EM stands for 1em value.



**Fig. 4** Typographic measures.

The subscript of U, X, L and EM rely upon which is the reference measure. For instance, value for  $U_X$  stands for the upper offset value relative to the x-height, while  $U_{EM}$  it is compared to 1em. Thus, the variable relative to itself will always equal 1 (e.g.  $EM_{EM} = 1$ ). Table 1 shows U, X and L values relative 1em of popular fonts with the maximum and minimum in the last two rows. Table 2 presents the data for the same fonts relative the x-height.

**Table 1** Typographic measure of some fonts relative to 1em.

Font face	$X_{EM}$	$L_{EM}$	$U_{EM}$
Times	39.84%	21.14%	39.02%
Georgia	42.15%	19.83%	38.02%
Arial	45.53%	18.70%	35.77%
Tahoma	44.62%	16.92%	38.46%
Courier	38.02%	23.97%	38.02%
Impact	53.91%	15.63%	30.47%
<b>Min</b>	<b>38.02%</b>	<b>15.63%</b>	<b>30.47%</b>
<b>Max</b>	<b>53.91%</b>	<b>23.97%</b>	<b>39.02%</b>

**Table 2** Typographic lines of the fonts relative to x-height.

Font face	$L_X = \frac{L_{EM}}{X_{EM}}$	$U_X = \frac{U_{EM}}{X_{EM}}$	$EM_X = 1/X_{EM}$
Times	53.06%	97.94%	251.00%
Georgia	47.05%	90.20%	237.25%
Arial	41.07%	78.56%	219.64%
Tahoma	37.92%	86.19%	224.11%
Courier	63.05%	100.00%	263.05%
Impact	28.99%	56.52%	185.51%
<b>Min</b>	<b>28.99%</b>	<b>56.52%</b>	<b>185.51%</b>
<b>Max</b>	<b>63.05%</b>	<b>100.00%</b>	<b>263.05%</b>

The choice of these fonts considered two aspects: the font set is either used by most printed documents or the font set exhibits the highest variation of parameters amongst the most widely used fonts. For the first, although a trustable reference that lists popular fonts was not found, the choice was based on the common sense of the authors. Regarding the second set of fonts, Impact and Courier New types were included as their imaginary lines differ strongly from the other font sets.

As Table 2 shows 1em value relative to the x-height is possible to make a link between the x-height and the typographic guidelines. Reference [3] claims that inter-word values should be between [0.2em; 0.25em]; reference [11] affirms that spacing between columns should be at least 1.5em. Table 3 presents the bounds for these spacing relative to x-height of the fonts studied herein.

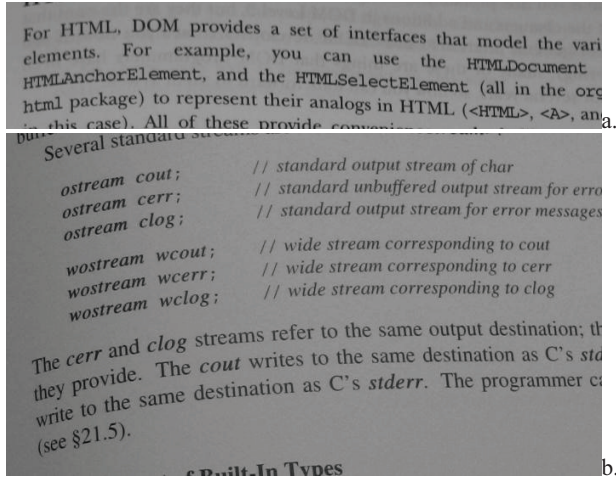
It is worth mentioning that the limit values are guidelines, not strict rules. The following scenarios represent exceptions:

- bad typographic design;
- advertisement design;
- mathematical equations in the text;
- digitalization errors;
- programming language source code, etc.

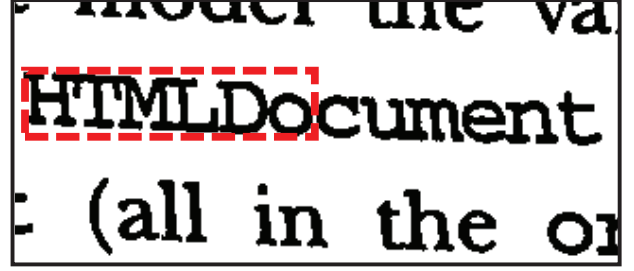
**Table 3** Spacing limit values relative to x-height of fonts.

Font face	EM <sub>x</sub>	Inter-letter limit (0.2em)	Inter-word limit (0.25em)	Inter-column limit (1.5em)
Times	251.00%	50.20%	62.75%	376.51%
Georgia	237.25%	47.45%	59.31%	355.87%
Arial	219.64%	43.93%	54.91%	329.45%
Tahoma	224.11%	44.82%	56.03%	336.17%
Courier	263.05%	52.61%	65.76%	394.57%
Impact	185.51%	37.10%	46.38%	278.27%
<b>Min</b>	<b>185.51%</b>	<b>37.10%</b>	<b>46.38%</b>	<b>278.27%</b>
<b>Max</b>	<b>263.05%</b>	<b>52.61%</b>	<b>65.76%</b>	<b>394.57%</b>

Figure 5 presents some of these scenarios extracted from CBDAR 2007 database [22]. Figure 6 shows the binary version of Figure 5.a available in the same database with letters smeared into the same connected components.



**Fig. 5** Non-standard typographic: higher spacing due to *HTMLAnchorElement* justification (a); source code alignment (b).



**Fig. 6** Figure 5.a binarization that “merged” characters.

### 3 Parallel lines regression

The parallel lines regression technique was introduced in the recent algorithm [18]. This section explains it with a simpler notation. The regression minimizes the error function of equation (1). Where  $m$  is the slope which is the same for both lines;  $c_T$  and  $c_B$  parallel lines intercept for top and bottom lines, respectively;  $minv = (u_{Tk}, v_{Tk})$  and  $maxv = (u_{Bk}, v_{Bk})$  are the top and bottom samples points relative to  $uv$  coordinates, respectively. The samples are in pairs of points, where  $N$  is the number of pairs.

$$E = \sum_{k=1}^N (m u_{Tk} + c_T - v_{Tk})^2 + \sum_{k=1}^N (m u_{Bk} + c_B - v_{Bk})^2 \quad (1)$$

The summation symbols are suppressed in equations (2) - (4) to simplify the notation.

$$a = \sum_{k=1}^N a_k \quad (2)$$

$$a^2 = \sum_{k=1}^N a_k^2 \quad (3)$$

$$[ab]_B = \sum_{k=1}^N a_{Bk} b_{Bk} \quad (4)$$

Making  $\partial E / \partial m = \partial E / \partial c_T = \partial E / \partial c_B = 0$ , results in eqs. (5)-(7).

$$m(u_T^2 + u_B^2) + c_T u_T + c_B u_B = [uv]_T + [uv]_B \quad (5)$$

$$m u_T + c_T N = v_T \quad (6)$$

$$m u_B + c_B N = v_B \quad (7)$$

Using Cramer's rule,  $m$ ,  $c_T$  and  $c_B$  values are obtained in eqs. (8)-(11).

$$\Delta = \begin{vmatrix} (u_T^2 + u_B^2) & u_T & u_B \\ u_T & N & 0 \\ u_B & 0 & N \end{vmatrix} \quad (8)$$

$$m = \frac{\Delta m}{\Delta}; \Delta m = \begin{vmatrix} ([uv]_T + [uv]_B) & u_T & u_B \\ v_T & N & 0 \\ v_B & 0 & N \end{vmatrix} \quad (9)$$

$$c_T = \frac{\Delta c_T}{\Delta};$$

$$\Delta c_T = \begin{vmatrix} (u_T^2 + u_B^2) & ([uv]_T + [uv]_B) & \Sigma_{x_B} \\ u_T & v_T & 0 \\ u_B & v_B & N \end{vmatrix} \quad (10)$$

$$c_B = \frac{\Delta c_B}{\Delta};$$

$$\Delta c_B = \begin{vmatrix} (u_T^2 + u_B^2) & u_T & ([uv]_T + [uv]_B) \\ u_T & N & v_T \\ u_B & 0 & v_B \end{vmatrix} \quad (11)$$

## 4 The New Algorithm

The new text-line segmentation algorithm takes as input a monochromatic image and performs the following steps:

1. Component Labeling;
2. Add components to multidimensional tree MDTREE;
3. Perform all k-nearest neighbors on MDTREE;
4. Create orientation neighbor priority queue;
5. For each neighbor on priority queue, group a text-line starting from them;
6. Identify text columns;
7. Merge text-lines that are in same text columns.

A multidimensional tree is used to perform k-nearest neighbors and range search efficiently. KD-tree and UB-trees are able to provide that [2] [21]. The computation of all k-nearest neighbors for document analysis was introduced by O’Gorman [17].

Reference [1] groups characters with similar properties walking through the document to find the upper and lower envelopes of text-lines. It uses independent linear regression for the detection, which may not work well when the starting point is a character whose case is different from the majority of those in the line, as may be observed in the example presented in Figure 7.a for finding envelopes of word “None”.



**Fig. 7** Neighbor issues: Different case sizes (a); ascend/descending stroke regression (b).

Such scenario is avoided by creating two priority queues: one for horizontal and other for vertical orientations. Once both queues

are filled out, the first queue to be consumed is the one which has more elements; the other is processed when the first becomes empty.

A component and its nearest neighbor will be added into a queue if their dimensions are compatible and they are close enough, with priority defined by equation 3. The ratio function is defined by equation 4. Assuming two neighboring blocks B and N, the priority is higher as:

- blocks dimensions of B and N are alike (inter block dimension similarity);
- and their width and height are similar (“squareness” of each block).

$$\text{priority}(B,N) = \text{ratio}(W_B, W_N) + \text{ratio}(H_B, H_N) + \frac{\text{ratio}(W_B, H_B) + \text{ratio}(W_N, H_N)}{2} \quad (12)$$

$$\text{ratio}(a,b) = \min(a,b)/\max(a,b) \quad (13)$$

The approach followed in reference [18], considers only the inter-block dimension similarity. However, there are some cases where the dimensions are close but the regression path does not replicate the mean and base lines (e.g. ascending followed by a descending stroke sequence as in Figure 7.b). Thus, it is better to process first blocks without ascending/descending strokes, such as letters “a”, “c”, etc). In this case, the aspect ratio of the letter case should have values close to 1:1, i.e. a square.

For a precise definition of “close enough”, two elements are defined: x-height upper bound and the percentage of allowed distance between neighbors characters (SFATOR). The former is set to the largest dimension of both blocks. The latter equals to inter-letter limit upper bound relative to the x-height, which is 52,61% (see Table 3). Neighbors that are farther away than this value have a high probability of not belonging to the same word. Parameters MIN\_RATIOW and MIN\_RATIOH rely on others that are explained foreword. Table 4 shows the derived formulas.

### Listing 1

Input: Blocks  
Output: if they are compatible

FUNCTION IS\_COMPATIBLE(Ci,Cj)

```
Let orientation be defined by
  GET_ORIENTATION(|Ci.x-Cj.x|,|Ci.y-Cj.y|);
Let xheightUpperBound =
  max(Ci.width,Ci.height,Cj.width,Cj.height);
RETURN
  ratio(Ci.uvwidth,Cj.uvwidth) ≥ MIN_RATIOW and
  ratio(Ci.uvheight,Cj.uvheight) ≥ MIN_RATIOH and
  Distance between rectangles defined by Ci and
  Cj is less than SFATOR×xheightUpperBound;
```

END FUNCTION

**Table 4** Algorithms derived parameters.

MIN_RATIOW	1/MAX_W	MIN_RATIOH	max(MIN_H, 1/MAX_H)
------------	---------	------------	---------------------

No pre-assumption is made regarding document orientation. The creation of horizontal and vertical priority queues helps to give better starting blocks for text-line formation regardless of document orientation. As the text-line is assembled, blocks with



the highest and lowest coordinates are recorded. Text-line orientation is obtained by listings 2 and 3.

**Listing 2**

Input: Abs. difference between coordinates  
Output: Text-line orientation

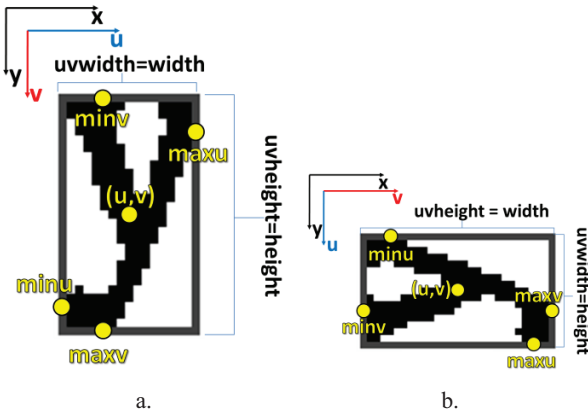
```
FUNCTION GET_ORIENTATION( $\Delta x, \Delta y$ )
  IF  $\Delta x > \Delta y$  THEN
    RETURN HORIZONTAL;
  ELSE
    RETURN VERTICAL;
  END IF
END FUNCTION
```

**Listing 3**

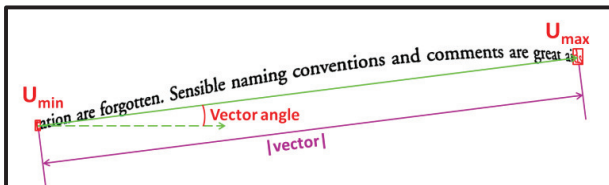
Input: Text-line  
Output: Line orientation

```
FUNCTION GET_TL_ORIENTATION(T)
//T.minX is the block with lowest x value
//T.minY is the block with lowest y value
 $\Delta x = \max(|T.minY.x - T.maxY.x|, |T.minX.x - T.maxX.x|);$ 
 $\Delta y = \max(|T.minY.y - T.maxY.y|, |T.minX.y - T.maxX.y|);$ 
RETURN GET_ORIENTATION( $\Delta x, \Delta y$ );
END FUNCTION
```

UV coordinates are defined according to the orientation; if it is horizontal the (u,v) values are the same as (x,y); if it is vertical (u,v) values are switched to (y,x). Figure 8 illustrates points with maximum and minimum (u,v) coordinates. One may notice that these are points that belong to the connected component with highest and lowest coordinates while some algorithms use the mean point of each rectangle side. Figure 9 shows one example of extreme u-coordinate blocks and the vector obtained by it in polar coordinates.



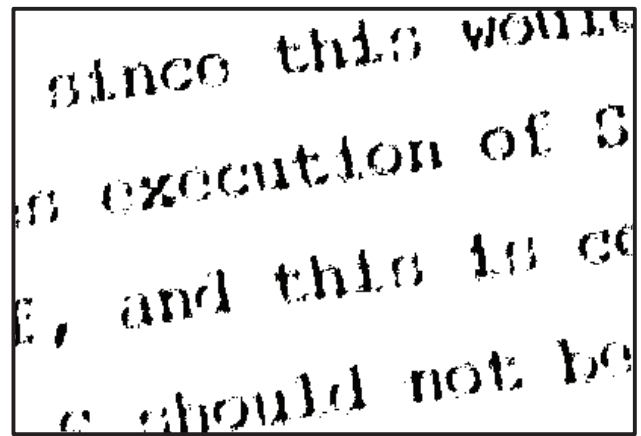
**Fig. 8** (u,v) coordinates for orientation = horizontal (a); (u,v) coordinates for orientation = vertical (b).



**Fig. 9** Text-line properties.

One important parameter is the W\_SIZE, which is the number of blocks used in the regression. Low values increase the variance of the parallel lines. Using too high, the estimation may not follow the curl of the text-lines. The average of letters per word in the English language is 4.5 [20]; in French is 4.84 and Spanish is 4.96 [26] (unreliable reference). Value of 7 was used for W\_SIZE that may include all blocks of words and include the last and first letters of words in the neighborhood.

The other parameter is k, it is related to the construction of k-nearest neighbor graph (K\_NN). O’Gorman [17] claims that for double line spacing value of 6 or 7 is enough; higher values add computational time and can be set for redundancy. Although, O’Gorman’s is applied to scanned documents, in this case, good binarization is achieved more easily. Herein, value of 8 is used due the digitalization errors that break characters such as the example presented on Figure 10.



**Fig. 10** Binarization with broken components of letters.

As the queue is processed, the text-lines are formed by parallel line regression using (u,v) coordinates. Text-line may be curled, but looking into a small vicinity of the characters the mean and base lines are approximately linear. The text-line is expanded by doing a regression with the  $U_{min}$  and  $U_{max}$  neighborhoods, alternately, as presented in Listing 4. In both expansions, a block is added when a k-neighbor of  $U_{min}$  or  $U_{max}$  has:

- uv width is below MAX\_W of the x-height, this value is empirically found in section 5 as a connected component may contain more than one letter such as in Figure 6 due binarization errors;
- uv height is below 263% of the x-height (MAX\_H), this value is the upper bound of  $EM_x$  as there is no character taller than 1em;
- typographic distance is below a threshold, the allowed values are between 65.76% (upper bound inter-word limit) and 278.27% (lower bound inter-column limit) relative to the x-height (MAX\_MDIST), the best value for CBDAR test-set is presented on section 5;
- is between the parallel lines defined by the corresponding vicinity with offset, which is set to 28.99% of the x-height (PR\_OFFSET) as this is the lower bound of font offsets defined in Table 2.

The previous restrictions apply to blocks that belong to a text-line. If all of these blocks are used in the regression, the x-height may be overestimated due to a sequence of small size components (e.g. subscripts, punctuation marks etc). On the other hand, those

elements are considered for the computation of the typographic distance of new blocks. Thus, two characters lists are attached to each text-line: one for the parallel lines regression, which is set to T.prBlocks; the other is the T.blocks, which includes all blocks. The parameter MIN\_H multiplied by the x-height defines:

- if the block has uv height below such threshold, it is added only to T.blocks list;
- otherwise, if it has uv height above such limit, it is added to both T.blocks and T.prBlocks lists.

The value for MIN\_H is defined empirically on section 5. Listing 4 provides the starting point of text-line extraction and Listing 5 adds blocks to both lists (T.prBlocks and T.blocks). Figures 11 and 12 illustrate how to compute the typographic distance and x-height, respectively. Figures 13 and 14 show examples for adding blocks in  $U_{min}$  and  $U_{max}$  directions, respectively.

**Listing 4**

Input: Neighbors  
Output: Text-line

```
PROCEDURE EXTRACT_TEXTLINE(Ci,Cj)
Create new text-line T;
T.blocks = {Ci, Cj};
T.prBlocks = {Ci, Cj};
WHILE a new letter is found for T DO
    FIND_LETTER(T, Umin); FIND_LETTER(T, Umax);
END WHILE
RETURN T;
END PROCEDURE
```

**Listing 5**

Input: Text-line and the block with min or max u  
Output: The text-line with neighbor of Ck added

```
PROCEDURE FIND_LETTER(T, Ck)

PR = parallel regression around Ck
xheight = PR.cb - PR.ct;
offset = xheight*PR.OFFSET;

IF there exists a neighbor block Nj of Ck such
    Nj.uvwidth ≤ xheight*MAX_W and
    Nj.uvheight ≤ xheight*MAX_H and
    mdist(Nj.u - Ck.u, PR.m) ≤ xheight*MAX_MDIST
    and (PR.m*Nj.u + PR.ct - offset) ≤ Nj.v
    and Nj.v ≤ (PR.m*Nj.u + PR.cb + offset) THEN

    T.blocks = T.blocks + {Ck};
    IF MIN_H*xheight ≤ Nj.uvheight THEN
        T.prBlocks = T.prBlocks + {Ck};
    END IF

END IF

END IF
END PROCEDURE
```

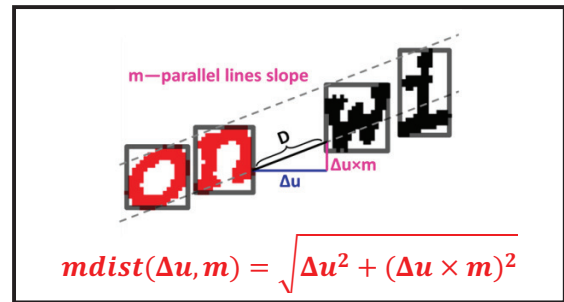


Fig. 11 Typographic distance definition.



Fig. 12 X-height calculation.

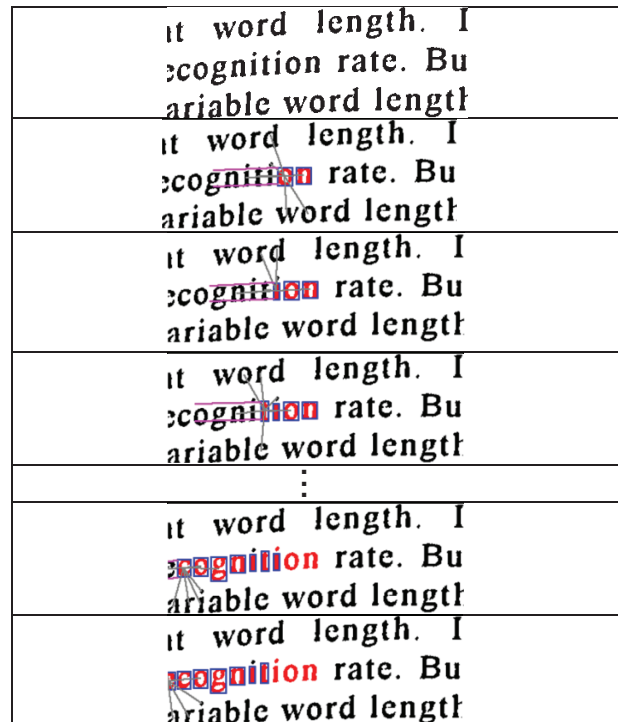
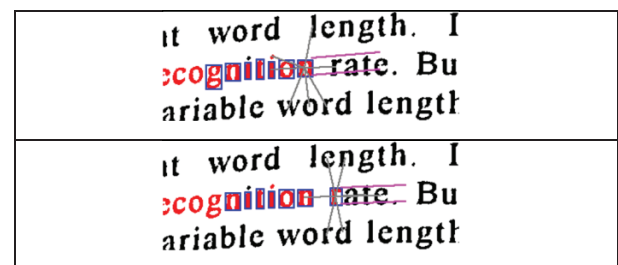


Fig. 13 Finding new blocks in  $U_{min}$  direction.



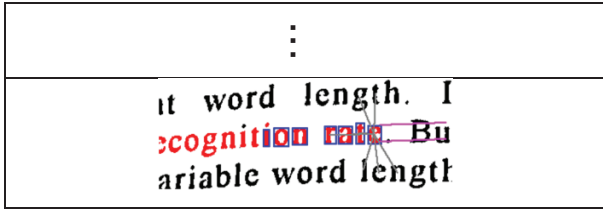


Fig. 14 Finding new blocks on  $U_{\max}$  direction.

A text column is identified by looking for parallel lines up and downwards in the  $v$  direction using the  $k$ -nearest neighbor that does not belong to the text-line. Listing 6 presents the pseudo code for that. A text column is transitively formed: two text-lines belong to the same column if they are parallel to each other; Listing 7 presents the pseudo code for that. Figure 15 exemplifies the preliminary segmentation followed by column extraction using this idea.

**Listing 6**

Input: A text-line  
Output: The text-line updated with parallel information

```
PROCEDURE EXTRACT_PARALLELS(T)

//T.minu and T.maxu are extreme u blocks
minu = T.min.u;
maxu = T.max.u;

FOR EACH Lt in T.prBlocks DO
  FOR EACH Ln in Lt.kneighbors
    IF Ln belongs to a text-line
      and minu ≤ Ln.u ≤ maxu THEN
      T.parallels = T.parallels +
        {Ln text-line};
    END IF
  END FOR
END FOR
END PROCEDURE
```

**Listing 7**

Input: A text-line  
Output: A set of text-lines that represents a text column

```
PROCEDURE EXTRACT_COLUMN(T)

COLUMN.textlines = empty set;
TLQUEUE = {T};

WHILE TLQUEUE is not empty DO
  Pop Ti from TLQUEUE;
  COLUMN.textlines = COLUMN.textlines + {Ti};
  FOR EACH Tj in Ti.parallels DO
    IF Ti ∈ Tj.parallels and
      Tj ∉ COLUMN.textlines THEN
      TLQUEUE = TLQUEUE + {Tj};
    END IF
  END FOR
END WHILE

END PROCEDURE
```

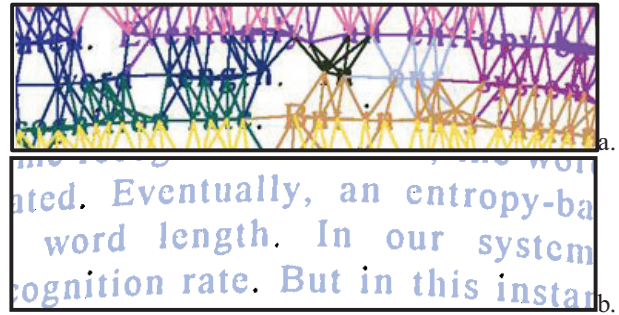


Fig. 15 Text column extraction: preliminary segmentation with neighbors (a); extracted block (b).

The text-line formation process may be split due to large inter-words distance caused by non-standard typography listed in section 2. In these cases, the  $k$ -nearest neighbor may not be able to connect those characters. A range search is done on the MDTREE for blocks that are between parallel lines in the vicinities of  $U_{\min}$  extremity and of  $U_{\max}$  to merge text-lines. Among these text-lines are merged together if there is a block that:

- belongs to another text-line;
- is in same text column;
- is not parallel to the current text-line.

This procedure is repeated until no merging is possible. It is worth noticing that MDTREE is a point data structure, i.e. range search verifies if the center of blocks are inside the query box. In the case of large blocks as in Figure 6, the search range must include their center. Therefore, the range ( $M\_RANGE$ ) is relative to the  $x$ -height and is set experimentally in section 5.

Listings 8 and 9 present the pseudo code of this procedure. Figure 16 illustrates this process. One may observe that the range search was rectified, but blocks outside parallel lines are not processed.

**Listing 8**

Input: Text column  
Output: Text-lines of the column merged

```
PROCEDURE MERGE_TEXTLINES(C)

FOR EACH Ti in C.textlines DO

  WHILE merging is possible DO

    //Search towards minu block
    Create parallel regression PR with blocks in
      Ti.prBlocks with W_SIZE-lowest u coords;
    searchRange = PR.xheight*M_RANGE;

    searchResult =
      Search on MDTREE blocks inside PR with
        offset of PR.xheight*PR_OFFSET and
        u=[Ti.minu-searchRange,Ti.minu];

    MERGE_NEIGHBOR_TL(Ti,searchResult);

    //Search towards maxu block
    Create parallel regression PR with blocks in
      Ti.prBlocks with W_SIZE-highest u coords;
    searchRange = PR.xheight*M_RANGE;

    searchResult =
```

```

Search on MDTREE blocks inside PR with
offset of PR.xheight*PR_OFFSET and
u=[Ti.maxu,Ti.maxu+searchRange];

MERGE_NEIGHBOR_TL(Ti,searchResult);

END WHILE
END FOR
END PROCEDURE

```

#### Listing 9

Input: Text-line and blocks from search result  
Output: Text-line merged with nearby text-lines

```

PROCEDURE MERGE_NEIGHBOR_TL(Ti,blocks)

FOR each block Bj in blocks DO
  IF Bj belongs to a text-line Tj and
  Ti  $\notin$  Tj.parallels and Tj  $\notin$  Ti.parallels THEN
    Merge Ti and Tj;
    exit procedure;
  END IF
END FOR
END PROCEDURE

```

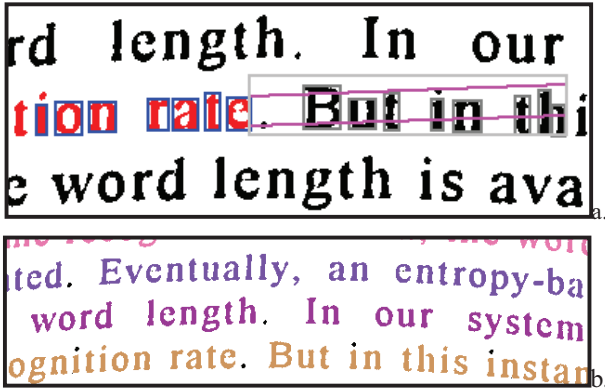


Fig. 16 Merging: searched blocks (a); text-lines merged (b).

Listing 10 presents the pseudocode for the algorithm.

#### Listing 10

```

Input: Binary image
Output: Text columns

PROCEDURE IDENTIFY_TEXTLINES()

Extract connect components of binary image;
Create multidimensional tree MDTREE with
connected components;
Create K-NNG of MDTREE with k set to K_NN;
Create orientation priority queue NQUEUE;

FOR EACH connected comp. Ci on MDTREE DO

  Let Cj be the nearest neighbor of Ci;

  IF IS_COMPATIBLE(Ci, Cj) THEN
    //NQUEUE priority defined by eq. (12)
    Add (Cj, Ci) to NQUEUE;
  END IF
END FOR

```

```

CREATE text-lines set TLS

WHILE NQUEUE is not empty DO
  Pop (Ci, Cj) elements on top of NQUEUE;
  IF Ci and Cj don't belong to a text-line THEN
    T = EXTRACT_TEXTLINE(Ci,Cj);
    TLS = TLS + {T};
  END IF
WHILE

FOR EACH text-line T on textLines DO
  EXTRACT_PARALLELS(T);
END FOR

CREATE columns set COLS

FOR EACH text-line T on textLines DO
  IF T does not belong to a column THEN
    C = EXTRACT_COLUMN(T);
    COLS = COLS + {C};
    MERGE_TEXTLINES(C);
  END IF
END FOR

END PROCEDURE

```

## 5 Results

References [8] and [9] compares the methods presented in [4][6][7][17][18][25] and another unpublished algorithm by the same authors using CBDAR de-warping dataset [22]. The same comparison methodology described in reference [23] is used here. The text-line information is stored in a color image, in which each line is painted with a different color. Ground truth and processed images are compared by a pixel-correspondence graph, where each node represents a text-line in both images; an edge between two lines exists if they share a pixel with weight set to the total number of pixels shared. Black and white are not used for text-line colors; they stand for noise and background pixels, respectively. An incoming edge is significant if  $w_i/P \geq Tr$  and  $w_i \geq Ta$ , where  $w_i$  is the weight of the edge;  $P$  is the total number of node pixels;  $Tr$  and  $Ta$  are the relative and absolute thresholds. The following parameters are computed (following [8]). Figure 17 exemplifies text-line segmentation errors.

- Number of ground truth lines ( $N_g$ ) – total number ground truth lines in the whole database.
- Number of ground truth lines ( $N_s$ ) – total number segmented components that are not false positive;
- Total correct segmentation ( $N_{o2o}$ ) – the number of one-to-one matches between the ground-truth components and the segmentation components.
- Total over segmentations ( $N_{os}$ ) – the number of significant edges that ground truth lines have, minus the number of ground truth lines.
- Total under segmentations ( $N_{us}$ ): the number of significant edges that segmented lines has minus the number of segmented lines.
- Over segmented components ( $N_{oc}$ ): the number of ground truth lines having more than one significant edge.
- Under segmented components ( $N_{uc}$ ): the number of segmented lines having more than one significant edge.
- Missed components ( $N_{mc}$ ): the number of ground truth components that matched the background in the hypothesized segmentation.

- False alarms ( $N_{fa}$ ): the number of components in the hypothesize segmentation that did not match any foreground component in the ground-truth segmentation.
- % correct segmentation ( $P_{o2o} = N_{o2o}/N_g$ )
- % over-segmented text-lines ( $P_{oc} = N_{oc}/N_g$ )
- % under-segmented text-lines ( $P_{uc} = N_{uc}/N_g$ )
- % missed text-lines ( $P_{mc} = N_{mc}/N_g$ )

**Correct segmentation without noise...**  
**Correct segmentation with wrong noise that increases  $P_{oc}$ ...**  
**Over segmentation**  
**Under segmentation**  
**in different lines**  
**123 Under segmentation in the same "line"**  
**missed lines happens when it is marked as noise**

**Fig. 17** Illustration of segmentation lines with noise in black pixels and detected information in color.

In addition to those data, three new ones are calculated [15]: precision, recall and f-measure. Their definitions are available in equations (14)-(16).

$$PR = precision = \frac{correct\ lines\ (N_{o2o})}{hypothesis\ lines\ (N_s + N_{falarm})} \quad (14)$$

$$RE = recall = \frac{correct\ lines\ (N_{o2o})}{reference\ lines\ (N_g)} \quad (15)$$

$$FM = f - measure = 2 \times \frac{precision \times recall}{(precision + recall)} \quad (16)$$

Where:

- the number of correct lines is  $N_{o2o}$ , it is equal to  $P_{o2o} \times N_g$ ;
- the total hypothesis lines is equal to  $N_s$  summed with  $N_{fa}$ , as  $N_s$  is "total number segmented components that are **not false positive [i.e. false alarms]**";
- the reference lines value is equivalent to the number of ground truth lines ( $N_g$ ).

Some parameters are exploited in the next section.

### 5.1 Parameter study

The parameters are studied using values of table 5 as starting values. Their values were empirically found to be good. The cells underlined represent the parameters exploited herein. Table 6 shows the interval of them.

**Table 5** Algorithms standard parameters values.

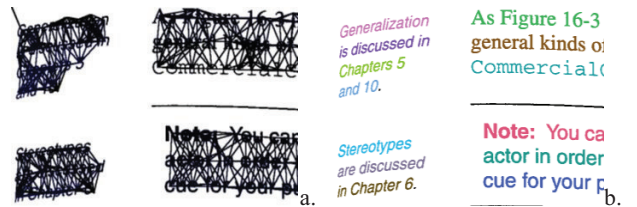
SFACTOR	<u>0.52</u>	K_NN	<u>8</u>	MAX_W	<u>9.0</u>
MAX_H	<u>2.63</u>	MIN_H	<u>0.7</u>	W_SIZE	<u>7</u>
MAX_MDIST	<u>1.97</u>	M_RANGE	<u>5.26</u>	PR_OFFSET	<u>0.2899</u>

**Table 6** Algorithms parameters variation.

Parameter	Minimum	Maximum
M_RANGE	3.9453 (upper bound inter-column limit, i.e. 1.5em)	14.4661 (upper bound for 5.5em)
MAX_MDIST	0.6576 (upper bound inter-word limit)	2.7827 (lower bound inter-column limit)
MAX_W	1.00 (no character is greater than 1em)	12.00 (higher limit arbitrary applied)
MIN_H	0.00 (include all blocks in regression)	1.00 (include only blocks that are greater than x-height in the regression)

The number of groundtruth components for all tests equals to 3103, it does not change according to the tests. Tables 7 and 8 show, respectively, Shafait *et al* and proposed data measured with different parameters. Bold font weight indicates the best results value among the variation. It is possible to conclude:

- The previous algorithm [18] defines the "Maximum distance between letters" parameter that is similar to MAX\_MDIST. High values for it merge text-lines of different columns. Figure 18.a shows the nearest neighbor of texts and left notes. One may see that there is no neighbor shared between different columns, this one of the advantages of using k-nearest neighbor that allows higher values for MAX\_MDIST. Figure 18.b shows a processing result of MAX\_MDIST=2.7827.
- Choosing large values for both MAX\_MDIST and M\_RANGE increases under-segmentation ( $N_{us}$ ). Figure 19 shows a page header with nearest neighbor of its components when MAX\_MDIST=2.78; this does not happen for M\_RANGE=14.47. The low density of blocks causes an "over neighboring" effect, thus the first parameter have greater influence into under-segmentation metric.
- High values for MAX\_W decrease errors which turns acceptance of blocks less restrict when compared to lower values. Figure 20 shows a bad binarized line correctly identified.
- Value for MIN\_H must be chosen with care due: it is not good to accept small characters (e.g. subscripts, punctuation marks) for regression; it is avoidable rejecting letters or numbers for the regression. The absence of this parameter (i.e. MIN\_H=0.00) have higher over segmentation ( $N_{os}$ ) and under-segmentation ( $N_{us}$ ) values.



**Fig. 18** Illustration of segmentation lines with noise in black pixels and detected information in color.



**Fig. 19** Page header merged with nearest neighbor.

```

catch (ClassNotFoundException e) {
    System.out.println("Could not load");
}
catch (MalformedURLException e) {
    System.out.println(
        "Incorrect URL for XML-RPC se
        -message());
}

```

Fig. 20 Text-line extraction with characters grouped together.

Table 7 Shafait *et al* [23] data for parameter variations.

P.	Val.	N <sub>s</sub>	P <sub>o2o</sub>	P <sub>oc</sub>	P <sub>uc</sub>	P <sub>mc</sub>	N <sub>os</sub>	N <sub>us</sub>	N <sub>fa</sub>
M_RANGE	3.95	3091	<b>97.81%</b>	1.71%	<b>0.26%</b>	0.90%	55	<b>9</b>	740
	6.58	3091	97.78%	<b>1.68%</b>	<b>0.26%</b>	0.90%	<b>54</b>	<b>9</b>	731
	9.21	3089	97.68%	1.71%	0.32%	0.90%	<b>54</b>	11	724
	11.84	3088	97.62%	1.71%	0.35%	0.90%	<b>54</b>	12	720
	14.47	3086	97.49%	1.71%	0.42%	0.90%	<b>54</b>	14	<b>710</b>
MAX_MDIST	0.66	6192	57.52%	44.51%	1.45%	0.90%	3677	55	1062
	0.87	4143	82.92%	19.37%	0.55%	0.90%	1331	20	985
	1.08	3643	91.52%	10.18%	0.06%	0.90%	712	3	916
	1.30	3393	94.10%	6.67%	<b>0.03%</b>	0.90%	408	<b>2</b>	840
	1.51	3198	96.13%	4.48%	<b>0.03%</b>	0.90%	191	<b>2</b>	785
	1.72	3119	97.55%	2.67%	0.06%	0.90%	92	3	764
	<b>1.93</b>	3095	<b>97.84%</b>	1.77%	0.19%	0.90%	57	7	735
	2.15	3086	97.68%	1.45%	0.35%	<b>0.87%</b>	46	12	720
	2.36	3082	97.71%	1.32%	0.39%	<b>0.87%</b>	41	13	713
	2.57	3076	97.65%	1.19%	0.45%	<b>0.87%</b>	37	16	702
2.78	3075	97.68%	<b>1.16%</b>	0.45%	<b>0.87%</b>	<b>36</b>	16	<b>692</b>	
MAX_W	2.63	3126	96.97%	7.06%	<b>0.23%</b>	0.90%	250	<b>7</b>	<b>698</b>
	3.57	3101	97.49%	2.42%	0.26%	0.90%	86	9	716
	4.50	3096	97.62%	1.90%	0.26%	0.90%	66	9	738
	5.44	3093	97.71%	1.80%	0.26%	0.90%	59	9	736
	6.38	3092	97.74%	1.77%	0.26%	0.90%	58	9	736
	7.32	3092	97.74%	1.71%	0.26%	0.90%	56	9	735
	8.25	3092	97.74%	1.71%	0.26%	0.90%	56	9	734
	9.19	3091	<b>97.78%</b>	<b>1.68%</b>	0.26%	0.90%	<b>54</b>	9	734
	10.13	3091	<b>97.78%</b>	<b>1.68%</b>	0.26%	0.90%	<b>54</b>	9	734
	11.06	3091	<b>97.78%</b>	<b>1.68%</b>	0.26%	0.90%	<b>54</b>	9	733
12.00	3091	<b>97.78%</b>	<b>1.68%</b>	0.26%	0.90%	<b>54</b>	9	732	
MIN_H	0.00	3233	95.00%	4.77%	0.35%	<b>0.81%</b>	217	11	881
	0.10	3226	95.10%	4.67%	0.35%	<b>0.81%</b>	209	11	881
	0.20	3201	95.75%	3.90%	0.32%	<b>0.81%</b>	173	10	873
	0.30	3151	96.87%	2.51%	0.29%	<b>0.81%</b>	109	10	862
	0.40	3120	97.49%	2.00%	<b>0.26%</b>	<b>0.81%</b>	74	<b>9</b>	848
	0.50	3099	97.74%	1.74%	<b>0.26%</b>	0.84%	60	<b>9</b>	822
	0.60	3097	<b>97.78%</b>	1.74%	<b>0.26%</b>	0.84%	59	<b>9</b>	796
	0.70	3091	<b>97.78%</b>	<b>1.68%</b>	<b>0.26%</b>	0.90%	<b>54</b>	<b>9</b>	734
	0.80	3094	96.94%	2.74%	0.55%	1.13%	89	17	659
	0.90	3138	92.46%	8.38%	2.38%	1.45%	302	76	559
1.00	3107	77.83%	22.62%	7.06%	5.70%	887	248	361	

SFACTOR=0.52, MAX\_H=2.63, W\_SIZE=7,  
 PR\_OFFSET=0.2899, M\_RANGE=5.26\*, MAX\_MDIST=1.97\*,  
 MIN\_H=0.7\*, MAX\_W=9.0\*

\* Except when P. (parameter) and Val. (value) specify different

Tables 7 and 8 show data values for different setups. The low values of precision are mainly because dataset ground truth defines illustrations with text as noise (e.g. Figure 22). Therefore, a large amount of lines are identified as text-lines and are considered false alarms decreasing precision metric.

A new setup is presented in Tables 9 and 10. Values of MAX\_W and M\_RANGE are changed to the ones with highest f-measures in Table 8. The other two remaining were varied into smaller intervals with minor step between the samples. Values of 0.700 were repeated for MIN\_H; 1.93 and 1.97 for MAX\_MDIST.

Table 8 Precision, recall and f-measure for parameter variations

P.	Val.	N <sub>o2o</sub>	N <sub>g</sub>	Hypo.	PR	RE	FM
M_RANGE	3.95	3035	3103	3831	79.22%	<b>97.81%</b>	87.54%
	6.58	3034	3103	3822	79.38%	97.78%	87.62%
	9.21	3031	3103	3813	79.49%	97.68%	87.65%
	11.84	3029	3103	3808	79.54%	97.62%	87.66%
	14.47	3025	3103	3796	<b>79.69%</b>	97.49%	<b>87.69%</b>
MAX_MDIST	0.66	1785	3103	7254	24.61%	57.52%	34.47%
	0.87	2573	3103	5128	50.18%	82.92%	62.52%
	1.08	2840	3103	4559	62.29%	91.52%	74.13%
	1.30	2920	3103	4233	68.98%	94.10%	79.61%
	1.51	2983	3103	3983	74.89%	96.13%	84.19%
	1.72	3027	3103	3883	77.96%	97.55%	86.66%
	1.93	3036	3103	3830	79.27%	<b>97.84%</b>	87.58%
	2.15	3031	3103	3806	79.64%	97.68%	87.74%
	2.36	3032	3103	3795	79.89%	97.71%	87.91%
	2.57	3030	3103	3778	80.20%	97.65%	88.07%
2.78	3031	3103	3767	<b>80.46%</b>	97.68%	<b>88.24%</b>	
MAX_W	2.63	3009	3103	3824	78.69%	96.97%	86.88%
	3.57	3025	3103	3817	79.25%	97.49%	87.43%
	4.50	3029	3103	3834	79.00%	97.62%	87.33%
	5.44	3032	3103	3829	79.19%	97.71%	87.48%
	6.38	3033	3103	3828	79.23%	97.74%	87.52%
	7.32	3033	3103	3827	79.25%	97.74%	87.53%
	8.25	3033	3103	3826	79.27%	97.74%	87.55%
	9.19	3034	3103	3825	79.32%	<b>97.78%</b>	87.59%
	10.13	3034	3103	3825	79.32%	<b>97.78%</b>	87.59%
	11.06	3034	3103	3824	79.34%	<b>97.78%</b>	87.60%
12.00	3034	3103	3823	<b>79.36%</b>	<b>97.78%</b>	<b>87.61%</b>	
MIN_H	0.00	2948	3103	4114	71.66%	95.00%	81.70%
	0.10	2951	3103	4107	71.85%	95.10%	81.86%
	0.20	2971	3103	4074	72.93%	95.75%	82.79%
	0.30	3006	3103	4013	74.91%	96.87%	84.49%
	0.40	3025	3103	3968	76.23%	97.49%	85.56%
	0.50	3033	3103	3921	77.35%	97.74%	86.36%
	0.60	3034	3103	3893	77.93%	<b>97.78%</b>	86.74%
	0.70	3034	3103	3825	79.32%	<b>97.78%</b>	87.59%
	0.80	3008	3103	3753	<b>80.15%</b>	96.94%	<b>87.75%</b>
	0.90	2869	3103	3697	77.60%	92.46%	84.38%
1.00	2415	3103	3468	69.64%	77.83%	73.50%	

SFACTOR=0.52, MAX\_H=2.63, W\_SIZE=7,  
 PR\_OFFSET=0.2899, M\_RANGE=5.26\*, MAX\_MDIST=1.97\*,  
 MIN\_H=0.7\*, MAX\_W=9.0\*

\* Except when P. (parameter) and Val. (value) specify different

**Table 9** Second exploitation of Shafait *et al* [23] measures.

P.	Val.	N <sub>s</sub>	P <sub>o2o</sub>	P <sub>oc</sub>	P <sub>uc</sub>	P <sub>mc</sub>	N <sub>os</sub>	N <sub>us</sub>	N <sub>fa</sub>
MIN_H	0.500	3080	97.39%	1.26%	0.58%	<b>0.81%</b>	42	20	742
	0.575	3081	<b>97.45%</b>	1.26%	<b>0.55%</b>	<b>0.81%</b>	42	<b>19</b>	722
	0.650	3075	97.42%	<b>1.19%</b>	0.58%	0.84%	38	20	687
	0.700	3072	97.42%	<b>1.19%</b>	0.58%	0.87%	<b>37</b>	20	659
	0.725	3072	97.33%	1.26%	0.61%	0.93%	39	21	645
	0.800	3072	96.33%	2.22%	1.10%	1.06%	73	36	<b>606</b>
MAX_MDIST	1.2951	3356	94.42%	6.25%	0.26%	0.87%	374	9	823
	1.4439	3238	95.39%	5.16%	0.19%	<b>0.84%</b>	239	7	794
	1.5926	3148	96.94%	3.45%	<b>0.13%</b>	0.90%	131	<b>5</b>	761
	1.7414	3110	97.33%	2.58%	0.23%	0.90%	86	8	737
	1.8901	3093	97.65%	1.87%	0.29%	0.90%	59	10	712
	1.9327	3090	97.55%	1.80%	0.35%	0.90%	57	12	709
	1.9514	3087	97.45%	1.77%	0.42%	0.90%	56	14	708
	1.9700	3086	<b>97.49%</b>	1.71%	0.42%	0.90%	54	14	708
	2.0389	3084	97.33%	1.64%	0.52%	0.87%	52	17	700
	2.1877	3080	97.42%	1.39%	0.52%	0.87%	43	17	687
	2.3364	3079	97.45%	1.35%	0.52%	0.87%	42	17	682
	2.4852	3077	97.52%	1.26%	0.52%	0.87%	39	17	670
	2.6339	3072	97.42%	<b>1.19%</b>	0.58%	0.87%	<b>37</b>	20	664
	2.7827	3072	97.42%	<b>1.19%</b>	0.58%	0.87%	<b>37</b>	20	<b>659</b>

SFACTOR=0.52, MAX\_H=2.63, W\_SIZE=7,  
PR\_OFFSET=0.2899, M\_RANGE=14.47, MAX\_MDIST=1.93\*,  
MIN\_H=0.7\*, MAX\_W=12.0

\* Except when P. (parameter) and Val. (value) specify different

**Table 10** Second exploitation of precision/recall measures.

P.	Val.	N <sub>o2o</sub>	N <sub>g</sub>	Hypo.	PR	RE	FM
MIN_H	0.500	3022	3103	3822	79.07%	97.39%	87.28%
	0.575	3024	3103	3803	79.52%	<b>97.45%</b>	87.58%
	0.650	3023	3103	3762	80.36%	97.42%	88.07%
	0.700	3023	3103	3731	81.02%	97.42%	88.47%
	0.725	3020	3103	3717	81.25%	97.33%	<b>88.56%</b>
	0.800	2989	3103	3678	<b>81.27%</b>	96.33%	88.16%
MAX_MDIST	1.2951	2930	3103	4179	70.11%	94.42%	80.47%
	1.4439	2960	3103	4032	73.41%	95.39%	82.97%
	1.5926	3008	3103	3909	76.95%	96.94%	85.80%
	1.7414	3020	3103	3847	78.50%	97.33%	86.91%
	1.8901	3030	3103	3805	79.63%	97.65%	87.72%
	1.9327	3027	3103	3799	79.68%	97.55%	87.71%
	1.9514	3024	3103	3795	79.68%	97.45%	87.68%
	1.9700	3025	3103	3794	79.73%	<b>97.49%</b>	87.72%
	2.0389	3020	3103	3784	79.81%	97.33%	87.70%
	2.1877	3023	3103	3767	80.25%	97.42%	88.01%
	2.3364	3024	3103	3761	80.40%	97.45%	88.11%
	2.4852	3026	3103	3747	80.76%	97.52%	88.35%
	2.6339	3023	3103	3736	80.92%	97.42%	88.40%
	2.7827	3023	3103	3731	<b>81.02%</b>	97.42%	<b>88.47%</b>

SFACTOR=0.52, MAX\_H=2.63, W\_SIZE=7,  
PR\_OFFSET=0.2899, M\_RANGE=14.47, MAX\_MDIST=1.93\*,  
MIN\_H=0.7\*, MAX\_W=12.0

\* Except when P. (parameter) and Val. (value) specify different

## 5.1 Comparison with other methods

Tables 13 and 14 show the results of the new algorithm and others published in [8], best values are in bold, where:

- N. RE denotes the new algorithm with best recall;
- N. FM represents the new algorithm with best f-measure;
- *prev.* stands for the result of algorithm [18];
- R(G) and R(B) for [7] the segmentation in grayscale and binary images, respectively;
- B-S for baby snakes [6];
- C-S for coupled snakes [4];
- C-S v2 for the extended version of coupled snakes that is on review phase for IJDAR [8];
- N-D for Neighborhood-Distance based algorithm [25];
- Doct. for Doctsum [17].

It is worth remarking that the first author of this paper detected that 0.5% of the ground truth text-lines were incorrectly annotated by the benchmark developers [8][9]. Therefore, P<sub>o2o</sub>, P<sub>oc</sub> and P<sub>uc</sub> percentages of other algorithms may oscillate in about 0.5%. Value for N<sub>g</sub> is 3091 for them; it is explicit in Table 14. In the case of the new algorithm, this value equals 3103.

**Table 11** Algorithms comparison metrics.

Algo.	N <sub>s</sub>	P <sub>o2o</sub>	P <sub>oc</sub>	P <sub>uc</sub>	P <sub>mc</sub>	N <sub>os</sub>	N <sub>us</sub>	N <sub>fa</sub>
N. RE	3095	<b>97.84%</b>	1.77%	<b>0.19%</b>	0.90%	57	<b>7</b>	735
N. FM	3072	97.33%	<b>1.26%</b>	0.61%	0.93%	<b>39</b>	21	<b>645</b>
Prev.	2924	91.10%	21.71%	1.81%	4.43%	682	57	785
R (G)	3045	89.10%	3.53%	3.85%	0.91%	115	131	1186
N-D	3134	93.47%	3.33%	0.55%	2.59%	134	18	2491
R (B)	3115	89.65%	4.40%	3.30%	0.29%	144	110	2183
C-Sv2	3093	95.21%	1.68%	1.59%	<b>0%</b>	54	51	3277
C-S	2799	78.26%	<b>1.26%</b>	9.06%	<b>0%</b>	<b>39</b>	359	3251
B-S	3371	87.58%	5.79%	2.91%	<b>0%</b>	294	117	13199
Doct.	6852	21.26%	66.90%	51.50%	<b>0%</b>	4383	2096	6066

**Table 12** Algorithms comparison with new metrics.

Algo.	N <sub>o2o</sub>	N <sub>g</sub>	Hypo.	PR	RE	FM
N. RE	3036.00	3103	3830	79.27%	<b>97.84%</b>	87.58%
N. FM	3020.00	3103	3717	<b>81.25%</b>	97.33%	<b>88.56%</b>
Prev.	2815.90	3091	3709	75.92%	91.10%	82.82%
R (G)	2754.08	3091	4231	65.09%	89.10%	75.23%
N-D	2889.16	3091	5625	51.36%	93.47%	66.30%
R (B)	2771.08	3091	5298	52.30%	89.65%	66.06%
C-S v2	2942.94	3091	6370	46.20%	95.21%	62.21%
C-S	2419.02	3091	6050	39.98%	78.26%	52.93%
B-S	2707.10	3091	16570	16.34%	87.58%	27.54%
Doctsum	657.15	3091	12918	5.09%	21.26%	8.21%

Reference [8][9] set recall as the main measure. The authors of this article claim that F-measure should be considered instead; recall does not include false alarms in its definition. The new and previous algorithms have better performance than others for precision and f-measure; the new algorithm was superior when compared to the previous version.

As for the other measures, the proposed method has the best performance for under segmentation and correct segmentation

figures, with no parallel line merging registered. The missed components of the proposed algorithm are characterized by small lines and mixture of letters and punctuation marks as shown in Figure 21.

Another advantage is the lower number of false alarms compared with competing algorithms. Figure 22 presents an example of the error in new approach, where one may observe that is hard to distinguish textual illustrations from the main text.

Another advance, when comparing the previous and new algorithms, regards the decrease of over segmented components ( $P_{oc}$  and  $N_{os}$ ). It was obtained due the inclusion of punctuation marks by the new algorithms.

An example of successful ( $P_{o20}=100\%$ ) processing can be seen in Figure 23 with noisy pixels in black. This picture was processed in 0.950s over a standard laptop (Dell Inspirion 1525 running Windows 7 Professional and Celeron 550 2.0GHz processor). The average execution time for the CBDAR 2007 benchmark images is 4s (the references on all the other methods do not provide execution times).



Fig. 21 Line missing examples: lines with punctuation marks and a single character (a); undersegmented binarization (b); page number with a single digit (c).

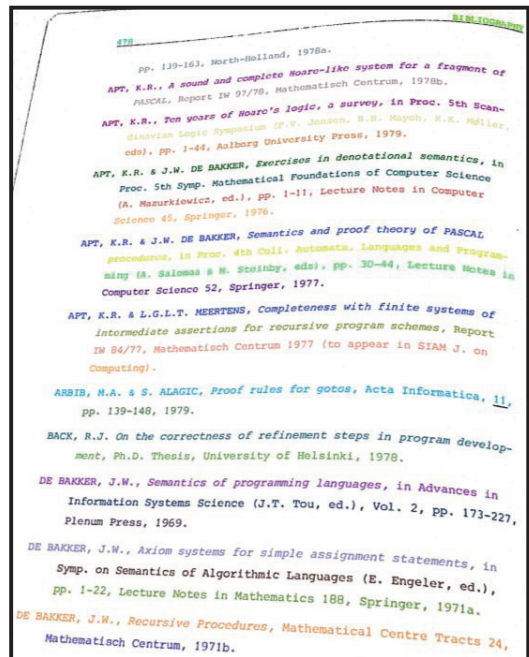


Fig. 23 Example of successful processing.

## 6 Conclusions

A new algorithm for text-line segmentation was presented. It outperforms the other state-of-the-art algorithms in precision, recall and f-measure metrics. It can automatically detect text lines of documents with any orientation. The time efficiency of the algorithm is high, a non-optimized Java implementation took on average 4s for CBDAR 2007 images, processed in a laptop (Dell Inspirion 1525 running Windows 7 Professional and Celeron 550 2.0GHz processor). The new process was used successfully in correcting the binding distortion in scanned books [14] where the document orientation is arbitrary.

## References

- [1] Ávila BT, Lins RD. A fast orientation and skew detection algorithm for monochromatic document images. Proceedings of the ACM Symposium on Document Engineering, pp. 118-126, 2005.
- [2] De Berg M, Otfried C, van Kreveld M, Overmars M. Computational Geometry: Algorithms and Applications, 3rd Edition, pages 99-101. Springer, 2008.
- [3] Bringhurst R. The Elements of Typographic Style. 3<sup>rd</sup> edition, Hartley & Marks Publishers, 2004.
- [4] Bukhari SS, Shafait F, Breuel TM. Coupled snakelet model for curled textline segmentation of camera-captured document images. In Proceedings 10th International Conference on Document Analysis and Recognition, pp. 61-65, Barcelona, Spain, 2009.
- [5] Bukhari SS, Shafait F, Breuel TM. Ridges based curled textline region detection from grayscale camera-captured document images. In Proc. The 13th International

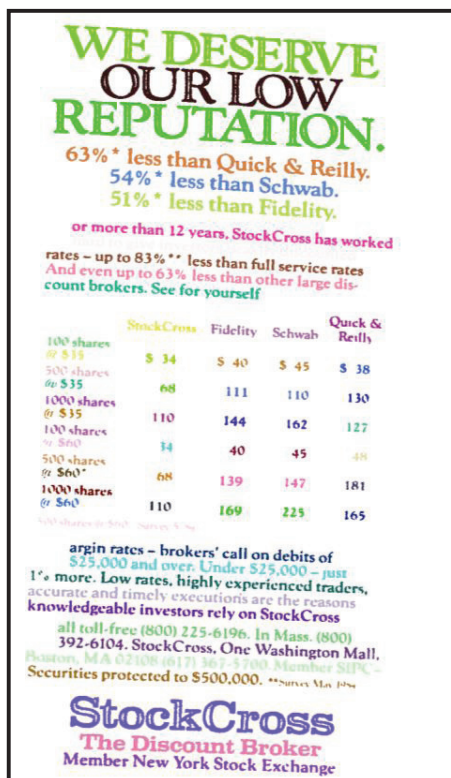


Fig. 22 False alarm sample.



- Conference on Computer Analysis of Images and Patterns, LNCS 5702, pp. 173–180, Springer Verlag, 2009.
- [6] Bukhari SS, Shafait F, Breuel TM. Segmentation of curled textlines using active contours. In Proceedings 8th IAPR Workshop on Document Analysis Systems, pp. 270–277, Nara, Japan, 2008.
- [7] Bukhari SS, Shafait F, Breuel TM. Textline information extraction from grayscale camera-captured document images. In Proc. The 13th International Conference on Image Processing, 2009.
- [8] Bukhari SS, Shafait F, Breuel TM. Performance Evaluation of curled textline segmentation algorithms on CBDAR 2007 dewarping contest dataset. 17th Int. Conf. on Image Processing, ICIP'10. Hong Kong, China, Sep. 2010.
- [9] Bukhari SS, Shafait F, Breuel TM. Performance Evaluation of Curled Textlines Segmentation Algorithms, 9th IAPR Workshop on Document Analysis Systems, 2010.
- [10] Chandrashekar R, Mohiaddin RH, Rueckert D. Analysis of myocardial motion in tagged MR images using nonrigid image registration. Progress in biomedical optics and imaging, 3(2):1168–1179, 2002.
- [11] French N. InDesign Type: Professional Typography with Adobe InDesign CS2. pp. 219, Adobe Press, March, 2006.
- [12] Fu B, Wu M, Li R, Li W, Xu Z. A model-based book dewarping method using text line detection. In 2nd Int. Workshop on Camera-Based Document Analysis and Recognition, Curitiba, Brazil, Sep. 2007.
- [13] Hohnhäuser B, Hommel G. 3D pose estimation using coupled snakes. J. WSCG, 12(1-3):1213–6972, Feb 2003.
- [14] Lins RD, Oliveira DM, Torreão G, Fan J, Thielo M. Correcting Book Binding Distortion in Scanned Documents. Proceedings of ICIAR 2010. LNCS 6112, pp. 355-365, Springer Verlag, 2010.
- [15] Makhoul J, Kubala F, Schwartz R, Weischedel R. Performance measures for information extraction. Proc. DARPA Workshop on Broadcast News Understanding. San Mateo, CA: Morgan Kaufmann, Mar., pp. 249–252, 1999.
- [16] Masalovitch A, Mestetskiy L. Usage of continuous skeletal image representation for document images de-warping. Proceedings of International Workshop on Camera-Based Document Analysis and Recognition, Curitiba, 2007, pp. 45-53.
- [17] O’Gorman L. The Document Spectrum for Page Layout Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 15(11), p. 1162-1173, 1993;
- [18] Oliveira DM, Lins RD, Torreão G, Fan J, Thielo M. A New Method for Text-line Segmentation for Warped Documents. Proceedings of ICIAR 2010. LNCS 6112, pp. 398-408, Springer Verlag, 2010.
- [19] Oliveira DM, Lins RD, Torreão G, Fan J, Thielo M. A New Algorithm for Segmenting Warped Text-lines in Document Images. ACM SAC 2011, Taichung, Taiwan.
- [20] Pierce JR. An Introduction to Information Theory: Symbols, Signals and Noise. Dover Publications, Inc., New York second edition, p. 75, 1980.
- [21] Ramsak F. Towards a general-purpose, multidimensional index: Integration, Optimization, and Enhancement of UB-Trees. Munich University of Technology. PhD Thesis. Munich , 2002.
- [22] Shafait F, Breuel TM. Document Image Dewarping Contest. 2nd Int. Workshop on Camera-Based Document Analysis and Recognition, CBDAR 2007, Brazil. Sep. 2007. pp 181-188.
- [23] Shafait F, Keysers D, Breuel TM. Performance evaluation and benchmarking of six page segmentation algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(6):941–954, Jun 2008.
- [24] Stamatopoulos N, Gatos B, Pratikakis I, Perantonis SJ. A two-step de-warping of camera document images. Proceedings 8th IAPR Workshop on Document Analysis Systems, pp. 209–216, Nara, Japan, 2008.
- [25] Ulges A, Lampert CH, Breuel TM. Document image dewarping using robust estimation of curled text lines. Proc. Eighth Int. Conf. on Document Analysis and Recognition, Aug. 2005, pp. 1001–1005.
- [26] Unknown Author. Letter frequency per 1000 words. Available at: <http://www.cs.trincoll.edu/~crypto/resources/LetFreq.html>. Accessed on: 6<sup>th</sup> of December of 2010.
- [27] Naylor M. Typographic line terms. Available at [http://en.wikipedia.org/wiki/File:Typography\\_Line\\_Terms.svg](http://en.wikipedia.org/wiki/File:Typography_Line_Terms.svg).

D. M. Oliveira, R. D. Lins e G. S. Torreão. *Characters' Shape Analysis of Printed Documents*. (Submetido para ser publicado no ICPR 2012).

# Characters' Shape Analysis of Printed Documents

Daniel Marques Oliveira, Rafael Dueire Lins, Gabriel Torreão  
Universidade Federal de Pernambuco, Recife, Pernambuco, Brazil  
{daniel.moliveira, rdl, gabriel.dsilva}@ufpe.br

## Abstract

Many algorithms in document image processing arbitrarily assume typographical elements as parameters, without being explicit about them most of times. This work presents the result of statistical analysis for bounds of the values of font-size, inter-letter, inter-word and inter-column relative to the x-height, which is assumed to be available for the most widely used font types and sizes. Some other values are obtained by recommendation of guidelines for good typographic design. A study of the distortion caused by the camera projection and skew variance in those parameters is also presented.

## 1. Introduction

Most textual documents have elements with different font sizes and types, thus these features are not uniform. When documents are digitized by portable cameras, most of the time the planes of device and document are not parallel to each other. This fact increases the non-uniformity of the text size and type, turns the text-lines curled and with perspective distortion. In this case is more difficult to extract the font information when compared to (perfectly flat) scanned documents.

Most algorithms that process document images have parameters that are set globally for a predefined dot-per-inch. Thus, the algorithm may not work with documents of different resolutions and variable one as well.

The only way to circumvent this scenario is to make the algorithm "parameterizable" by local typographic features of the document. The feature exploited by this paper is the x-height, which are extracted by the text-line segmentations [5] [6] [7] on documents acquired by portable digital cameras. The

x-height is imaginary lines between the median and baseline as presented in Figure 1.

One important typographic measure is the 1 em value. It is relative to the font size; thus 1em for 12pt size font is 12pt. Several values for spacing between letters, words and columns recommended are relative to 1em [2].



Figure 1. Font typographic lines following Naylor [10]

The value of 1em is not available for a given text-region. In the other hand, the value of x-height relative to 1em plays a crucial aspect for OFT (Optical Font Recognition) [9] and Text-line segmentations [5] [6].

This paper shows studies regarding various aspects of character shape. The first part presents bounds for 1em, font limits and spacing constraints relative to the x-height. Having pre-estimated bonds such as the ones presented here casts light and systematizes parameter estimation, allowing much better performance for those algorithms that makes use of font and line parameters. The second part shows the effect of the camera projection on the character shapes followed by an analysis on the frequency of neighbor characters according to their shape. Finally, skew distortion effects over the character dimensions are also studied.

## 2. X-height relations

The relative sizes of typographic lines are measured here by placing the different character types in Microsoft Word 2007 with 80pt font size and applying a gray highlight on them. The height of the highlight is considered as 1em.

The choice of fonts considered two aspects: font sets either used by most printed documents or sets which widely differ from the parameters of the most commonly used ones. For the first choice was based



ranging from old to new books: Adam Smith – An Inquiry into the Nature and Causes of the Wealth of Nations; Jane Austen – Pride and Prejudice; Jules Verne – 20,000 Leagues Under the Sea; Jules Verne – Around the World in 80 Days; Karl Marx – Capital Volume 1; Jean-Jacques Rousseau – The social contract and discourses; Sigmund Freud – The Interpretation of Dreams; The Code of Hammurabi; Thomas Hobbes – Leviathan; UNESCO – TEN KEYS to the Convention on the Protection and Promotion of the Diversity of Cultural Expressions; Victor Hugo – Les Miserables; William Shakespeare – A Midsummer Night's Dream; William Shakespeare – Hamlet, Prince of Denmark (Collins edition); William Shakespeare – Julius Caesar (Hudson edition); Winston Churchill – A Traveller in War-Time.

Table 4 shows at each cell the percentage of the combination of two characters with the Table 3 classification; the first and second neighbor characters' shape are located at the corresponding first column and first row, respectively.

**Table 4.** Characters neighbor statistics

$1^{st}$ \ $2^{nd}$	Asc.	Desc.	Neutral	EM	Lower	Upper	Space
Asc.	5.39%	0.73%	13.19%	0.02%	0.73%	0.03%	4.20%
Desc.	0.57%	0.12%	2.37%	0.00%	0.26%	0.01%	1.04%
Neu.	11.71%	2.69%	29.49%	0.02%	1.35%	0.07%	7.11%
EM	0.04%	0.00%	0.01%	0.00%	0.01%	0.00%	0.02%
Lower	0.41%	0.02%	0.23%	0.02%	0.12%	0.11%	1.55%
Upper	0.17%	0.00%	0.08%	0.01%	0.00%	0.03%	0.07%
Space	6.00%	0.84%	7.26%	0.02%	0.01%	0.07%	0.60%

The cryptography text book [3] informs the letter frequency in English, although it does not distinguish the case of letters and numbers are not used in the statistics. Therefore, considering the statistics of [3] is valid for lower case, it is summarized in Table 5 first row. The second row shows the results among just the letters. The increase in the ascender is perceivable due the inclusion of upper case and numbers in this group. Thus, the information provided in Table 5 shows that Table 4 is coherent with previous study.

**Table 5.** Comparison between letter frequency statistics [3] and ascender, descender and neutral

Source	Ascender	Descender	Neutral
Reference [3]	27.92%	6.17%	65.91%
This paper	29.95%	5.40%	64.65%

#### 4. Projection distortion analysis

This section provides analysis of the characters proportion deviation when the document is obtained

by cameras. The distance between the paper sheet and capturing device is greater by an order of magnitude when compared to a single character in the text. For instance, 1em of a 12pt font size has 12/72 inches (0.43 cm) [2] while the normal focus range for a Fujifilm X10 is 50 cm [12], i.e. more than 100 times. Therefore, the projection of each character into the image can be modeled as orthographic [11].

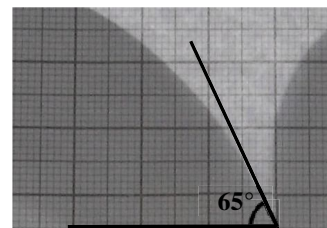
In this case, the typographic lines remain parallel under small distances. Thus, the ratio between the lengths of two line segments in the same direction is preserved. Previous section showed how the x-height is related to distances that are orthogonal to it (e.g. inter-column spacing, inter-letter etc). These relations are affected by the camera perspective distortion.

The character is modeled as a bounding box; the projection is the rotation of it over the x and y axes. Equation (1) shows the rotation operation of a point in (x, y, 0) relative to the x and y axes. Let a line segment in z = 0 represented by the points P<sub>1</sub> and P<sub>2</sub> be rotated over these axes, the ratio of the dimensions of the projected segmented can be expressed by equation (2). This equation shows that the ratio error is equal to 1 minus the division of the cosines, thus there is no error if the segment is rotated over the both axes by the same angle. Therefore, the distortion is maximized if one of the angles equals to 0. Figure 5 shows how the highest angle can be estimated. In this case the value is 65°, which gives a ratio error of 57.7%.

$$\begin{pmatrix} \cos \theta_Y & 0 & \sin \theta_Y \\ 0 & 1 & 0 \\ -\sin \theta_Y & 0 & \cos \theta_Y \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_X & -\sin \theta_X \\ 0 & \sin \theta_X & \cos \theta_X \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 0 \end{pmatrix} \quad (1)$$

$$= \begin{pmatrix} x \cos \theta_Y \\ y \cos \theta_X \\ \sin \theta_X - \sin \theta_Y \end{pmatrix}$$

$$\frac{\Delta x}{\Delta y} \text{ ratio error} = \frac{x_1 \cos \theta_Y - x_0 \cos \theta_Y}{y_1 \cos \theta_X - y_0 \cos \theta_X} = \frac{\Delta x}{\Delta y} \times \frac{\cos \theta_Y}{\cos \theta_X} \quad (2)$$



**Figure 4.** Example of a book binding and its highest slope angle

#### 5. Skew distortion analysis

This section study letter dimensions for different typesets at various skew angles. Each letter was painted on a blank image with 70pt using ImageJ [15] and rotated by 15°, 30° and 45° using Arial, Courier and Times font faces.

Table 6 presents some statistics of the connected component dimensions of the characters for plain, italic, bold and italic bold styles, respectively. Where  $R_w$  and  $R_h$  are as in equations (3) and (4), respectively; it measures the highest and lowest dimensions of different fonts;  $\bar{w}$  and  $\bar{h}$  are mean of the width and the height measured in pixels, respectively;  $\sigma_w$  and  $\sigma_h$  are width and height standard deviation percentages relative to  $\bar{w}$  and  $\bar{h}$ , respectively.

$$R_w = \frac{\min width}{\max width} \quad (3) \quad R_h = \frac{\min height}{\max height} \quad (4)$$

Table 6 shows that the height of fonts varies less than their width as:  $\sigma_h < \sigma_w$  and  $R_h > R_w$  in almost all cases (except for 45°). Courier fonts present less dimension dissimilarity among the three fonts studied as its standard deviation is the minimum when compared to others in all angles; probably due to the fact that Courier being a font with fixed width. On the other hand, Times font makes the greatest disparity to be observed, with the lowest  $R_w$  and the highest deviation of the font sets analyzed.

## Conclusions

This paper presents a statistical analysis of the typographical features of the font sets of most widespread use in electronic printing today. It allows establishing bounds for font-size, inter-letter, inter-word and inter-column relative to the x-height. A study of dimension variance of font sets and the distortion observed by the camera projection were also developed. Such analysis is fundamental in setting the parameters of document processing algorithms from skew and orientation analysis to textline segmentation. The data provided here was used to tune the values of parameters two text-line segmentation methods described in references [5] and [6], and can improve the performance of other applications such as OFT and OCR.

## Acknowledgements

The research reported herein was partly sponsored by CNPq- Brazilian Government.

## References

[1] N. French. InDesign Type: Professional Typography with Adobe InDesign CS2. pp. 219, Adobe Press, March, 2006.  
 [2] R. Bringhurst. Elementos do estilo tipográfico (versão 3.0). São Paulo, Cosac Naify, 2005.  
 [3] Robert Lewand. Cryptological Mathematics. The Mathematical Association of America, 2000. ISBN 978-0883857199.

[4] F. Shafait and T. M. Breuel. "Document Image Dewarping Contest", 2<sup>nd</sup> Int. Workshop on Camera-Based Document Analysis and Recognition, CBDAR 2007, Brazil. Sep. 2007. pp 181-188.  
 [5] D.M.Oliveira, R.D.Lins, G. Torreão, J. Fan and M. Thielo. A New Method for Text-line Segmentation for Warped Documents. Proceedings of ICIAR 2010. LNCS 6112, pp. 398-408, Springer Verlag, 2010.  
 [6] D.M.Oliveira, R.D.Lins, G. Torreão, J. Fan and M. Thielo. A New Algorithm for Segmenting Warped Text-lines in Document Images. Proceedings of the ACM SAC 2011, pp. 259-265, Tunghai, Taiwan, ACM Press 2011.  
 [7] S. S. Bukhari, F. Shafait, and T. M. Breuel. Coupled snakelet model for curled textline segmentation of camera-captured document images. In Proceedings 10th International Conference on Document Analysis and Recognition, pages 61-65, Barcelona, Spain, 2009.  
 [8] F. Bapst and R. Ingold. Using Typography in Document Image Analysis. RIDT'98: Fourth Int'l Conf. Raster Imaging and Digital Typography, San Malo, France, Apr. 1998.  
 [9] A. Zramdini and R. Ingold. Optical font recognition using typographical features. IEEE Trans. PAMI, 20(8):877-882, 1998.  
 [10] J.J. Weinman. Typographical Features for Scene Text Recognition. Proceedings of ICPR'10, pp.3987-3990, IEEE Press.  
 [11] I. Carlbom and J. Paciorek. Planar Geometric Projections and Viewing Transformations. ACM Computing Surveys 10 (4): 465-502, 1978.  
 [12] Dpreview. Fujifilm X10: Digital Photography Review. [http://www.dpreview.com/products/fujifilm/compacts/fujifilm\\_x10](http://www.dpreview.com/products/fujifilm/compacts/fujifilm_x10)  
 [13] M. Naylor. Typographic line terms. Available at [http://en.wikipedia.org/wiki/File:Typography\\_Line\\_Terms.svg](http://en.wikipedia.org/wiki/File:Typography_Line_Terms.svg)  
 [14] Domínio público. <http://www.dominiopublico.gov.br/>.  
 [15] ImageJ. <http://rsb.info.nih.gov/ij/>.  
 [16] Apple Inc. Fine typography: Numbers. <http://developer.apple.com/fonts/tools/toolkit/TrueEdit/Documentation/TE/TE3numbers.html>.

**Table 6.** Analysis of the variation of font width and height

	$\theta$	$R_w$	$R_h$	$\bar{w}$	$\bar{h}$	$\sigma_w$	$\sigma_h$
Arial	0°	0.0725	0.5538	32.33	48.46	37.56%	14.65%
	15°	0.2174	0.5758	39.46	52.40	26.89%	13.67%
	30°	0.3243	0.4861	45.41	52.36	21.57%	16.03%
	45°	0.4079	0.3816	50.30	49.71	17.62%	20.20%
Courier	0°	0.0714	0.5283	30.67	40.33	25.68%	15.31%
	15°	0.2400	0.5472	35.91	44.69	20.27%	12.55%
	30°	0.3393	0.4912	41.11	45.64	17.50%	13.28%
	45°	0.4211	0.3966	44.33	44.09	15.22%	16.03%
Times	0°	0.0469	0.5079	33.04	45.91	37.83%	17.13%
	15°	0.1918	0.5224	39.07	49.87	29.43%	15.95%
	30°	0.2692	0.4722	44.81	50.03	23.69%	18.83%
	45°	0.3766	0.3377	48.96	47.69	19.68%	23.17%

R. D. Lins, D. M. Oliveira, G. Torreão, J. Fan e M. Thielo. *A Dewarping Algorithm to Compensate Volume Binding Distortion in Scanned Documents*. ACM SAC 2010, 2010.

# A DeWarping Algorithm to Compensate Volume Binding Distortion in Scanned Documents

Rafael D. Lins  
Daniel M. Oliveira  
Gabriel Torreão  
UFPE, Recife, Brazil  
{rdl, daniel.moliveira}@ufpe.br

Jian Fan  
HP Labs.,  
Palo Alto, USA  
jian.fan@hp.com

Marcelo Thielo  
HP Brazil R&D,  
Porto Alegre, Brazil  
marcelo.resende.thielo@hp.com

## ABSTRACT

This paper presents a new algorithm for dewarping the volume binding distortion often found whenever one scans a book. The existing algorithms in the literature need scanner parameters, seldom known by the end user. The algorithm proposed here only needs the user to print a calibration grid and to scan it in the central page of a volume. The algorithm collects all the data needed to dewarp all pages in the volume.

## Categories and Subject Descriptors

I.4.9 [Image Processing and Computer Vision]: Applications.

## General Terms

Algorithms, Experimentation.

## Keywords

Dewarping, document processing, image processing, scanners.

## 1. INTRODUCTION

Digitalization by scanners can yield an unpleasant volume binding warp such as in Figure 2.a. The readability and OCR transcription accuracy rate are increased when such distortion is removed.

Tan *et al* propose in [1] a dewarping algorithm for scanned documents by estimating page shape using shading pattern and several scanner parameters (focus, photoelectric gain/bias, etc.), most of these are not available to end users. Ukida *et al* [2] restore warp binding distortions assuming that the scanner has three or four light sources (red, blue and green), where commercial scanners often have single white light. The survey paper [3] shows methods where shading is used to estimate object shapes, these are too time consuming and yield poor results [4]. The shape is inferred using stereo vision in [7] by scanner customization, but often this is not feasible.

The dewarping algorithms used for camera-captured documents [6] cannot be used to fix the distortion such as presented in Figure 1 due to the projection model which is orthographic in the direction of the lamp movement  $y$  (i.e. lack of line curl on Figure 1) and with perspective in the  $x$  direction [8].

The method proposed here, as in [1], also uses the shading pattern, but it does not depend on parameters provided by the of the scanner. A calibration grid image is scanned only once to

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC'10, March 22-26, 2010, Sierre, Switzerland.

Copyright 2010 ACM 978-1-60558-638-0/10/03...\$10.00.

gather shape-from-shading relations, which is used to correct the distortion found in all pages in the volume.

## 2. THE PROPOSED METHOD

Figure 1 shows that paper shading turns darker as gets closer to the book spine (bound area) which does not lie on the scanner flatbed. The relation between shading and paper depth can be found by scanning the undistorted calibration grid that is placed in middle of the book volume, which stands is the worst distortion case (see Figure 1.a). Often the scanning process produces some skewed pages, thus PhotoDoc [11] perspective correction can be used in these prior to dewarping. Assuming that the calibration grid is scanned in grayscale, to infer shading/depth relation one does:

1. Find the grid corners using Harris Corner detector [9].
2. Measure the modal value of the distance between two grid corners (which corresponds to undistorted square side).
3. Calculate the height increments as in Figure 1.b, then measure total height.
4. Find the whitish value around corner, which is set as the shading value.
5. For the same column, one has the shading ( $I$ ) and height ( $h$ ) values, thus find a function  $h(I)$  mapping them.

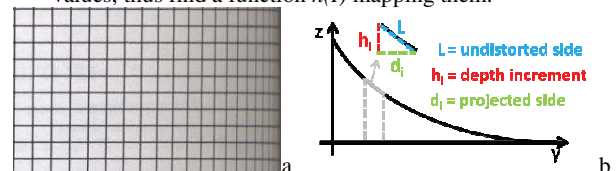


Figure 1. Calibration: distorted grid (a); Obtaining depth (b)

Given any page image with book binding warp the following steps should be followed to de-warp it:

1. For each column  $i$  of the image find the mode of the intensity of the pixels in the highest 10-percentile in the histogram of the pixels in the column. This value  $b_i$  is taken as the representative of the background (paper) intensity in that column.
2. For each  $b_i$  found in the step above one uses the curve  $h(I)$  from the calibration grid to find the corresponding height  $h(b_i)$ .
3. Split the image in 8-pixel wide columns blocks. For each of them make a linear regression of the  $h(b_i)$  in the block.
4. For each block of 8-pixel wide columns, using Pythagoras theorem in the triangle shown in Figure 1.b with  $d_i=8$  and  $h_i=Y(b_i)-Y(b_{i+8})$ , where  $Y(x)$  is the linear regression of the block, one may calculate the hypotenuse which is rounded-off to an integer and this value corresponds to the width of the block in the final de-warped image.



- The final block image is obtained by re-sizing the columns to the width calculated in the previous step by bi-linear interpolation.

Once the final width is calculated for each block, the dewarped image can be obtained. Figure 2 shows processing results together with shading removal methods [5] (c) and [10] (d). For the ten warped images tested, seven where adjusted correctly, local noises can yield to poor results (e.g. Figure 3).

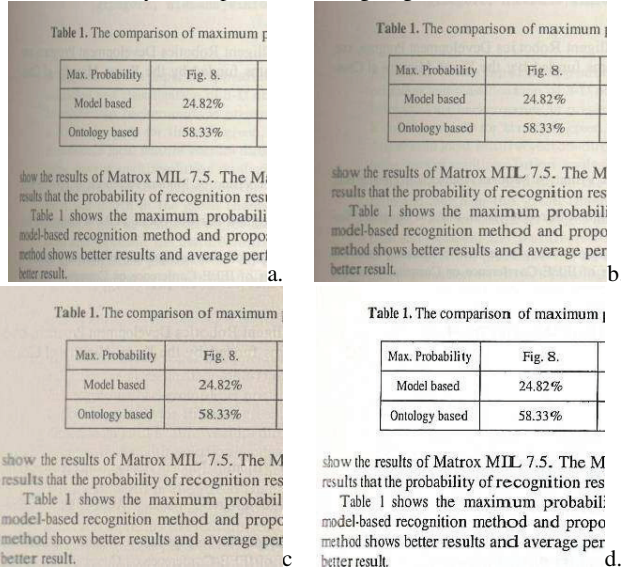


Figure 2. Processing result: original image (a); dewarped (b); Oliveira-Lins [5] (c); Fan [10] (d)

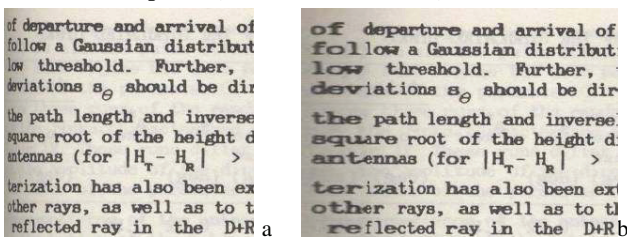


Figure 3. Unsatisfactory result: original (a); dewarped (b)

To quantitatively validate the performance of the proposed dewarping a sheet with repeated letters served as ground-truth. It was inserted in a bound book and was scanned. After dewarping each character width was compared. It was verified that font sizes are completely recovered on the book binding until 2.5 cm of the margin, which covers most part of text in the deformation area. Between 2.5 cm and 1.5 cm, the recovery is not perfect, but the width of the font increased. From 1.5 cm to the spine most information is lost. Figure 4 shows the results obtained.

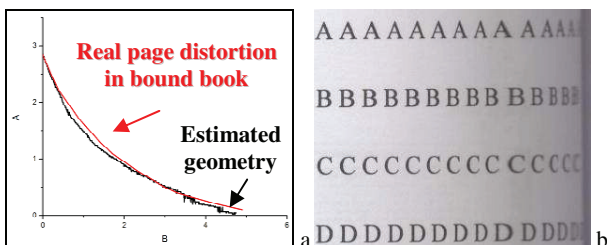


Figure 4. Validation: ground truth (a); letter sizes (b)

### 3. CONCLUSIONS

This paper presented a new method for dewarping book bound scanned images. The user is asked to scan a calibration grid inserted in the middle of volume to be dewarped. This is enough to get shape-from-shading information to reconstruct the book surface and does not need any scanner parameter provided by the scanner manufacturer. The algorithm worked satisfactorily in most of the real images tested.

### 4. REFERENCES

- C.L. Tan, L. Zhang, Z. Zhang, and T. Xia, Restoring Warped Document Images through 3D Shape Modeling, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 28, no. 2, pp. 195-208, Feb. 2006.
- H.Ukida, Y.Tanimoto, T.Sano, H.Yamamoto, 3D Shape Reconstructions Using Image Scanner under Various Number of Illuminations. IEEE International Workshop on Imaging Systems and Techniques, 2007. IST '07. May 5, 2007, Krakow, Poland, 2007, pp. 1-6.
- J.-D.Durou, M.Falcone, M.Sagona. Numerical Methods for Shape-from-shading: A New Survey with Benchmarks. Computer Vision and Image Understanding, 2007.
- R.Szeliski. Computer Vision: Algorithms and Applications. Available at <http://research.microsoft.com/en-us/um/people/szeliski/Book/>. (to be published).
- D.M.Oliveira and R.D.Lins. A New Method for Shading Removal and Binarization of Documents Acquired with Portable Digital Cameras. Proceedings of International Workshop on Camera-Based Document Analysis and Recognition, 2009, Barcelona. pp. 3-10.
- S.S.Bukhari, F.Shafait, T.M. Breuel. Dewarping of Document Images using Coupled-Snakes. Proceedings of International Workshop on Camera-Based Document Analysis and Recognition, 2009, Barcelona. pp. 3-10.
- R.Pintus, T.Malzbenber, O.Wang, R.Bergman, H.Nachlieli, and G. Ruckenstein. Photo Repair and 3D Structure from Flatbed Scanners. International Conference on Computer Vision Theory and Applications, Lisbon, Portugal, pp. 40-50, 2009.
- R.Schubert. Using a Flatbed Scanner as a Stereoscopic Near-Field Camera, IEEE Computer Graphics and Applications, 2000, pp. 38-45.
- W.Burger and M.J.Burge. Digital Image Processing: An Algorithmic Introduction using Java. ISBN: 978-1-84628-379-6. Springer, New York, 2008.
- J.Fan. Robust Color Image Enhancement of Digitized Books, Proceedings of International Conference on Document Analysis and Recognition, pp 561-565, IEEE Press, 2009.
- G.F.P. Silva, R.D. Lins. PhotoDoc: A Toolbox for Processing Document Images Acquired Using Portable Digital Cameras. Proceedings of International Workshop on Camera-Based Document Analysis and Recognition, 2007, Curitiba. pp. 107-115.

R. D. Lins, D. M. Oliveira, G. Torreão, J. Fan e M. Thielo. *Correcting Book Binding Distortion in Scanned Documents*. Proceedings of ICIAR 2010. LNCS 6112, pp. 355-365, Springer Verlag, 2010.

# Correcting Book Binding Distortion in Scanned Documents

Rafael Dueire Lins<sup>1</sup>, Daniel M. Oliveira<sup>1</sup>, Gabriel Torreão<sup>1</sup>,  
Jian Fan<sup>2</sup>, and Marcelo Thielo<sup>3</sup>

<sup>1</sup> Universidade Federal de Pernambuco, Departamento de Eletrônica e Sistemas,  
Av. Prof. Luiz Freire, s/n, Cidade Universitária  
50740-540, Recife, PE, Brasil

{rdl,daniel.moliveira,gabriel.dsilva}@ufpe.br

<sup>2</sup> Hewlett-Packard Labs, Palo Alto, USA

jian.fan@hp.com

<sup>3</sup> Hewlett-Packard Labs, Porto Alegre, Brazil

marcelo.resende.thielo@hp.com

**Abstract.** This paper presents an exponential model to correct the distortion that appears when bound books are digitalized using flatbed scanners. The proposed algorithm infers the height of the book spine for every column in the image and fits an exponential curve. The de-warping is done regarding the scanner acquisition model and shape binding. This new de-warping method is effective, fast, accurate and easy to implement.

**Keywords:** Shape-from-shading, image processing, book binding de-warping.

## 1 Introduction

Scanners are the device most widely used for document digitalization. Figure 1 (a) shows only one page scanned at a time, in which the sweeping line is perpendicular to the book spine. Figure 1(b) depicts the scanning of two pages a time, with the book spine lying parallel to the scanning line. Examples of images obtained in such scenarios can be seen in Figures 1(c) and 1(e). In both situations, an unpleasant warp appears near the book binding, which is better visualized in the enlarged parts of Figures 1.d and 1.f, respectively. It is important to notice that the two scenarios yield very different images: the scanner projection is orthographic in the direction of the lamp movement, but has perspective projection in the direction orthogonal to the lamp movement [1][11].

This paper proposes a new method to correct both scenarios presented in Fig.1 assuming that the document image has only one page. It is organized as follows: Section 1.1 shows a brief review of the de-warping literature and sketches the proposed method. Section 2 presents the scanner acquisition model and its parameters extraction. Section 3 explains new distortion correction methods whose results are provided in Section 4. Conclusions and lines for further work are in Section 5.

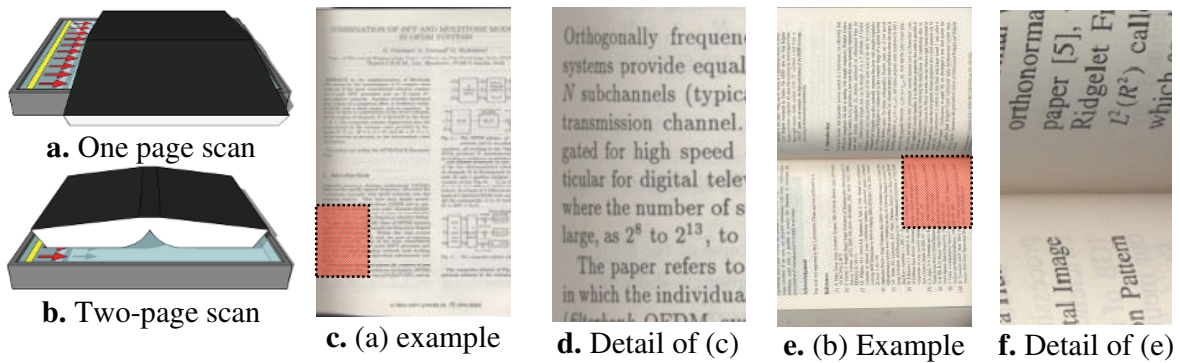


Fig. 1. Book scanning scenarios

## 1.1 Review of the De-warping Literature

Tan *et al* propose in [1] a de-warping algorithm for scanned document images that reconstructs the volume surface using a shape-from-shading pattern in the scenario of Fig. 1.b. The main disadvantage of this method is the need for several scanner parameters (focus, gain and bias of the photoelectric transformation, etc). Scanner manufacturers seldom provide those parameters, making impossible the application of the algorithm. Ukida and Konishi [2] introduce a warp restoration in scanned images, using the scanner with three different light sources (red, blue and green) at fixed locations. Ukida *et al* [3] propose a shape and color reconstruction method using a scanner equipped with four light sources which are located symmetrically under the scanning line. Most scanners have a single white light source, thus the application of the algorithms in references [2] and [3] may not be possible.

Reference [15] introduces a new method that only asks the user to scan a calibration grid placed at the center of the volume to be scanned. This grid allows inferring the relation between pixel intensity and depth. Such parameters are used to correct the distortion in the other pages of the volume. This method is efficient in a large number of book images, but in a few pages tested the noisy areas had their shading wrongly estimated causing excessive widening of the de-warped area.

There are several works which focus on estimating the 3D-model using the shading pattern of photo objects. Two surveys can be found in references [4] and [5]. Most of those solutions are time intensive, besides not yielding good de-warping results [6].

Another approach, recently proposed by Pintus *et al* in [10] modifies the scanner to gather two images at the same time to build a 3D model by stereo vision 3D reconstruction. This method claims for a customized scanner.

There are also several other methods for removing such distortion for documents captured with portable digital cameras. Most algorithms try to find document text-lines [8][9], assuming that they are parallel in the original document. Those algorithms get the envelope of each line and attempt to “straighten” them by moving font cases, not correcting the narrowing of the fonts in the distorted region. Such algorithms are unsuitable for scanned documents, because as shown in Fig. 1.c, images from scanned books often present binding distortion but lines remain parallel to each other. This happens due to the orthographic/perspective projection of scanners as explained earlier on, therefore most portable camera de-warping algorithms in the literature cannot be used.

The algorithm proposed herein borrows some ideas of [15], by using the document shading pattern to estimate its depth using the relation obtained in section 2. The depth is fitted onto an exponential curve as shown in section 3.

## 2 The Scanner Acquisition Model

To validate this scanner projection model, a square grid was placed on a flat slope [2] as illustrated in Fig. 2.a. Using HP ScanJet 5300c scanner with 300dpi, Fig. 2.b was acquired with  $\Psi=90^\circ$  and  $\theta=30^\circ$  (1.c scenario); Fig. 2.d with  $\Psi=0^\circ$  and  $\theta=30^\circ$  (1.e scenario). The height of the squares is constant, independently of the paper depth (orthographic projection), whereas the square “widths” are narrower and closer to the focal axis. The paper becomes darker with increasing distance, as shown in Fig. 2.c.

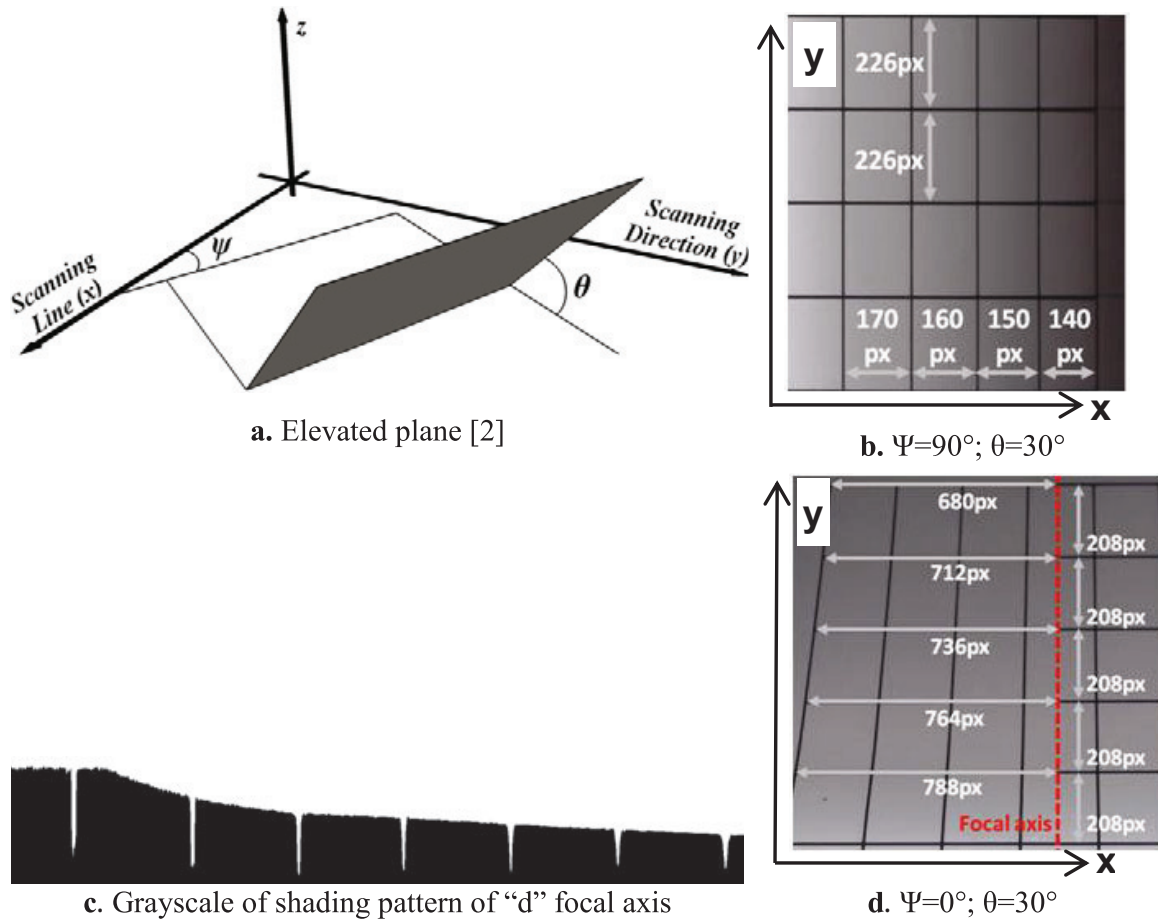


Fig. 2. Square grid for validating the projection model

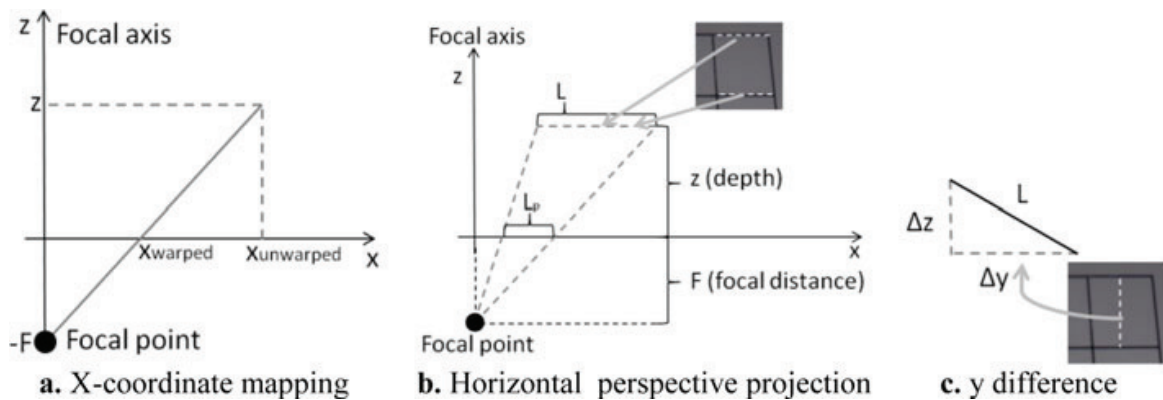


Fig. 3. Warped/unwarped (flat) projection relations

From Fig. 3.a it is possible to infer warped/flat  $x$  values relation in equation (1) with triangle relations, where  $F_a$  is the focal axis  $x$ -coordinate. Fig. 3 also shows Fig. 2.d distorted square. Using triangular relations, one gets eq. (2). Let  $L_{p1}$  and  $L_{p2}$  be the values of the upper and lower projected square sides. The difference between them yields to (3). The value for  $\Delta z$  is calculated using Pythagoras theorem (Fig. 3.b). Finally, the focal distance can be obtained with (4). As there is more than one square available, the focal distance is calculated for each square and the median of the values obtained is chosen for use.

$$x_{warped} = F_a + F \times (x_{unwarped} - F_a)/(z(y) + F) . \quad (1)$$

$$\frac{L}{L_p} = \frac{z+F}{F} . \quad (2)$$

$$L \times \left( \frac{1}{L_{p1}} - \frac{1}{L_{p2}} \right) = \frac{\Delta z}{F} ; \text{ where } \Delta z = z_1 - z_2 . \quad (3)$$

$$F = \frac{1}{L} \times \sqrt{L^2 - \Delta y^2} \times \left| \frac{1}{L_{p1}} - \frac{1}{L_{p2}} \right|^{-1} . \quad (4)$$

The relation between the paper depth and shading is assumed to be linear (see Fig. 2.c), where zero depth means that the paper is on the scanner flatbed (region without distortion). With the focal distance, it is possible to infer the distance to the flatbed ( $z$ -value) using equation (2).

- Let  $I_0$  and  $I_{flat}$  be the square grid whitest paper shading values in the region around the corner.  $I_0$  is measured from the warped horizontal side,  $I_{flat}$  is measured from the flat region.
- Let  $b$  and  $b_{flat}$  be as above, but measured from the warped document, respectively.

Thus, the shading/depth relation is expressed in (5). One may observe that for  $b = b_{flat}$  the depth equals 0, that means that it is on the flatbed. For the HP ScanJet 5300c, the values found for 300dpi were: 2,894.37 pixels (24.51cm using eq. (6)) for  $F$ , 233/255 for  $I_{flat}$ , 174/255 for  $I_0$  and 270.1412 pixels for  $Z_0$ .

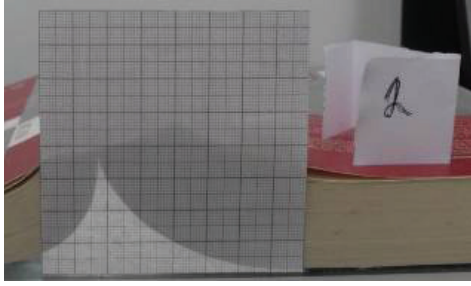
$$d(b) = Z_0 \times (1 - b/b_{flat})/(1 - I_0/I_{flat}) . \quad (5)$$

$$\text{pixelsToCm}(v) = (v/\text{DPI}) \times 2.54 . \quad (6)$$

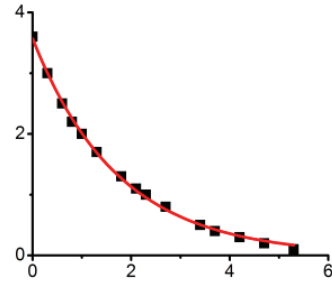
### 3 The De-warping Procedure

To better understand the shape of the book binding distortion, 8 pictures were taken with different warp levels (Fig. 4.a shows one of them). Sample points were fitted into an exponential curve as described by eq. (7) (Fig. 4.b shows plotting for 4.a) with a mean error of less than 0.08 cm for all 8 images. Thus, eq. (7) models binding with acceptable accuracy.

$$z(x) = Z_0 \times e^{\lambda x} . \quad (7)$$



a. Distortion of a page in middle of book



b. Plot and exponential fitting of (a)

**Fig. 4.** Validation of book binding modeled as an exponential curve

To correct the book binding distortion, the depth is estimated and fitted onto the exponential curve represented by equation (7). One simple and effective method is presented in [14] using equations (8) and (9). De-warping is done by the line integral of (7) which is presented by equations (10) and (11). The next sections show how to obtain the value of the depth for both scenarios in Fig. 1.

$$Z_0 = e^{D_0}; D_0 = \frac{\sum_{i=1}^n \ln z_i \sum_{i=1}^n x_i^2 - \sum_{i=1}^n x_i \sum_{i=1}^n x_i \ln z_i}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \tag{8}$$

$$\lambda = \frac{n \sum_{i=1}^n x_i \ln z_i - \sum_{i=1}^n x_i \sum_{i=1}^n \ln z_i}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \tag{9}$$

$$s(x) = \frac{1}{\lambda} \left( \sqrt{(\lambda z(x))^2 + 1} - \ln \left| \frac{\sqrt{(\lambda z(x))^2 + 1} + \lambda z(x)}{\lambda z(x)} \right| \right) \tag{10}$$

$$x_d(x) = s(x) - s(0) \tag{11}$$

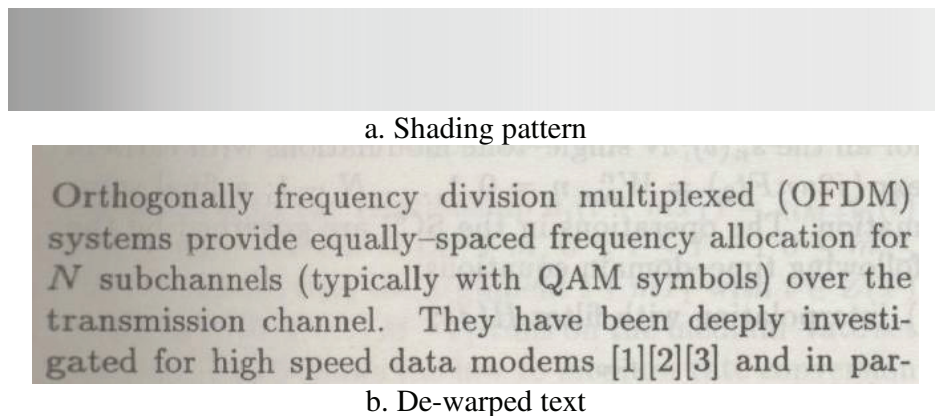
### 3.1 Shape-from-Shading

Assuming that an image was obtained in the scenario depicted in Fig. 1.a, and the projection to be pure orthographic due to small flatbed distances, the proposed procedure for de-warping is summarized as:

1. For each column  $i$  of the image find the mode of the intensity of the pixels in the highest 10-percentile in the histogram of the pixels in the column. This value  $b_i$  is taken as representative of the intensity of the paper column.
2. Identify the shading value of a flat region ( $b_{flat}$ ) by getting the most frequent value of  $b_i$ .
3. The exponential fitting is done to obtain  $I$  and  $\lambda$  with  $(i, z_i)$  (where  $z_i = d(b_i)$ ) using equations (8) and (9) [14]. The values for  $z_i$  that are too close to 0 have a greater influence than others as the logarithm goes to minus infinite as the input approaches zero [14]. The curve fitting is done with  $|b_i - b_{flat}| \geq 20/255$ .
4. Apply the de-warping procedure using the line integral of  $z(x)$  as described in equations (10) and (11).

The inverse function for equation (10) cannot be calculated analytically. The direct mapping is done (from source  $x$  to target  $x$ ) with values for the orphan  $x$  columns in the target, calculated by linear interpolation of the closest columns. Figure 5 shows the proposed procedure applied to Figure 1.c, with the shading pattern at (a) and the detail of the final result in (b). The advantages of proposed method are:

- The fitting procedure detects the deformation side (left or right) automatically as  $\lambda$  can be positive or negative
- There is no need to detect when the warped region starts. The exponential has low derivative values on the opposite side of the deformation, thus it acts as a constant straight line in this area. Thus equation (7) can be applied to every value of  $x$ .



**Fig. 5.** Proposed shape-from-shading applied to Figure 1.c

### 3.2 Text-Line Depth Extraction

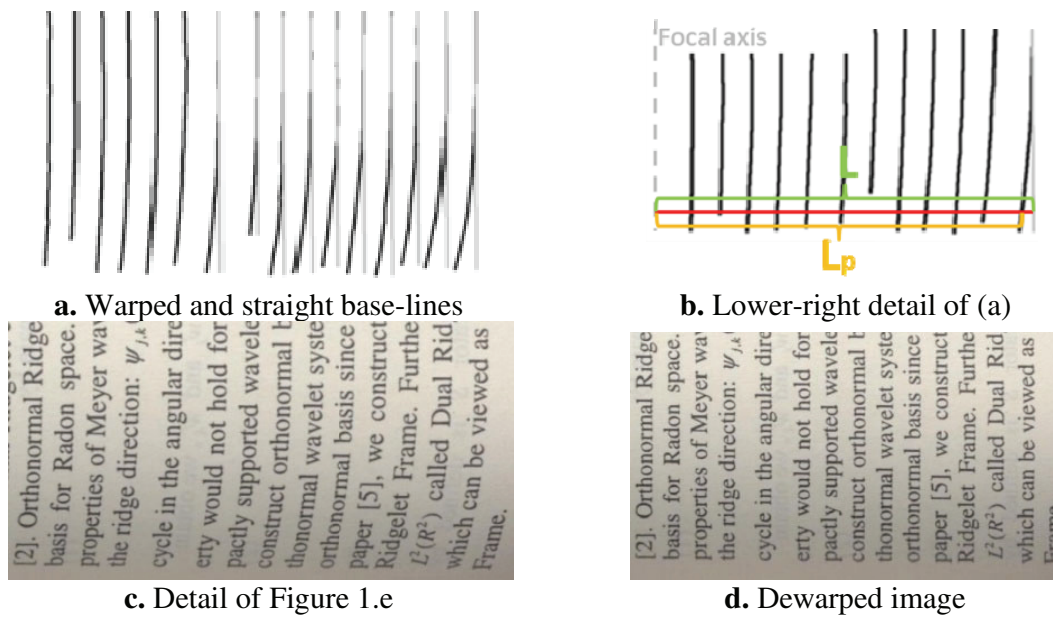
Looking at the central part of the book image in Figure 1.e one may observe that closer to the book spine the top page looks brighter than the bottom page. Thus, at the same depth one finds two different luminance values, therefore shape-from-shading cannot be used to estimate the distance to the scanner flatbed in the scenario of Figure 1.b. As one may observe, the lines are curved in this case, thus text-line segmentation can be used to get depth values. There are several text-line segmentation for black-and-white warped documents such as [8][9]. For the sake of space, the method used herein is not presented. As segmentation requires a monochromatic image as input, image binarization [7] is performed. One assumes that the text-lines are already segmented and the baseline envelope for each of them is found. The following procedure is executed:

1. Un-warped straight lines are estimated by doing a linear regression with text-line central letters, as illustrated in Figures 6.a and 6.b in gray.
2. For each point of the baseline envelope there is a corresponding point on the straight line that represents where it should be if there were no warp. For the given point, depth ( $z$  value) can be obtained using eq. (2), where  $L$  and  $L_p$  are the distance to the focal axis for “un-warped” and warped points, respectively, with the focal axis assumed to be the vertical middle line (e.g. Fig. 6.b);



3. The exponential curve fitting is done with equations (8) and (9) [14] using  $(y_i, z_i)$  of baseline points. Not all baseline points can be used, thus they are restricted to:
  - a. As a point gets closer to the focal axis  $L$  and  $L_p$  gets smaller and more prone to errors, thus the curve fitting only uses points far away from the focal axis.
  - b. Higher depth values than a given threshold (see shape-from-shading step 3);
4. Use the line integral of  $z(y)$  as described in equations (10) and (11) to obtain the de-warped  $y$  value, and use equation (1) to obtain the warped  $x$ -value for every point.

In step 1, the central letters is in a region where  $y$  value is between 25% and 75% of the image height. In step 3.a, the baseline points far from the focal axis are the ones with  $x$ -coordinate less than 25% or greater than 75% of the image height. In step 3.b, the low depth values are the ones greater than 0.169 cm (or 20 pixels for 300 dpi using equation (6)). There is no analytical inverse for equation (10), thus orphan line points in the resulting image are calculated by bilinear interpolation of the closest points. Figure 6.d shows details of processing Figure 1.e. This method also has the same advantages as the previous one.



**Fig. 6.** Text-line depth extraction on Figure 1.e

## 4 Results

To validate the proposed method 20 different pages were scanned in the scenario of Figure 1.a and 15 in the scenario depicted in Figure 1.b. Some examples can be seen in Table 1 with the original and de-warped images on the left and the right hand sides, respectively. The first row shows a page with text lines with repeated letters in order to compare the result of the width restoration of Figure 1.a. The only limitation perceived is when binding is small and changes rapidly; the last row of Table 1 shows one of such cases. The next sections show pre- and post processing steps to improve the method proposed herein.

**Table 1.** Dewarping results: original (left) with dewarped (right)

<p>Fig. 10. ontology (left) vs. model-based</p> <p>Table 1. The comparison of maximum probab</p> <table border="1"> <thead> <tr> <th>Max. Probability</th> <th>Fig. 8.</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>Model based</td> <td>24.82%</td> <td>29</td> </tr> <tr> <td>Ontology based</td> <td>58.33%</td> <td>60</td> </tr> </tbody> </table> <p>show the results of Matrox MIL 7.5. The Matrox I results that the probability of recognition results ex</p> <p>Table 1 shows the maximum probability of model-based recognition method and proposed m</p> <p>method shows better results and average perform</p> <p>better result.</p>	Max. Probability	Fig. 8.	F	Model based	24.82%	29	Ontology based	58.33%	60	<p>Fig. 10. ontology (left) vs. model-based</p> <p>Table 1. The comparison of maximum probab</p> <table border="1"> <thead> <tr> <th>Max. Probability</th> <th>Fig. 8.</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>Model based</td> <td>24.82%</td> <td>29,</td> </tr> <tr> <td>Ontology based</td> <td>58.33%</td> <td>60,</td> </tr> </tbody> </table> <p>show the results of Matrox MIL 7.5. The Matrox I results that the probability of recognition results ex</p> <p>Table 1 shows the maximum probability of model-based recognition method and proposed m</p> <p>method shows better results and average perform</p> <p>better result.</p>	Max. Probability	Fig. 8.	F	Model based	24.82%	29,	Ontology based	58.33%	60,
Max. Probability	Fig. 8.	F																	
Model based	24.82%	29																	
Ontology based	58.33%	60																	
Max. Probability	Fig. 8.	F																	
Model based	24.82%	29,																	
Ontology based	58.33%	60,																	
<p>er of all pixels whose values</p> <p>erators to score the extent of</p> <p>ween 0 and 1, because</p> <p>6 denotes the product of all</p> <p>ument of the denominator is</p> <p>use - the most equalization in</p> <p>im of 0, add 1 to each factor,</p> $\log \left[ \frac{\prod_{k=0}^{L-1} n_k}{\sum_{k=0}^{L-1} n_k} \right]$ <p>of</p>	<p>er of all pixels whose values</p> <p>erators to score the extent of</p> <p>ween 0 and 1, because</p> <p>6 denotes the product of all</p> <p>ument of the denominator is</p> <p>use - the most equalization in</p> <p>im of 0, add 1 to each factor,</p> $\log \left[ \frac{\prod_{k=0}^{L-1} n_k}{\sum_{k=0}^{L-1} n_k} \right]$ <p>of</p> <p>a uniform intensity or color</p> <p>lication. From Table 5, it is</p>																		
<p>of departure and arrival of the direc</p> <p>follow a Gaussian distribution, above</p> <p>low threshold. Further, that the</p> <p>deviations <math>s_{\theta}</math> should be directly prop</p> <p>the path length and inversely proport</p> <p>square root of the height difference</p> <p>antennas (for <math> H_r - H_k  &gt; 60 \text{ m}</math>).</p> <p>terization has also been extended to</p> <p>other rays, as well as to the grazing</p> <p>reflected ray in the D+R mechanis</p> <p>portionality constant in the relation</p> <p><math>s_{\theta}</math>, the path length and the height di</p> <p>been adjusted to fit observations by</p> <p>The actual values of '<math>s_{\theta}</math>' used in the</p>	<p>of departure and arrival of the direc</p> <p>follow a Gaussian distribution, above</p> <p>low threshold. Further, that the</p> <p>deviations <math>s_{\theta}</math> should be directly prop</p> <p>the path length and inversely proport</p> <p>square root of the height difference</p> <p>antennas (for <math> H_r - H_k  &gt; 60 \text{ m}</math>). T</p> <p>terization has also been extended to</p> <p>other rays, as well as to the grazing</p> <p>reflected ray in the D+R mechanis</p> <p>portionality constant in the relation</p> <p><math>s_{\theta}</math>, the path length and the height di</p> <p>been adjusted to fit observations by</p> <p>The actual values of '<math>s_{\theta}</math>' used in the</p>																		

### 4.1 Pre-processing Improvements

Very often, despite the best efforts of those who scan bound volumes to avoid image skew, the obtained images may not be perfectly parallel to the axis of the scanner

flatbed. To correct such distortion, PhotoDoc [16] perspective correction routine is used. An example of such pre-processing is shown in Figure 8, with zoomed results in (d)-(f). The improvement of the new method can be seen in (e) if compared with method [15] in (f).

Text-line segmentation can be also used to detect portrait/landscape orientation assuming that horizontal lines are dominant. As the end-user does not know when to apply shape-from-shading or text-line depth extraction, the flow in Figure 7 is proposed. For all tested images it correctly identified all page orientations.

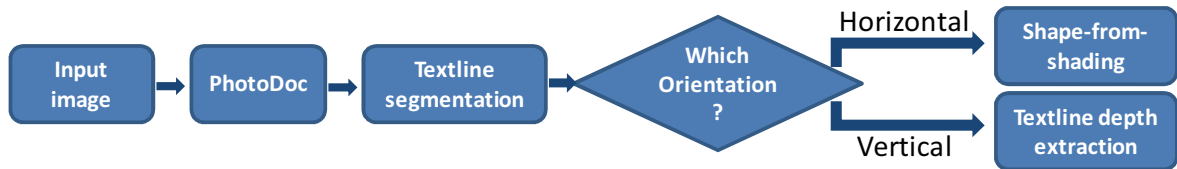
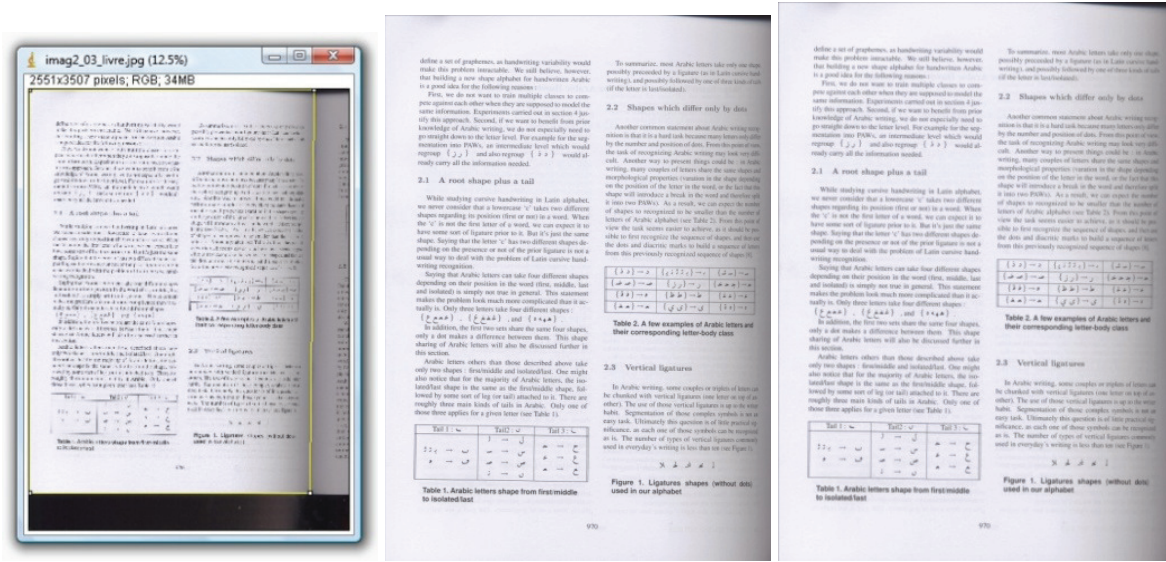
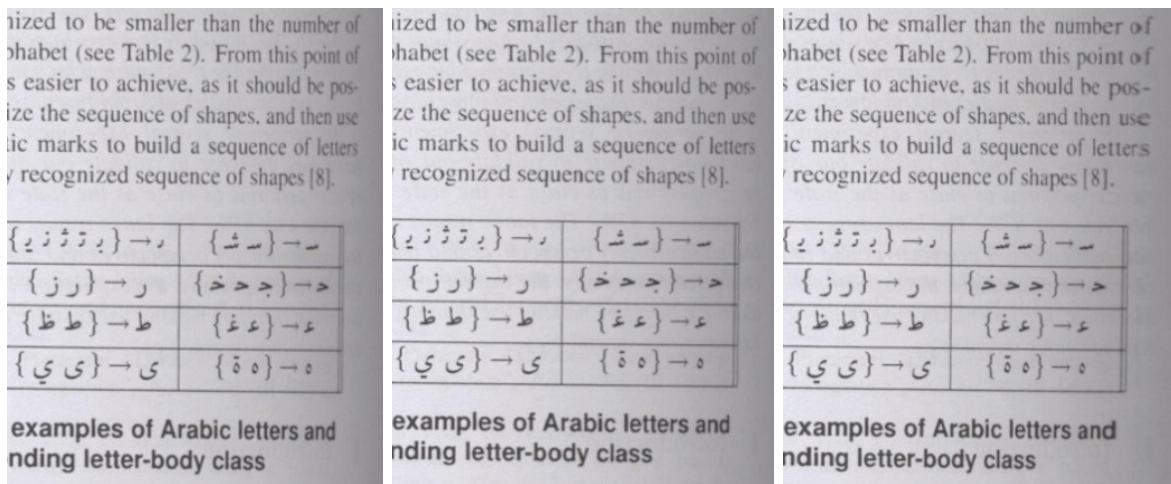


Fig. 7. End-users processing flow



a. Original image      b. Perspective corrected image      c. De-warped image

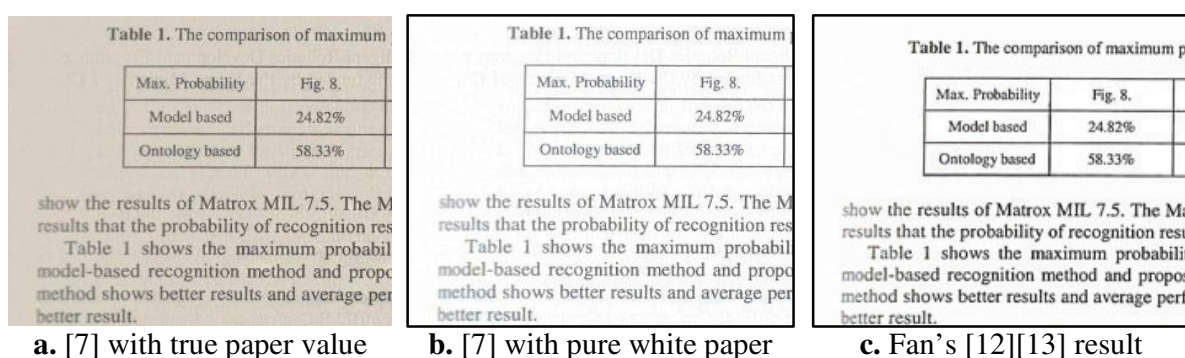


d. Zoom into (a) binding      e. New method on (d)      f. Method [15] on (d)

Fig. 8. Example of image with perspective and volume binding warp

## 4.2 Post-processing Improvements

The image de-warping algorithm presented herein does not correct the variation in illumination which darkens the region towards the volume binding of the page. Compensating illumination variation is part of the full de-warping procedure. Figure 9 shows the application of two [7] [12][13] shading removal algorithms. Oliveira and Lins [7] searches for blocks with low color variation then tries to find overlapping blocks that belongs to document paper followed by shading removal. Fan's method [12][13] uses a watershed-based color segmentation that works in a wide range of document contents such as those containing large continuous-tone image regions, with a high computational complexity. Figure 9.a shows the application of [7] with inferred paper value; Figure 9.b shows the result forcing the final background to pure white. Figure 9.c presents the result of applying the algorithms in references [12][13].



**Fig. 9.** Shading removal

## 5 Conclusions and Lines for Further Work

This paper presents a new algorithm for the correction of volume binding distortion in scanned books. First, the document orientation is automatically detected, and then proper depth estimation is chosen followed by exponential fitting and de-warping is performed by using an exponential line integral. For depth estimation, two new methods were proposed. The first one uses shape-from-shading, the other extracts the height by analyzing the envelope of text-lines. Both of them yielded satisfactory results. The application of shading removal algorithms is also recommended.

As some scanned documents with binding distortion may have blurred characters a document de-blurring strategy seems to be appropriate, but is left for further work.

## Acknowledgments

The authors like to thank Professor Liliana Gheorghe, of the Department of Mathematics of UFPE for checking the mathematical formalism.

The research reported herein was sponsored by an MCT-Brazilian Government R&D Grant and CNPq funding.

## References

- [1] Tan, C.L., Zhang, L., Zhang, Z., Xia, T.: Restoring Warped Document Images through 3D Shape Modeling. *IEEE Trans. Pattern Analysis and Machine Intelligence* 28(2), 195–208 (2006)
- [2] Ukida, H., Konishi, K.: 3D Shape Reconstruction Using Three Light Sources in Image Scanner. *IEICE Trans. on Inf. & Syst.* E84-D(12), 1713–1721 (2001)
- [3] Ukida, H., Tanimoto, Y., Sano, T., Yamamoto, H.: 3D Shape Reconstructions Using Image Scanner under Various Number of Illuminations. In: *IEEE IST '07, Krakow, Poland*, May 5, pp. 1–6 (2007)
- [4] Zhang, R., Tsai, P.S., Cryer, J.E., Shah, M.: Shape from shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21(8), 690–706 (1999)
- [5] Durou, J.-D., Falcone, M., Sagona, M.: Numerical Methods for Shape-from-shading: A New Survey with Benchmarks. *Computer Vision and Image Understanding* (2007)
- [6] Szeliski, R.: *Computer Vision: Algorithms and Applications* (to be published), <http://research.microsoft.com/en-us/um/people/szeliski/Book/>
- [7] Oliveira, D.M., Lins, R.D.: A New Method for Shading Removal and Binarization of Documents Acquired with Portable Digital Cameras. In: *CBDAR 2009*, pp. 3–10. IAPR Press, Barcelona (2009)
- [8] Bukhari, S.S., Shafait, F., Breuel, T.M.: Dewarping of Document Images using Coupled-Snakes. In: *CBDAR 2009*, pp. 17–24. IAPR Press, Barcelona (2009)
- [9] Stamatopoulos, N., Gatos, B., Pratikakis, I., Perantonis, S.J.: A two-step dewarping of camera document images. In: *Proceedings 8th IAPR Workshop on Document Analysis Systems*, pp. 209–216 (2008)
- [10] Pintus, R., Malzbender, T., Wang, O., Bergman, R., Nachlieli, H., Ruckenstein, G.: Photo Repair and 3D Structure from Flatbed Scanners. *Computer Vision Theory and Applications*, 40–50 (2009)
- [11] Schubert, R.: Using a Flatbed Scanner as a Stereoscopic Near-Field Camera. *IEEE Computer Graphics and Applications*, 38–45 (2000)
- [12] Fan, J.: Enhancement of Camera-captured Document Images with Watershed Segmentation. In: *CBDAR'07, September 2007*, pp. 87–93 (2007)
- [13] Fan, J.: Robust Color Image Enhancement of Digitized Books. In: *Proceedings of International Conference on Document Analysis and Recognition*, pp. 561–565. IEEE Press, Los Alamitos (2009)
- [14] Wolfram Research. Least Squares Fitting–Exponential, <http://mathworld.wolfram.com/LeastSquaresFittingExponential.html> (accessed October 14, 2009)
- [15] Lins, R.D., Oliveira, D.M., Torreão, G., Fan, J., Thielo, M.: A De-Warping Algorithm to Compensate Volume Binding Distortion in Scanned Documents. In: *ACM SAC 2010* (March 2010)
- [16] Silva, G.F.P., Lins, R.D.: PhotoDoc: A Toolbox for Processing Document Images Acquired Using Portable Digital Cameras. In: *CBDAR 2007, Curitiba*, pp. 107–115 (2007)

D. M. Oliveira, R. D. Lins, G. Torreão, J. Fan and M. Thielo. *A New Deblurring Algorithm for Textual Document Images*. Proceedings of the International Telecommunications Symposium, Manaus, Brasil, 2010.

## A New Deblurring Algorithm for Textual Document Images

Daniel M. Oliveira, Rafael Lins, Gabriel P. Silva  
Departamento de Eletrônica e Sistemas - UFPE  
Recife - Brazil  
{daniel.moliveira, rdl, gabriel.psilva}@ufpe.br

Jian Fan<sup>1</sup>, Marcelo Thielo<sup>2</sup>  
Hewlett-Packard Labs.  
<sup>1</sup>Palo Alto - USA, <sup>2</sup>Porto Alegre - Brazil  
{jian.fan, marcelo.resende.thielo}@hp.com

**Abstract**—Documents digitalized by portable cameras or flatbed scanners may exhibit some blurred areas. Most deblurring algorithms are hard to implement and slow. Often they try to solve the problem for any kind of image. In the case of text document images, the transition between characters and the paper background has a high contrast. With that in mind, a new algorithm is proposed for deblurring of textual documents; there is no need to estimate the PSF and the filter can be directed applied to the image.

*Deblurring; blur; camera documents; scanner documents;*

### I. INTRODUCTION

The recent paper [9] presents a taxonomy for noises in document images and, besides providing an explanation of how such noise appeared in the final image, may provide pointers to the literature that show ways of avoiding or removing it. Noise is defined here as any phenomenon that degrades document information. In the classification proposed [9], there are four kinds of noise:

1. *The physical noise – whatever “damages” the physical integrity and readability of the original information of a document. The physical noise may be further split into the two sub-categories proposed in as internal and external.*
2. *The digitalization noise – the noise introduced by the digitalization process. Several problems may be clustered in this group such as: inadequate digitalization resolution, unsuitable palette, framing noises, skew and orientation, lens distortion, geometrical warping, out-of-focus digitized images, motion noises.*
3. *The filtering noise – unsuitable manipulation of the digital file may degrade the information that exists in the digital version of the document (instead of increasing it). The introduction of colors not originally present in the document due to arithmetic manipulation or overflow is an example of such a noise.*
4. *The storage/transmission noise – the noise that appears either from storage algorithms with losses or from network transmission. JPEG artifact is a typical example of this kind of undesirable interference.*

Blur is the effect of unsharpening images, which may fit the four categories above. The physical blur may be the result of document “washing”, for instance, in which a document, printed with water soluble ink, gets wet. Blur may also be the result of unsuitable digitalization, due to several reasons: non-flat objects, digitalization errors, out of focus, motion etc. The presence of blur may be an indicator of low quality digitalization, but can also be associated with other problems such as the scanning of hard-bound volumes. Blur may be the result of unsuitable filtering, such as a Gaussian or low-pass filter. And finally, blur may appear as the result of storing images in a file format with losses that perceptually degrades the image.

The technical literature points at several approaches proposed for deblurring images in general. To list a few of them: Demoment [2] uses statistics, Neelamani, Choi, and Baraniuk [3] use Fourier and wavelet transforms, references [4] and [5] apply variational analysis, and Roth and Black [6] use total variation and Field of experts. Most times the computational complexity is prohibitively high and can yield undesirable artifacts such as ringing [7] as presented in Figure 1.



**Figure 1.** Ringing artifact [7]

The most successful approaches to blur removal point at focusing at one specific kind of blur. For instance, the literature presents several algorithms [11, 12, 13, 14, 15, 16, 17] that address the problem of motion blur, an specific kind of digitalization noise.

In this paper, to increase the chances of better deblurring, the application domain is restricted to monochromatic

scanned documents in which the blur is a digitalization noise originated from the unsuitable document placing on the scanner flatbed due to a number of factors, one of which is book binding warping [10]. The document images treated here are basically constituted by text and plain paper background. The transition between them in the original physical document is sharp. Using this fact a new algorithm is proposed by using nearby pixels to increase the difference between them, no Point Spread Function (PSF) [18] estimation is done and blur is minimized into a direct application of the image.

## II. THE NEW METHOD

### A. Blur effect

The study performed here on the compensation of the blur effect was done for scanned images. Patterns were arranged in an elevated plane model [1] as illustrated in Figure 2 with  $\psi = 0$ . A HP PhotoSmart C4280 and HP ScanJet 5300c scanners were used. Some of the results are shown in Figure 3, where one may observe that in this case blur kernel can be approximated to a 1D horizontal function.

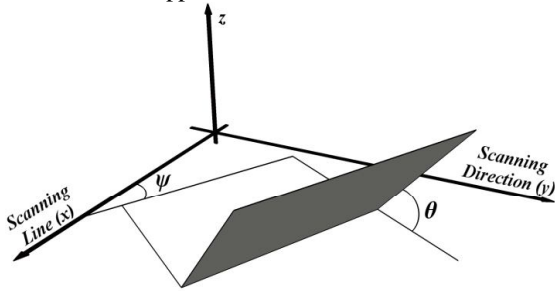


Figure 2. Elevated plane [1]

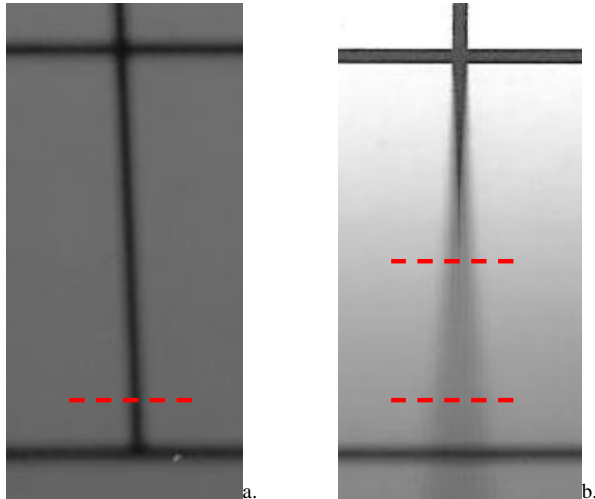


Figure 3. Grid image: ScanJet,  $\theta = 45^\circ$  (a), PhotoSmart C4280,  $\theta = 30^\circ$  (b)

In both images shown in Figure 3, as the paper is farther away from the scanner flatbed the blur increases and illumination is weaker; this happens due to the scanning device being calibrated to digitalize documents at a pre-defined distance that is exactly the flatbed surface.

Figure 4 shows cross sections of the image of Figure 3 on an area with and without blur. One may observe that the signal in part 4.e the blur kernel is greater than the grid thickness, in this case it is not possible to recover the unblurred signal. Besides that, the blur is stronger in the HP Photosmart than in the ScanJet as the first is a “All-in-one” device (printer, fax and scanner) and has less space to place the components whereas the latter is a single purpose scanner.

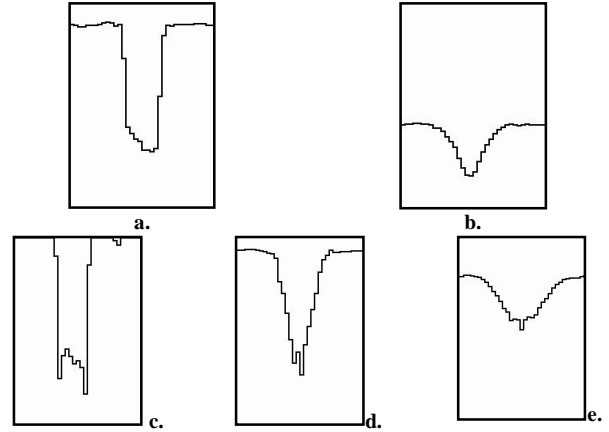


Figure 4. Cross section signals: Figure 3.a unblurred (a) and blurred (b); Figure 3.b unblurred (c), blur kernel close to limit (d) and totally blurred (e)

### B. Reconstruction function

Figure 4 shows that if a signal has a blur kernel smaller than the grid thickness it is possible to improve or even recover the un-blurred signal. In the case of characters, the scenario is similar in a smaller scale.

For the values that belong to the paper background, the blurred intensity value is closer to the unblurred ones. In the same way, for blurred stokes values are closer to undistorted part. In this way a S-function can be built, with input and output varying from 0 to 1, whereas the output is below the line of the identity function between 0 and 0.5, and above it between 0.5 and 1.0.

This work proposes function  $S(t)$  with the fixed parameter  $p$  that varies between 0 and 1.0, which controls how strong the correction will be. For  $p$  values closer to 0, the function shape looks similar to a step function with higher transitions; for values closer to 1.0, the shape gets closer to a sin function scaled by  $\pi$ . Figure 5 shows two plots for  $p = 0.06$  and  $p = 1.00$ .

$$S(t) = 0.5 - 0.5 \times \text{sign}(\cos t\pi) \times |\cos t\pi|^p \quad (1)$$

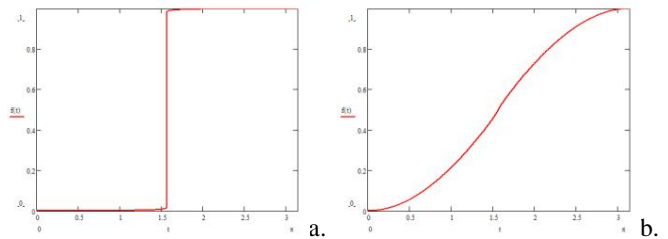


Figure 5. S-function plots:  $p = 0.06$  (a);  $p = 1.0$  (b)

To apply the S-shape function, two reference values must be determined for the paper background and character stroke.



This is done by looking out in a window for the pixel with largest and lowest intensity, then to obtain the unblurred value eq. (2) is applied, where  $I_b$  is the blurred intensity value (i.e. original image), min and max are the lowest and the highest intensity values in the given window, respectively.

$$I_u = \left[ S \left( \frac{I_b - \min}{\max - \min} \right) \times (\max - \min) \right] + \min \quad (2)$$

### III. RESULTS AND ANALYSIS

Figure 6 shows the results of applying the proposed algorithm to Figure 3.b using a 7x7 window. One may observe that vertical line grid was recovered until blur kernel got larger than a 7x7 window; although the blurred horizontal line on the bottom part could be partially recovered.

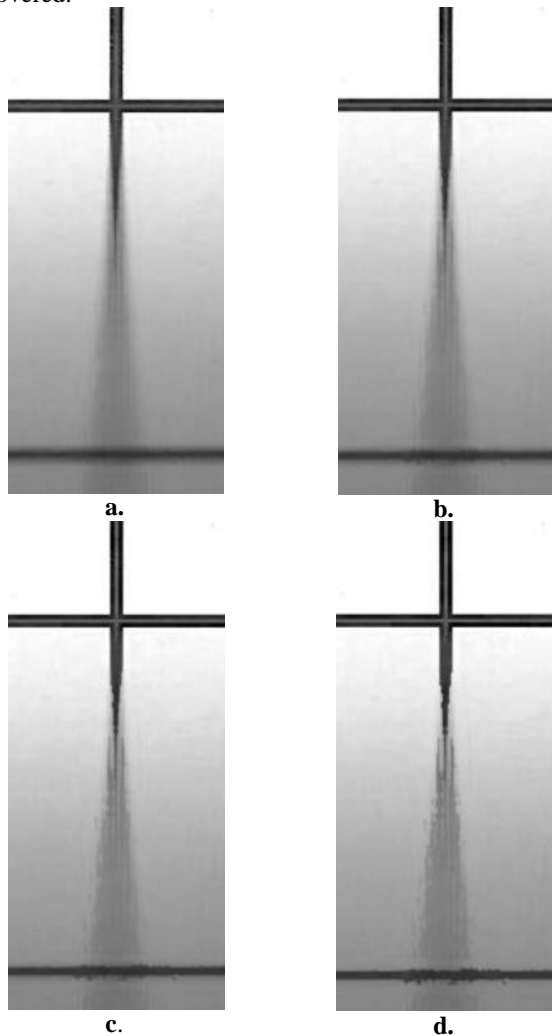


Figure 6. Results for a 7x7 window:  $p = 1.00$  (a);  $p = 0.5$  (b);  $p = 0.25$  (c);  $p = 0.06$  (d)

Increasing the window size is possible to recover the area where the blur is larger, which is shown in Figure 7 with windows of sizes 11x11 and 19x19 and  $p = 0.25$ .

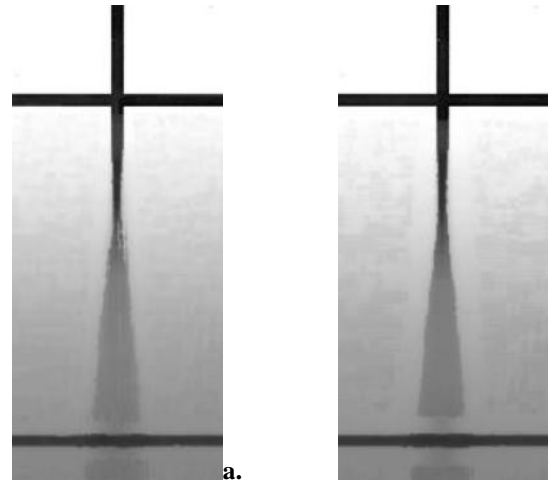


Figure 7. Results with  $p = 0.25$ : 11x11 window (a); 19x19 window (b);

For figure 2.a, in which the blur is weaker than in 2.b, Figure 8 provides the results for same parameters. One may note that satisfactory results were obtained for  $p = 0.5$  with a 7x7 window.

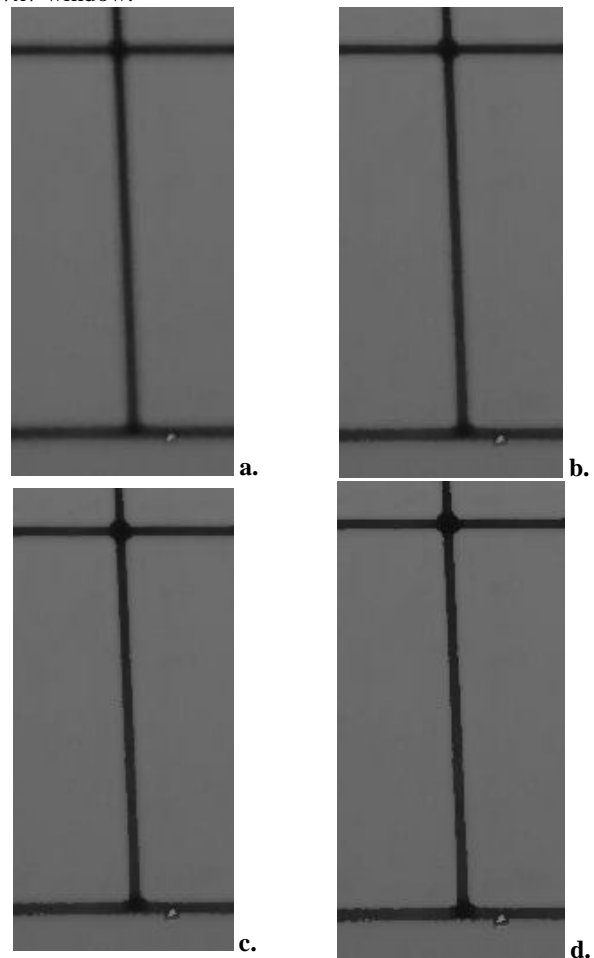


Figure 8. Results for 7x7 window:  $p = 1.00$  (a);  $p = 0.5$  (b);  $p = 0.25$  (c);  $p = 0.06$  (d)

Finally, Figure 9 shows some examples of de-blurring applied to real document images.

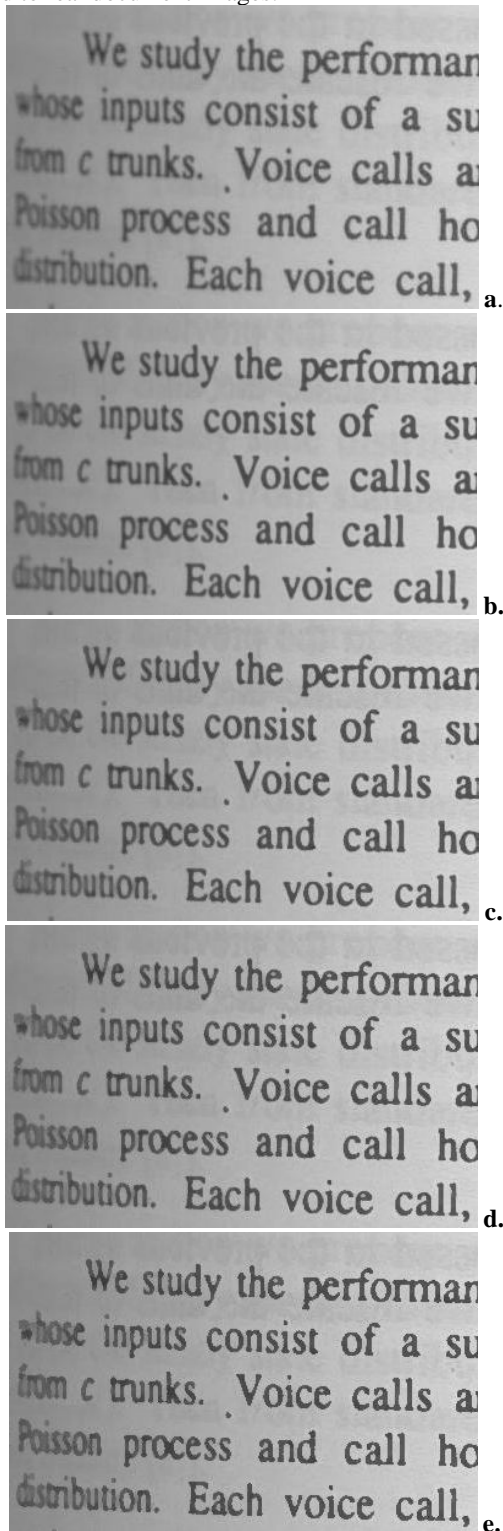


Figure 9. Document processed with 5x5 window: original image (a);  $p = 1.0$  (b);  $p = 0.50$  (c);  $p = 0.25$  (d);  $p = 0.06$  (e)

#### IV. CONCLUSIONS

The study performed here shows that focusing the scope of application of deblurring algorithms stand a better chance of better solving such a complex noise that may appear due to several sources: physical, digitalization, filtering and storage. This paper presented an algorithm to compensate the digitalization blur in scanned monochromatic documents with satisfactory results.

The automatic inference of the parameters of the algorithm through the use of blur intensity classifier such as the one described in reference [8] is under implementation and shows some promising results already.

#### REFERENCES

- [1] Ukida, H. and Konishi, K. . 3D Shape Reconstruction Using Three Light Sources in Image Scanner. *IEICE Trans. on Inf. & Syst.*, Vol.E84-D, No. 12, pp.1713-1721, Dec. 2001.
- [2] Demoment, G.. Image reconstruction and restoration: Overview of common estimation structures and problems. *IEEE Transactions on Acoustics, Speech, & Signal Processing*, 37(12), 2024–2036, 1989.
- [3] Neelamani, R., Choi, H., and Baraniuk, R. G.. Wavelet-based deconvolution for ill-conditioned systems. *Proc. of IEEE ICASSP*, Vol. 6, pp. 3241–3244, March 1999.
- [4] Chambolle, A., and Lions, P. L.. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76(2), 167–188, 1997.
- [5] Rudin, L. I., Osher, S., and Fatemi, E.. Nonlinear total variation based noise removal algorithms. *Physica D*, 60, 259–268, 1992.
- [6] Roth, S., and Black, M. J.. Fields of experts: a framework for learning image priors. *CVPR*, Vol. 2, pp. 860–867, 2005.
- [7] Joshi, N. S.. Enhancing photographs using content-specific image priors. Phd thesis, University of California, San Diego, 2008.
- [8] Lins, R.D, Silva, G.F.P., Banergee, S., Kuchibhotla, A. and Thielo, M. Automatically Detecting and Classifying Noises in Document Images, *ACM-SAC'2010*, ACM Press, v.1. p.33 – 39, March 2010.
- [9] Lins, R.D. A Taxonomy for Noise Detection in Images of Paper Documents - The Physical Noises. *ICIAR 2009*. LNCS v. 5627. p. 844-854, Springer Verlag, 2009.
- [10] Lins, R.D., Oliveira, D. M., Torreão G., Fan, J., and Thielo, M., Correcting Book Binding Distortion in Scanned Documents. *ICIAR 2010*. LNCS 6112, pp. 398-408, Springer Verlag, 2010.
- [11] Chang, M. M., Tekalp, A. M., and Erdem, A. T., Blur identification using the bispectrum, *IEEE Trans. Signal Process.*, Vol. 39, No. 10, 1991, pp. 2323-2325.
- [12] Mayntz, C., Aach T., and Kunz D., Blur identification using a spectral inertia tensor and spectral zeros, *Proc. of IEEE ICIP*, 1999.
- [13] Cannon M., Blind deconvolution of spatially invariant image blurs with phase, *IEEE Trans. Acoust. Speech Signal Process.*, Vol. 24, No. 1, 1976, pp. 56-63.
- [14] Biemond, J., Lagendijk, R. L., and Mersereau R. M., Iterative methods for image deblurring, *Proc. of the IEEE*, 1990, pp. 856-883.
- [15] Rekleities, I. M., Optical flow recognition from the power spectrum of a single blurred image, *Proc. of IEEE ICIP*, 1996.
- [16] Moghaddam, M.E. and Jamzad, M., Motion blur identification in noisy images using fuzzy sets, *Proc. IEEE ISSPIT*, Athens, 2005.
- [17] Lokhande, R., Arya, K.V., Gupta, P. Identification of parameters and restoration of motion blurred images, *ACM-SAC'2006*, Dijon, 2006.
- [18] Jain, A.K. , *Fundamentals of digital image processing*, Prentice-Hall, Inc., Upper Saddle River, NJ, 1989.

D. M. Oliveira, R. D. Lins e G. S. Torreão. *Deblurring Textual Document Images*  
(Submetido para ser publicado no LNCS do GREC 2011).

# De-blurring Textual Document Images

Daniel M. Oliveira<sup>1</sup>, Rafael Lins<sup>1</sup>, Gabriel P. Silva<sup>1</sup>, Jian Fan<sup>2</sup>, Marcelo Thielo<sup>3</sup>

<sup>1</sup>Universidade Federal de Pernambuco

Recife - Brazil

{daniel.moliveira, rdl, gabriel.psilva}@ufpe.br

Hewlett-Packard Labs.

<sup>1</sup>Palo Alto - USA, <sup>2</sup>Porto Alegre - Brazil

{jian.fan, marcelo.resende.thielo}@hp.com

**Abstract.** Document images may exhibit some blurred areas due to a wide number of reasons ranging from digitalization, filtering or even storage problems. Most de-blurring algorithms are hard to implement, slow, and often try to be general, attempting to remove the blur in any kind of image. In the case of text document images, the transition between characters and the paper background has a high contrast. With that in mind, a new algorithm is proposed for de-blurring of textual documents; there is no need to estimate the PSF and the filter proposed can be directed applied to the image. The presented algorithm reached an improvement rate of 17.08% in the SSIM metric.

Keywords: De-blurring, blur, camera documents, scanner documents.

## 1 Introduction

Noise is any phenomenon that degrades information. A taxonomy for noises in document images is proposed in reference [9] which besides providing an explanation of how different noises appeared in the final image, it gives pointers to the literature that show ways of avoiding or removing them. In the classification proposed [9], there are four kinds of noise:

1. *The physical noises – whatever “damages” the physical integrity and readability of the original information of a document. It may be further split into the two sub-categories proposed in as internal and external.*
2. *The digitization noises – introduced by the digitization process. Several problems may be clustered in this group such as: inadequate digitization resolution, unsuitable palette, framing noises, skew and orientation, lens distortion, geometrical warping, out-of-focus digitized images, motion noises.*
3. *The filtering noise – unsuitable manipulation of the digital file may degrade the information that exists in the digital version of the document (instead of increasing it).*
4. *The storage/transmission noise – the noise that appears either from storage algorithms with losses or from network transmission. JPEG artifact is a typical example of this kind of undesirable interference.*

The blur noise has the effect of unsharpening images. Depending on how it arises it may be included in any of the four categories above. The physical blur may be the result of document “washing”, for instance, in which a document, printed with water soluble ink, gets wet. Blur may also be the result of unsuitable digitization, due to several reasons: non-flat objects, digitization errors, out of focus, motion etc. The presence of blur may be an indicator of low quality digitization, but can also be associated with other problems such as the scanning of hard-bound volumes. Blur may be the result of unsuitable filtering, such as a Gaussian or low-pass filter. And finally, blur may appear as the result of storing images in a file format with losses that perceptually degrades the image.

The technical literature points at several approaches proposed for de-blurring images in general. To list a few of them: Demoment [2] uses statistics, Neelamani, Choi, and Baraniuk [3] use Fourier and wavelet transforms, references [4] and [5] apply variational analysis, and Roth and Black [6] use total variation and Field of experts. Most of times, the computational complexity of those algorithms is prohibitively high and can yield undesirable artifacts such as ringing [7] as presented in Figure 1.

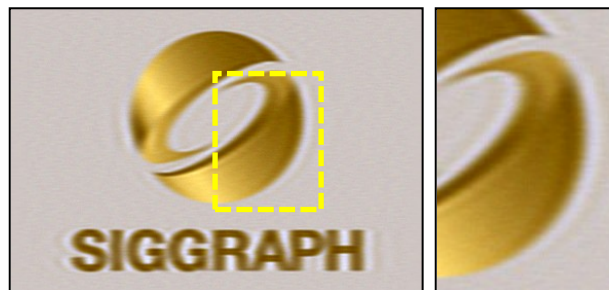


Fig. 1. Ringing artifact [7]

The most successful approaches to blur removal focus at one specific kind of blur. For instance, the literature presents several algorithms [11, 12, 13, 14, 15, 16, 17] that address the problem of motion blur, an specific kind of digitization noise.

In this paper, to increase the chances of better de-blurring, the application domain is restricted to monochromatic scanned documents with book binding warping [10]. The resulting image has uneven blur and illumination. The document images treated here are basically constituted by text and plain paper background. The transition between them in the original physical document is sharp. Using this fact a new algorithm is proposed by using nearby pixels to increase the difference between them. No Point Spread Function (PSF) [18] estimation is done and blur is minimized into a direct application of the image.

## 2 The New Method

The study performed here focus on the compensation of the blur noise which appears in scanning hardbound documents. Patterns were arranged in an elevated plane model [1] as shown in Figure 2 to simulate the hardbound warp. Two HP scanners (PhotoSmart C4280 and a HP ScanJet 5300c) were used to digitize a pattern of lines in a grid. Two examples of blurred cross sections can be seen on Figures 3 and 4,

corresponding to the two different scanning devices, respectively with two elevation and skew angles.

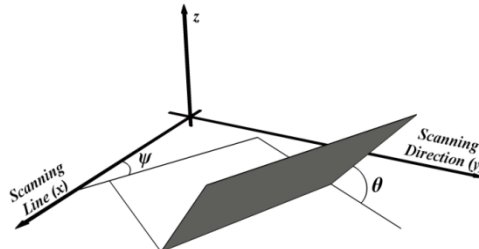


Fig. 2. Elevated plane [1]

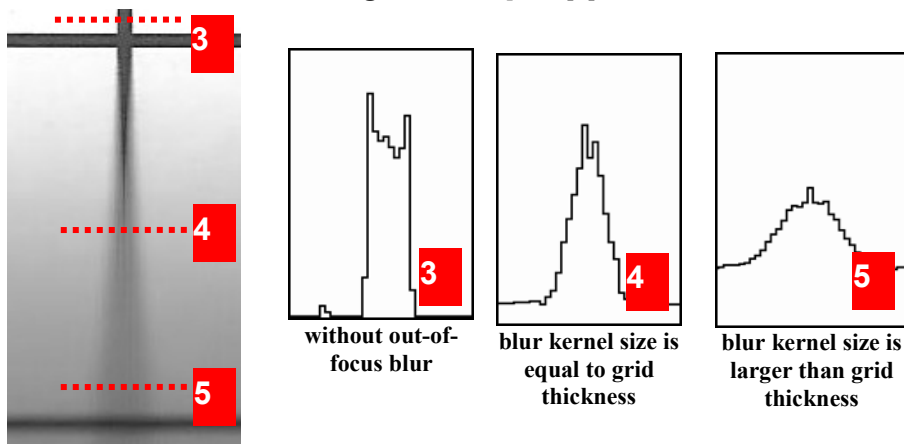


Fig. 3. Line grid scanned with HP PhotoSmart C4280 with  $\psi = 0^\circ$  and  $\theta = 30^\circ$

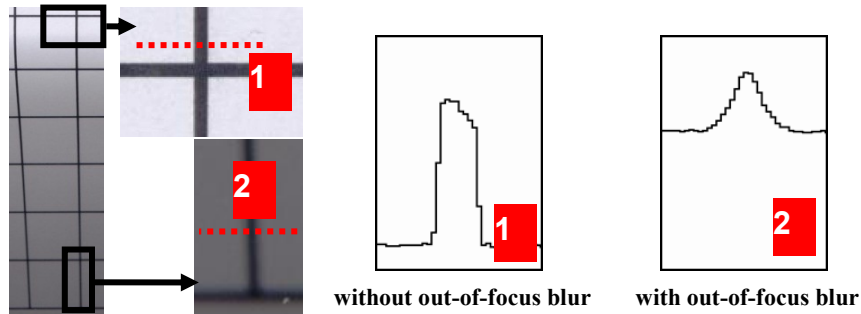


Fig. 4. Line grid scanned with HP ScanJet 5300c with  $\psi = 0^\circ$  and  $\theta = 45^\circ$

In Figures 3 and 4, as the paper is further away from the scanner flatbed, the blur increases and illumination is fades out; as the scanning device is calibrated to digitize documents at a pre-defined distance, which is exactly the flatbed surface.

These figures also present several cross sections at different parts of the calibration grid images. They show regions without blur (cross sections “1” and “3”) and regions with blur kernel size larger than the grid thickness. The line labeled with number “4” is the limit when is not possible to remove the blur totally.

In the case of characters, corners of the strokes are vulnerable regions to the blur. The kernel area in this region is dominated by the information not related to the given point. Figure 5 shows two kinds of corners in the letter “M” that can be irrecoverable.



Fig. 5. Corners of upper case “M” which are vulnerable to the blur noise

## 2.1 Reconstruction function

Thoulin and Chang [21] identify document background and foreground locally for the resolution expansion of document images. Proposed method obtains these colors by searching the maximum and minimum on the pixel neighborhood and uses it into the reconstruction function.

Most pixels that belong to the paper background have their intensities values closer to the background intensity. Similarly, for blurred strokes values are closer to the foreground intensity. In this way an S-function can be built, with input and output varying from 0 to 1, whereas the output is below the line of the identity function between 0 and 0.5, and above it between 0.5 and 1.0.

In this work the function  $S(t)$  is defined by equation 1 with the fixed parameter  $p$  that varies between 0 and 1.0, which controls how strong the correction will be. For  $p$  values closer to 0, the function shape looks similar to a step function with higher transitions; for values closer to 1.0, the shape gets closer to a *sin* function scaled by  $\pi$ . Figure 6 shows two plots for  $p = 0.06$  and  $p = 1.00$ .

$$S(t) = 0.5 - 0.5 \times \text{sign}(\cos t\pi) \times |\cos t\pi|^p \quad (1)$$

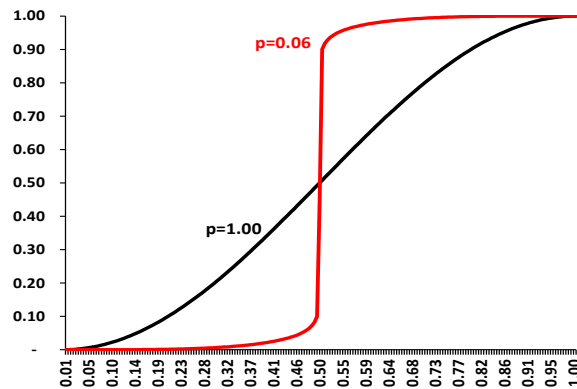


Fig. 6. S-function plots for  $p$  equal to 1.00 and 0.06

To apply the *S-shape* function, two reference values must be determined for the paper background and character stroke. This is done by looking out in a window for the pixel with largest and lowest intensity. The un-blurred value is obtained by eq. (2),

where  $I_b$  is the blurred intensity value (i.e. original image);  $min$  and  $max$  are the lowest and the highest intensity values in the given window, respectively.

$$I_u = \left[ S \left( \frac{I_b - min}{max - min} \right) \times (max - min) \right] + min \quad (2)$$

One may notice that using large windows and low values for  $p$  is more intrusive than the other way round. The choice of this parameter will depend on the blur level.

### 3 Results and analysis

The evaluation of the proposed algorithm is done in three parts. The first part uses computer generated images to obtain blurred and un-blurred images; the latter is used as the ground-truth. At the second part, a study is done in scanned images using the elevated plane model [1]. At last, processing results of real documents are presented.

Unfortunately, comparing the method proposed here with other algorithms described in the literature was not possible as they do not offer enough details for granting their implementation, their executable code is not available, and the evaluation datasets used by them do not include document images.

#### 3.1 Computer generated images

In order to provide objective quality measures of the proposed algorithm processing, letters and chess shape were generated. Several levels of blur were applied in these images using the GaussianBlur filter available in ImageJ [20]; two examples are presented in Figure 7. The Gaussian radius represents the region with 61% of the whole Gaussian, thus the kernel size is not restricted to this radius [20].

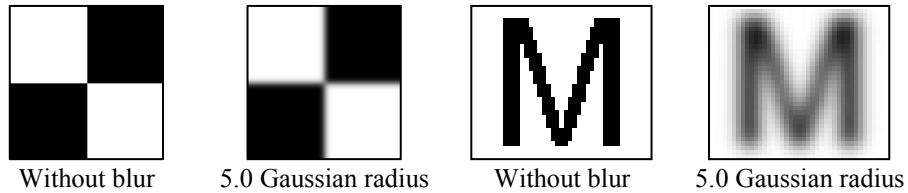


Fig. 7. Blur applied to chess pattern image and upper case “M”

The de-blurring method proposed herein was applied to the blurred images. Images corrected and without blur were compared using SSIM [19] index. It measures how images are perceptually different with values between 0 (different) and 1 (the same).

Tables 2 and 3 provide at each cell average SSIM between the ground truth and the de-blurred images of the chess and letters pattern, respectively. Cells in the same row represent the de-blurring of the same input but with different window sizes. The values in parenthesis are the amount of information “gain” computed by eq. (3). Table 1 shows the values of the average SSIM between de-blurred and without blur images. Value 0.1 was set to  $p$  in all tests. One may notice that the proposed algorithm could improve the SSIM metric by 17.08% at most.

$$SSIM \text{ gain} = \frac{\text{Average of SSIM between **deblurred** and without blur images}}{\text{Average of SSIM between **blurred** and without blur images}} - 1 \quad (3)$$



**Table 1.** Average of SSIM between blurred and without blur images

Blur kernel size	1	2	3	4	5
Average of Chess pattern SSIM	0.999	0.965	0.926	0.888	0.852
Average of Letters SSIM	0.999	0.976	0.954	0.932	0.910

**Table 2.** Average of SSIM between de-blurred and without blur chess pattern images

5	0.923 (+8.41%)	0.970 (+13.94%)	0.989 (+16.16%)	0.996 (+16.96%)	0.997 (+17.08%)
4	0.961 (+8.25%)	0.991 (+11.68%)	0.997 (+12.36%)	0.998 (+12.47%)	0.999 (+12.50%)
3	0.988 (+6.75%)	0.998 (+7.86%)	0.999 (+7.96%)	0.999 (+7.96%)	0.999 (+7.96%)
2	0.999 (+3.55%)	1.000 (+3.64%)	1.000 (+3.64%)	1.000 (+3.64%)	1.000 (+3.64%)
1	1.000 (+0.08%)	1.000 (+0.08%)	1.000 (+0.08%)	1.000 (+0.08%)	1.000 (+0.08%)
Blur Win	1	2	3	4	5

**Table 3.** Average of SSIM between de-blurred and without blur letters pattern images

5	0.942 (+3.56%)	0.947 (+4.06%)	0.940 (+3.30%)	0.941 (+3.38%)	0.942 (+3.53%)
4	0.968 (+3.85%)	0.973 (+4.37%)	0.970 (+4.09%)	0.971 (+4.17%)	0.972 (+4.27%)
3	0.987 (+3.44%)	0.991 (+3.85%)	0.991 (+3.82%)	0.990 (+3.78%)	0.991 (+3.80%)
2	0.999 (+2.30%)	0.998 (+2.25%)	0.998 (+2.23%)	0.998 (+2.23%)	0.998 (+2.22%)
1	1.000 (+0.07%)	1.000 (+0.07%)	1.000 (+0.07%)	1.000 (+0.07%)	1.000 (+0.07%)
Blur Win	1	2	3	4	5

**Table 4.** Letters with various blur sizes applied with different window sizes

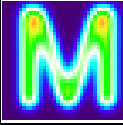
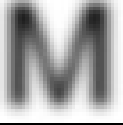
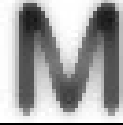


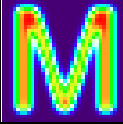
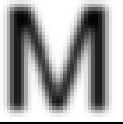
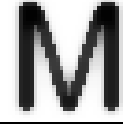


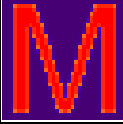
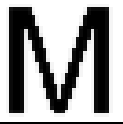
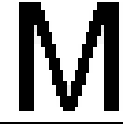
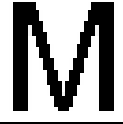
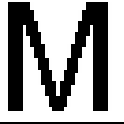
 5				
 3				
 1				
Thermal map	Blur Win.	1	3	5

Table 4 shows some results of processing the examples used for Table 3. The first column shows the blurred images thermal map for each blur level, with red representing the stroke color (black) and blue the background (white). The second column show the blurred images used as inputs.

Images with the Gaussian radius set to 1, is possible to recover most of the stroke. The thermal map of blurred “M” with 3.0 radius shows that only the upper part of the stroke is red, thus only it can be totally recoverable. Although, the corrected images appear to be restored properly as the blurred pixels are too close to black.

The same is not observed to the correct images of radius set to 5. The thermal map shows that most part of the stroke are green, showing that the blur affected the character structure. The quality improvement is observable in all images on Table 4.

### 3.2 Blurred images generated from scanned images

The dataset presented on previous section has noise-free images. In order to evaluate the proposed algorithm with more realistic scenarios, 3 documents were digitized by a scanner flatbed yielding into images with low noise level, which is considered as the ground truth. Analogically to previous section, the blurred images and the effectiveness of the correction were obtained. Table 5 shows the SSIM performance results.

**Table 5.** Average of SSIM between de-blurred and scanned images

5	0.898 (+3.02%)	0.899 (+3.05%)	0.895 (+2.62%)	0.893 (+2.36%)	0.890 (+2.12%)
4	0.924 (+2.52%)	0.920 (+2.05%)	0.916 (+1.61%)	0.914 (+1.37%)	0.912 (+1.16%)
3	0.944 (+1.32%)	0.938 (+0.58%)	0.934 (+0.18%)	0.931 (-0.10%)	0.929 (-0.37%)
2	0.960 (-0.25%)	0.952 (-1.11%)	0.946 (-1.67%)	0.942 (-2.15%)	0.938 (-2.58%)
1	0.969 (-2.94%)	0.946 (-5.27%)	0.928 (-7.04%)	0.915 (-8.40%)	0.905 (-9.42%)
Blur Win	1	2	3	4	5

The scanned images present a low degree of blur; thus the similarities between the scanned and weakly blurred images are too high. Therefore, the gains of the SSIM quality are negative for the images with 1-3 radiuses on Table 5.

The de-blurred image provides image with sharper edges than the ground truth and input images, thus it has better visual quality than both of these images. Figure 8.a shows the scanned image; a zoomed part can be seen on (b); (c) shows the blur level 1 applied to (b) with the de-blurred version on (d). It is noticeable that Figure 8.d is more pleasant to see than Figures 8.b and 8.c.

For the strong blur (4 and 5 radius), the SSIM gain is positive; this shows the proposed algorithm improves the quality of the image. One can be seen on the de-blurred version of Figure 8.e (8.b blur level 5) on Figure 8.f.



### 3.3 Elevated plane image analysis

The images obtained with the elevated plane model [1] shows the blurred stroke combined with other noises. Figure 9 shows the results of applying the proposed algorithm to Figure 3 using a  $7 \times 7$  window for  $p$  equal to 1.00 (a) and 0.25 (b). One may observe that vertical line grid was recovered until blur kernel got larger than a  $7 \times 7$  window (dashed rectangle of 8.b); although the blurred horizontal line on the bottom part could be partially recovered. Increasing the window size is possible to recover the area where the blur is larger, which is shown in Figure 9 with windows of sizes  $11 \times 11$  (c) and  $19 \times 19$  (d) for  $p = 0.25$ .

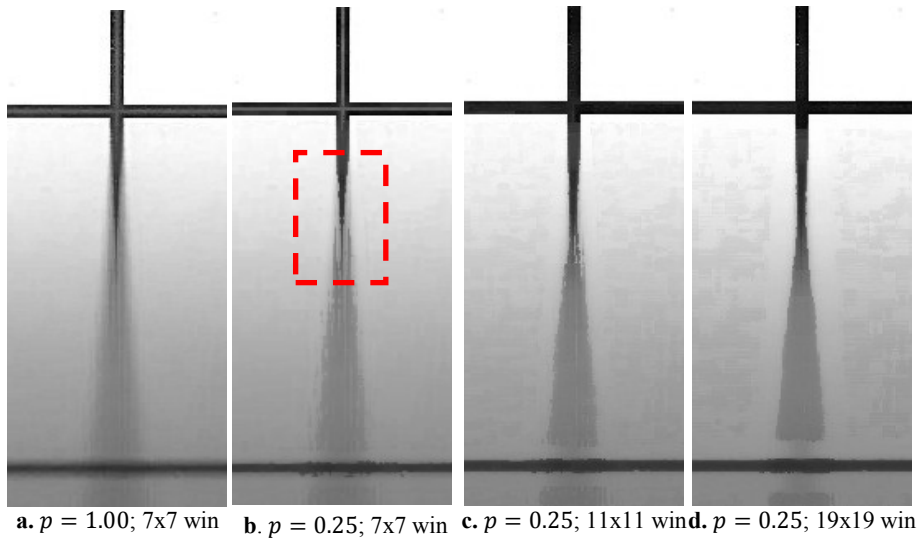


Fig. 9. Output with  $7 \times 7$  window

### 3.4 Document images

Finally, some examples of de-blurring applied to real document images are presented. Figures 10 and 11 show the output at different window sizes and values of  $p$ . The resulting image was improved by the proposed algorithm.

## 4 Conclusions

The study performed here shows that focusing the scope of the application of a de-blurring algorithm stands a better chance of more adequately and efficiently removing such complex noise that may appear due to several different sources: physical, digitization, filtering and storage. This paper presents an algorithm to compensate the digitization non-constant blur that appear in scanning bound grayscale documents, for instance. The algorithm performance was confirmed by high values of SSIM metric in computer generated images.

The automatic inference of the parameters of the algorithm through the use of blur intensity classifier such as the one described in reference [8] is under implementation and shows some promising results already.

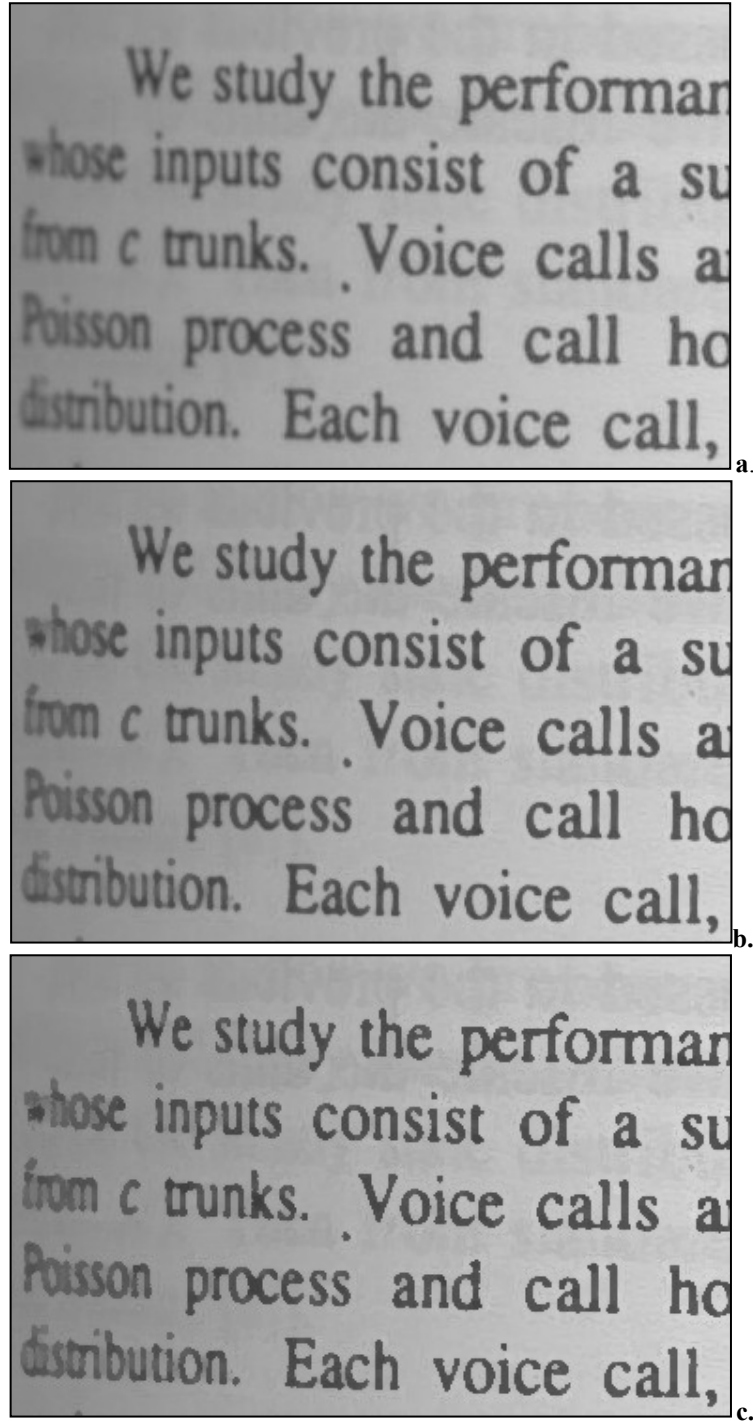
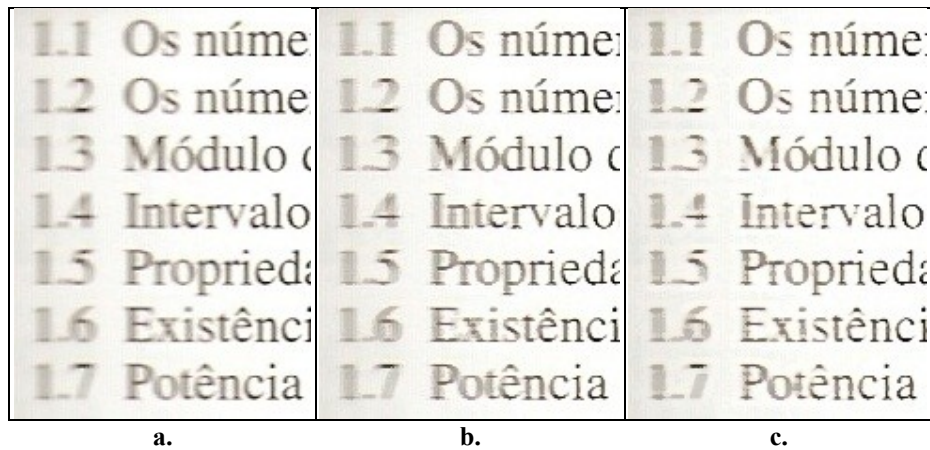


Fig. 10. Result with 5x5 window: original image (a);  $p = 0.50$  (b);  $p = 0.06$  (c)



**Fig. 11.** Result with  $p=0.5$ : original image (a); 3x3 window (b); 7x7 window (c)

### Acknowledgements

The research reported herein was partly sponsored by CNPq and MCT-Lei de Informática grants both from the Brazilian Government.

### References

- [1] Ukida, H. and Konishi, K. . 3D Shape Reconstruction Using Three Light Sources in Image Scanner. IEICE Trans. on Inf. & Syst., Vol.E84-D, No. 12, pp.1713-1721, Dec. 2001.
- [2] Demoment, G.. Image reconstruction and restoration: Overview of common estimation structures and problems. IEEE Transactions on Acoustics, Speech, & Signal Processing, 37(12), 2024–2036, 1989.
- [3] Neelamani, R., Choi, H., and Baraniuk, R. G.. Wavelet-based deconvolution for ill-conditioned systems. Proc. of IEEE ICASSP, Vol. 6, pp. 3241–3244, March 1999.
- [4] Chambolle, A., and Lions, P. L.. Image recovery via total variation minimization and related problems. Numerische Mathematik, 76(2), 167–188, 1997.
- [5] Rudin, L. I., Osher, S., and Fatemi, E.. Nonlinear total variation based noise removal algorithms. Physica D, 60, 259–268, 1992.
- [6] Roth, S., and Black, M. J.. Fields of experts: a framework for learning image priors. CVPR, Vol. 2, pp. 860–867, 2005.
- [7] Joshi, N. S.. Enhancing photographs using content-specific image priors. Phd thesis, University of California, San Diego, 2008.
- [8] Lins, R.D, Silva, G.F.P., Banerjee, S., Kuchibhotla, A. and Thielo, M. Automatically Detecting and Classifying Noises in Document Images, ACM-SAC'2010, ACM Press, v.1. p.33 – 39, March 2010.
- [9] Lins, R.D. A Taxonomy for Noise Detection in Images of Paper Documents - The Physical Noises. ICIAR 2009. LNCS v. 5627. p. 844-854, Springer Verlag, 2009.
- [10] Lins, R.D., Oliveira, D. M., Torreão G., Fan, J., and Thielo, M., Correcting Book Binding Distortion in Scanned Documents. ICIAR 2010. LNCS 6112, pp. 398-408, Springer Verlag, 2010.

- [11] Chang, M. M., Tekalp, A. M., and Erdem, A. T., Blur identification using the bispectrum, *IEEE Trans. Signal Process.*, Vol. 39, No. 10, 1991, pp. 2323-2325.
- [12] Mayntz, C., Aach T., and Kunz D., Blur identification using a spectral inertia tensor and spectral zeros, *Proc. of IEEE ICIP*, 1999.
- [13] Cannon, M.. Blind deconvolution of spatially invariant image blurs with phase, *IEEE Trans. Acoust. Speech Signal Process.*, Vol. 24, No. 1, 1976, pp. 56-63.
- [14] Biemond, J., Lagendijk, R. L., and Mersereau R. M., Iterative methods for image de-blurring, *Proc. of the IEEE*, 1990, pp. 856-883.
- [15] Rekleities, I. M., Optical flow recognition from the power spectrum of a single blurred image, *Proc. of IEEE ICIP*, 1996.
- [16] Moghaddam, M.E. and Jamzad, M., Motion blur identification in noisy images using fuzzy sets, *Proc. IEEE ISSPIT*, Athens, 2005.
- [17] Lokhande, R., Arya, K.V., Gupta, P. Identification of parameters and restoration of motion blurred images, ACM-SAC'2006, Dijon, 2006.
- [18] Jain, A.K. , Fundamentals of digital image processing, Prentice-Hall, Inc., Upper Saddle River, NJ, 1989.
- [19] Wang, I. Z., Bovik , A. C., Sheikh, H. R. and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, Apr. 2004.
- [20] ImageJ. GaussianBlur (ImageJ API). Available at: <http://rsbweb.nih.gov/ij/developer/api/ij/plugin/filter/GaussianBlur.html>.
- [21] Thouin, P. D. and Chang, C. I. A method for restoration of low-resolution document images, *IJDAR* 2000.

D. M. Oliveira e R. D. Lins. *Generalizing Tableau to Any Color of Teaching Boards*. In Proceedings of ICPR 2010. pp.2411-2414, 2010.



# GENERALIZING TABLEAU TO ANY COLOR OF TEACHING BOARDS

Daniel M. Oliveira and Rafael D. Lins

Departamento de Eletrônica e Sistemas

Universidade Federal de Pernambuco

Recife, Brazil

{daniel.moliveira, rdl}@ufpe.br

**Abstract**—Teaching boards are omnipresent in classrooms throughout the world. Tableau is a software environment for processing images from teaching-boards acquired using portable digital cameras and cell-phones. The previous versions of Tableau were restricted to whiteboard processing. This paper generalizes the enhancement algorithm to work with boards of any color, being the first software environment able to process non-white boards.

*Enhancement, whiteboard, black board, green board.*

## I. INTRODUCTION

Teaching boards are possibly the oldest and most universal didactic tool used throughout the world. Originally, they were made with slices of dark stone to be written on with a piece of chalk. Over the centuries it evolved to whiteboards used with erasable felt pens. WhiteboardIt [6, 7, 8, 9] pioneered the research in white board image processing as an internal research project at Microsoft.

Independent work lead to the development of Tableau [2,3], a simple software tool to process images of teaching boards acquired using portable digital cameras. Tableau aims to provide a way to generate digital content for courses, respecting particular aspects of the group such as syllabus, class learning speed, teacher experience, regional content, local culture, etc.

The system consists of three parts. The first is database formation. As soon as images are transferred from the camera to the PC, information is collected to generate a simple database that will organize images for later content formation. Information such as teacher name, course name, discipline, subject, class number, group number, etc. are requested. The second module is for image processing. This module improves the image acquired in a number of ways: background removal, image segmentation, skew correction, image enhancement, etc. The third part of the processing environment deals with outputting the content. Three different ways are under development: printed handouts, webpage generation and slide production. Each of these media receives the information of the processed image part of the environment and makes it suitable to its best use.

Although whiteboards have the advantage over chalkboards of generating no chalk dust, that causes allergy to teachers and students, chalkboards are still of widespread use, overall in developing countries. Thus, to reach its original aim to help students and teachers to generate didactic content, Tableau could not restrict its applicability to whiteboards, and had to be generalized to any color of board. This paper presents the new image enhancement algorithm developed for Tableau to work with boards of any color. The proposed algorithm generalizes previous version of Tableau on reference [2]. It was tested on 102 board pictures (green, black and white) collected from Flickr [10] and Picasa [11] users, also from W3C website [16]. Besides those images, 75 whiteboards photos were taken with a Olympus C60 portable digital camera and 8 pictures were taken with a Nokia 6120 classic portable phone. In all cases, Tableau was able to work satisfactorily. The current version of Tableau is implemented as a plug-in for ImageJ and is available at: <http://sites.google.com/site/tableauboard/>.

## II. PROCESSING COLOR BOARD IMAGES

The new image enhancement algorithm divides the board image in blocks. The first step scans the whole image to identify which belongs to the board background; the second step re-scans the image to normalize the illumination. The following assumptions are made:

- The image only contains the board itself without the board frame or any object that surrounds it.
- The board has one predominant background color (green, white, black, etc).

Tableau as well as the other board processing tools satisfactorily pre-process images to remove board surroundings object [2,3] (second assumption above).

## III. FINDING BACKGROUND BLOCKS

The search for board background blocks is a two-step process. The first step is a local search based on the RGB-histogram of each block that sieves possible candidates. The second step is a global outlier block candidate elimination.

### A. Local analysis

The board image is split into blocks of 20x20 pixels. This block size is determined empirically by considering if RGB-histogram of block of pixels for exhibits a Gaussian distribution and background blocks has uniform illumination. Figure 1 presents sample blocks that are background with their histograms.



Figure 1. Example of background blocks and their histograms

Each block is classified as candidate to belong to the image background by:

1. Compute its RGB-histogram
2. Find the mode  $M_R, M_G, M_B$  for each component.
3. Find the number of pixels  $Q_R$  in the red histogram which is between  $M_R - 10$  and  $M_R + 10$ . Repeat for  $Q_G$  and  $Q_B$ . If  $Q_R \geq 75\%$ ,  $Q_G \geq 75\%$ , and  $Q_B \geq 75\%$  of the total of pixels of the block, then it is a candidate to be board background block, its color is set equal to  $(M_R, M_G, M_B)$ .

At the end of this process one has a list  $C$  of candidates to board background blocks.

### B. Global Outlier Block Elimination

Now, one starts a clustering process to identify the actual board background blocks. Color component scale used here are between 0 and 255.

1. For each pair of blocks  $C_i$  and  $C_j$  in  $C$ , if the absolute difference of the local background color in every component is less than 6 (value empirically found), set this pair to same cluster set.
2. Cluster sets with common entries may be merged until all sets are disjoint.
3. The cluster set with the largest cardinality is formed by the board background blocks. The global background color is obtained by averaging each RGB component of the blocks from this cluster.
4. The local background color for the non-background blocks is defined by the average of the local background color of the closest neighbors marked as background blocks.

One may observe that the inferred background for whiteboard most times is not equal to the pure RGB white (255,255,255), as the inferred global background is also affected by the uneven illumination of the environment. This is also true for boards of other colors. An alternative scheme to overcome this is to make a list of most common board colors on (R,G,B): (0,0,0) for black; (255,255,255) for white; and (0,82,0) for green. The color of the global background is set to the closest color from this list to the inferred global background in step 3 using color difference CIEDE2000 [1] with the changes proposed in [2]. Unfortunately, images used in the study had an error rate of 25% using CIEDE2000. To avoid this error, the user may be asked to confirm which background color may be adopted.

## IV. IMAGE ENHANCEMENT

The purpose of this step is to make colors sharper and uniform, better differentiating the board from the writing strokes. Color transformation should compensate the effect of uneven illumination. To obtain the enhanced pixel value, each component is set up independently. Let the local and global background component values be  $B_l$  and  $B_g$ , respectively. And let the original pixel and the enhanced pixel component values be  $P_o$  and  $P_e$ , respectively. The main idea is to keep the ratio between a pixel with local and global backgrounds, which is done by equalities (1) and (2), where  $P_e$  is the unknown desired value. The following scenarios are considered to calculate  $P_e$ :

$$\frac{P_e}{B_g} = \frac{P_o}{B_l} \quad (1) \quad \frac{1 - P_e}{1 - B_g} = \frac{1 - P_o}{1 - B_l} \quad (2)$$

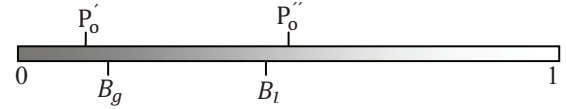


Figure 2. Scenarios for  $P_e$  computation

1. if  $P_o < B_l$  ( $P'_o$  of figure 2), to find  $P_e$  multiply both sides of (1) by  $B_g$ .
2. if  $P_o > B_l$  ( $P''_o$  of figure 2), equation (2) is used instead of equation (1), due to the ratio  $\frac{P_o}{B_l}$  be in  $(1, \infty]$ , while in equation (2) the ratio  $\frac{1 - P_o}{1 - B_l}$  is in  $[0, 1)$ , which is easier to handle computationally. A direct way to find  $P_e$  is shown in eq. (3).  

$$P_e = 1 - (1 - P_o) \times (1 - B_g) / (1 - B_l) \quad (3)$$
3. if  $P_o = B_l$ , the value of  $P_e$  is  $B_g$ .

The result of the algorithm as described so far can be seen in Figure 3, where one may see that the yielded background around the fish in the filtered image is noisy and uneven.

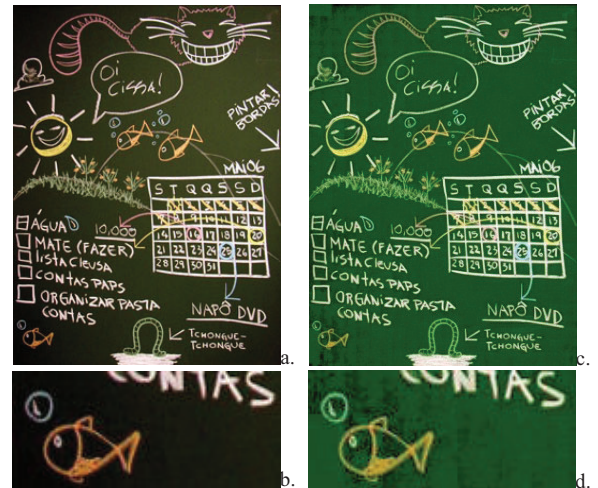


Figure 3. Result of the algorithm as described so far: original image (a); (a) details (b); enhanced image (c); (c) details (d)

The calculation can be changed with the addition of functions on the ratio and the difference between  $B_l$  and  $P_o$  as shown in table I. If  $S_D$  and  $S_R$  are set to be the identity functions, the equations in Table 1 become equivalent to the equations described in scenarios  $P_o < B_l$  and  $P_o > B_l$ . The restrictions for the functions used by  $S_D$  and  $S_R$  are: their values must be 0 when the input is 0 to make the function continuous when  $P_o = B_l$ ; their values must be 1 when the input is 1; they must be non-decreasing functions and odd.

TABLE I. CALCULATION OF ENHANCED PIXEL VALUE

if $P_o < B_l$	if $P_o > B_l$
$D_1 = S_D(B_l - P_o)$	$D_2 = S_D(B_l - P_o)$
$P_e = \left(1 - S_R\left(\frac{D_1}{B_l}\right)\right)B_g$	$P_e = 1 - \left(1 + S_R\left(\frac{D_2}{1 - B_l}\right)\right)(1 - B_g)$

In this work  $S_D$  and  $S_R$  are set to the S-shape function described by equation 1, as it makes entries close to 0 smaller, while entries close to 1 greater. Changing  $p$  makes the algorithm more ( $p < 1$ ) or less ( $p > 1$ ) sensitive. The default values for  $p$  used are 0.6 and 0.7 for  $S_D$  and  $S_R$ , respectively.

$$S(x) = \text{sign}(x) \times (0.5 - 0.5 \times \cos(|x|^p \pi)) \quad (4)$$

The scheme presented herein for pixel enhancement generalizes the one of WhiteboardIt, which assumes that the global background color is pure white and that  $S_D$  is equal to the identity function, as no function was used to modify the difference. Although, WhiteboardIt uses the S-shape function described by equation 1 for finding  $S_R$ . By making the global background color to pure white, WhiteboardIt applies the case where  $P_o < B_l$ . WhiteboardIt does not mention the condition  $P_o > B_l$  as it is not needed for yielding a pure white global background. In reference [4] the image enhancement algorithm presented here was successfully extended to work with document images.

## V. RESULTS

This section provides some results which confirm the effectiveness of the new color board enhancement algorithm in Tableau. They are compared with Qipit® [13], ScanR® [14], ClearBoard Demo 2.0.2 [15], and WhiteboardPhoto Demo 1.3.0.5 [12], which are commercial tools for processing board images. The first two used to be free Internet services (Qipit shutdown, ScanR is paid with a free trial version) and work only for white boards at the time this research was done; the others are Desktop softwares which work only with green, black and white boards. All systems were tested with 23 images of black, green and white boards; these images were sent to the both Internet services on 4-Jan-2009.

Figures 4 and 5 show the results of processing two green boards. On both Desktop softwares, board color must be set explicitly, for ClearBoard a printer-friendly version was generated, for WhiteboardPhoto enhanced background was set to pure black. Figure 4 (c) and 5 (d)

show the processing provided by the method presented in this paper on the same image, where one may observe that the image enhancement made the board background color uniform and provides a board image closer to the actual board color without illumination interference.

Now, the results obtained with the new version of Tableau in whiteboard images are compared with existing tools which is summarized in Figure 6 including the previous version of Tableau [2]. ScanR did not process Figure 6 (a) due as its size is less than 1Mpixel. As one may observe, the algorithm proposed herein provided the best results of all the tested tools as witnessed by Figure 6 (g) and (h). In Figure 6 (g) the inferred background color was used, while in (h) the closest color from the list as global background was applied.

## VI. CONCLUSIONS

This paper presents a new and better version of Tableau, an efficient software environment to process teaching board images of any background acquired with portable digital cameras. As far as the authors are aware of, Tableau is the only general tool available in the academic literature for such purpose.

Current research in Tableau looks for ways to automatically edit and efficiently print the content of processed boards.

## ACKNOWLEDGMENT

The research reported herein was sponsored by an MCT-Brazilian Government R&D Grant and CNPq funding.

## REFERENCES

- [1] CIE. Improvement to industrial colour-difference evaluation. CIE Publication No. Central Bureau of the CIE, Vienna, 2001.
- [2] D. M. Oliveira and R. D. Lins. Improving the Border Detection and Image Enhancement Algorithms in Tableau. In: ICIAR 2008, LNCS 5112, pp. 1111-1121, Springer Verlag.
- [3] D. M. Oliveira and R. D. Lins. Tableau – Processing Teaching-board Images Acquired with Portable Digital Cameras. In: CBDAR, pp.79-86, IAPR, 2007.
- [4] D.M.Oliveira and R.D.Lins. A New Method for Shading Removal and Binarization of Documents Acquired with Portable Digital Cameras. CBDAR 2009, pp. 3-10., Barcelona, IAPR Press.
- [5] G. Sharma, W. Wu, E. N. Dalal. The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. Color Research & Applications 30 (1), p. 21–30. Wiley Interscience, 2005.
- [6] L. He, Z. Liu, Z. Zhang. Why take notes? Use the whiteboard system. Proceedings of ICASSP, pp. 776-779. April 2003.
- [7] Z. Liu, Z. Zhang, L. He, Whiteboard scanning and image enhancement. DSP, Volume 17, pp. 414-432. Issue 2 March 2007.

[8] Z. Zhang and L. He. Notetaking with a camera: Whiteboard scanning and image enhancement. In IEEE ICASSP, volume 3, pp. 533–536. Montreal, Quebec, Canada, May 2004.

[9] Z. Zhang et al. 2006. System and method for whiteboard scanning to obtain a high resolution image. U.S. Patent 7,119,816, filed March 31, 2003, issued October 10, 2006.

[10] Flickr. <http://www.flickr.com>

[11] Picasa. <http://picasaweb.google.com/>

[12] POLYVISION. Whiteboard Photo.

[http://www.touchboards.com/polyvision/whiteboard\\_photo.asp](http://www.touchboards.com/polyvision/whiteboard_photo.asp)

[13] Qipit®. <http://www.qipit.com>

[14] Scanr®. <http://www.scanr.com>

[15] SOFTTOUCHIT. ClearBoard.

<http://www.softtouchit.com/products/clearboard/index.html>.

[16] W3C. <http://www.w3c.org>

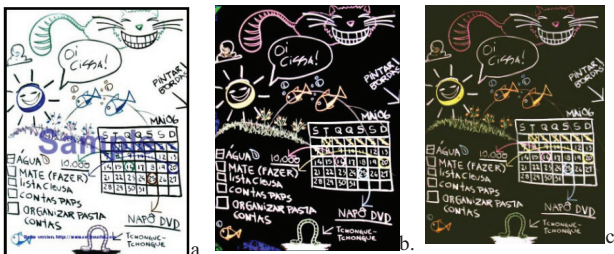


Figure 4. Fig. 3 results: ClearBoard (b), WhiteboardPhoto (c), new (d)

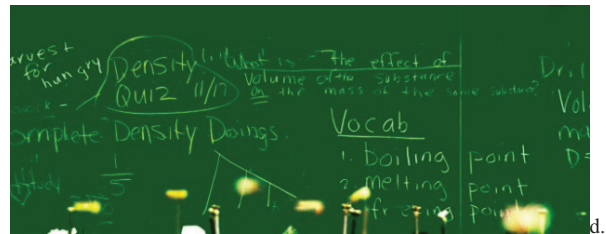
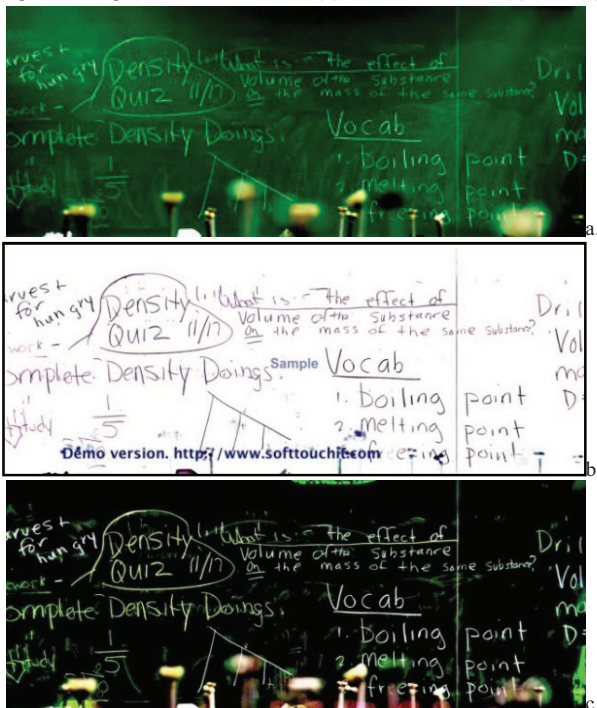


Figure 5. Original image (a), ClearBoard (d), WBPhoto (e), new (f)



Figure 6. Results: original (a); Qipit (b); ClearBoard® (c), Whiteboard Photo®(d), WhiteboardIt (e), Tableau v.1.0 [3] (f), new method with inferred background (g) and it set to closest color (h)

D. M. Oliveira e R. D. Lins. *A New Method for Shading Removal and Binarization of Documents Acquired with Portable Digital Cameras*. Em: Proceedings of Third International Workshop on Camera-Based Document Analysis and Recognition, pp.3-10, 2009.

# A New Method for Shading Removal and Binarization of Documents Acquired with Portable Digital Cameras

Daniel Marques Oliveira and Rafael Dueire Lins  
Universidade Federal de Pernambuco – Recife – PE – Brazil  
{daniel.moliveira, rdl}@ufpe.br

## Abstract

*Photo documents, documents digitized with portable digital cameras, often are affected by non-uniform shading. This paper proposes a new method to remove the shade of document images captured with digital cameras followed by a new binarization algorithm. This method is able to automatically work with images of different resolutions and lighting patterns without any parameter adjustment. The proposed method was tested with 300 color camera documents, 20 synthetic images with ground truth lighting pattern and grayscale images of the dataset of CBDAR 2007 dewarping contest. The results show that the new algorithm proposed here works in a wide variety of scenarios and keeps the paper texture of the original document.*

## 1. Introduction

Use of portable digital cameras for the digitalization of documents is a new research area. Such images, called *photo documents*, often have non-uniform shading, perspective distortion and blur. Fixing these distortions improves image readability and increases OCR accuracy rates. A new method for shading removal and binarization of photo documents is proposed in this paper.

For shade removal, Tan et al [12] proposes a scheme for scanned document images when the shading appears as a result of documents not being perfectly positioned on the scanner flatbed, due to book binding for instance. The variation in the illumination pattern can be modeled as a light source, which is found by a Hough transform. Reference [13] identifies document boundaries and assumes that the document has a rectangular shape; this *a priori* knowledge is used to remove image warp and shading.

Several papers in the literature attempt to remove shading by using 3D model obtained with special setup such as in [1], these alternatives are not portable.

Non-uniform shading requires adaptative binarization, first approaches use a sliding window around every image pixel to measure pixel threshold. Niblack's approach [8] uses equation (1) to calculate

the pixel threshold, where  $m$  and  $s$  are the mean and the standard deviation of pixels in the window around the current pixel, and  $k$  is a constant value in  $[-1,0)$ . It was improved by Sauvola [10] using equation (2) with same  $m$  and  $s$ ; constant  $k$  set in  $(0,1)$  which controls the standard deviation contribution to the threshold; constant  $R$  is set according to image contrast.

$$T = m + k \times s \quad (1)$$

$$T = m \times \left[ 1 + k \left( \frac{s}{R} - 1 \right) \right] \quad (2)$$

Reference [4] estimates the document background by a polynomial surface approximation of grayscale images. The surface is used to remove shading followed by binarization using a global threshold. This idea is also used in method proposed herein, with the advantage of shading removal of true color images.

## 2. The proposed method

For document images, two assumptions can be made [12]: the book surface has lambertian reflection (i.e. the specular index is too low) and it has non-uniform albedo distribution. With these assumptions, the image of the paper background has a constant value of intensity ( $I_{us}$ ), independent of the location of the viewer if illuminated by the same amount of light. As the image is the result of reflected light ( $I_o$ ) at arbitrary levels, one can express light variance by ratio (3):

$$\frac{I_o(i, x, y)}{I_{us}(i, x, y)} = L(i, x, y); \quad i \in \{r, g, b\} \quad (3)$$

The goal of the proposed method is to calculate the value of  $I_{us}(i, x, y)$  for all pixels in an image by assuming that the document background (paper) has one predominant color which is not uniform in the image due to illumination variance.

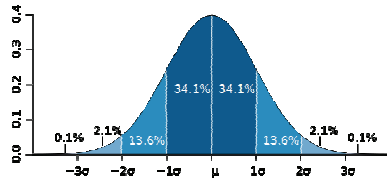
### 2.1. Narrow Gaussian Blocks

When observing a small area of the document one may notice that when this area belongs to the document background the histogram on all components have a "Narrow Gaussian shape" as depicted in the Figure 1.a. As other elements of the document are present, the histogram starts to be spread as can be seen on Figure

1.b. The relative area of Gaussians distributions in the interval  $[-\Delta\sigma + \mu, +\Delta\sigma + \mu]$  is shown in Figure 2.



**Figure 1.** Small blocks and their histograms: Background block (a); Block with "A" letter (b)

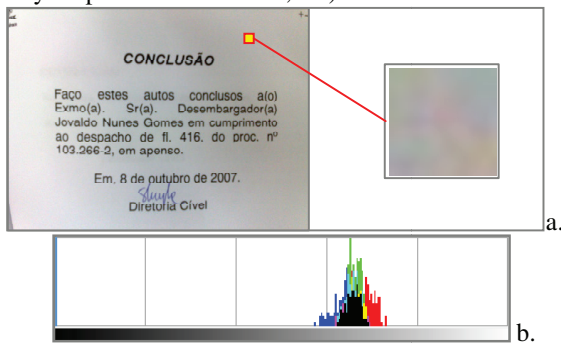


**Figure 2.** Gaussian distribution area relative to  $\sigma$  [16]

The aim here is to identify which areas belong to the document background by verifying if it has a histogram close to a Narrow Gaussian form. The choice of a block size must consider:

- There should be enough pixels for a statistical representation.
- If it is too large, will have no uniform color
- Its size should be less than the spacing between lines

A block area of 15x15 pixels was proven to be enough for all the test images used. In order to estimate the parameters for a Narrow Gaussian block (NGB), an analysis was done with Nokia N95 image (Figure 3.a), which was considered as the worst case due to the presence of blur and other noises (salt-and-pepper, Bayern pattern smoothness, etc).



**Figure 3.** Worst case scenario: whole image with highlighted block (a); histogram of the block (b)

The use of the mode rather than the mean for the Gaussian center is more straightforward as it is less affected by noisy values [5], thus modes of Figures 1.a and 1.b are the same.

Observing the worst case image, it was empirically found that the uniform areas have at least 75% of pixels within the area defined by  $\pm 6$  pixels around the

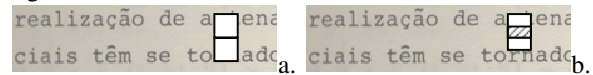
component mode (with component values in  $[0,255]$ ). Then a process to identify a NGB block is outlined as follows.

1. Compute histogram of each component value
2. Identify the histogram modes of each component:  $mode_r$ ,  $mode_g$  and  $mode_b$ .
3. Count the number of pixels in the interval  $(mode-6, mode+6)$ . If this count in all components is more than 75% of the number of pixel in the block, then the area is identified as NGB.
4. The background value is set to be equal to the value of the pixel with the smallest value using eq. (4), which is an approximation to the Euclidean distance in the RGB space.

$$D(c) \cong |c_r - mode_r| + |c_g - mode_g| + |c_b - mode_b| \quad (4)$$

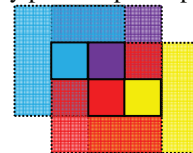
## 2.2. Defining block area

Ideally, every image pixel should be evaluated to identify if its surrounding pixels have a uniform color. This can be done by a sliding window around every pixel and using the procedure in section 2.1, although this is not feasible. On the other hand, the image could be divided into non-overlapping blocks of sizes 15x15 pixels. The main drawback of this approach is that the text spacing is between 15 and 25 pixels for high resolution documents, and blocks may not located, necessarily, in the middle of text lines as illustrated on Figure 4.a.



**Figure 4.** Example of location of blocks

A third approach is proposed by dividing the image into 5x5 blocks. An "expanded" 15x15 area centered on each 5x5 block is used for histogram calculation as depicted in Figure 5. This increases the probability of the expanded block to be located between text characters as shown on Figure 4.b. The computational complexity is  $9 \times W \times H$ , as each pixel is read  $(3 \times 3)$  times due to every pixel to participate in 9 blocks.



**Figure 5.** Four 5x5 pixels center boxes with a window of size 15x15 pixels

Then, to find the NGBs the following steps are executed:

1. Split the image into small blocks of 5x5 pixels.
2. Identify which blocks have expanded version with Narrow Gaussian histograms (as described in 2.1).

### 2.3. Finding background blocks

Now, the gathered information is a set of NGBs. The aim is to split into NGBs regions with little color variation; one of these will be the document background as will be described in this section.

The main criterion is to put together pairs of NGBs neighbors with background component-wise difference of its color less than a threshold (*Tneighbor*) into the same region. If this threshold is set too low it can yield a false-negative classification of similar neighbors. Otherwise, the threshold is set if too high it blurred text areas can be set as part of document background. This work assumed *Tneighbor* set to 5. Considering the component values in [0,255], the flood-fill algorithm with the whole procedure is showed bellow:

```

B: Set of all NGBs
Q: Queue used by to flood-fill
N: Closest NGBs used
Tneighbor: Threshold for BDmax value
BDmax(b1,b2): max(|b1.red-b2.red|,|b1.green-
b2.green|,|b1.blue-b2.blue|)
for every element b of B do
  If element b was not visited then
    Q <= {b};
    while Q is not empty
      q = first(Q);
      q visited state is set true;
      Q = Q - {q};
      N <= NGBs of every 8-direction
      closest to q
      Remove visited NGBs from N
      For every element n of N
        If (BDmax(n,q) < Tneighbor) then
          Q = Q + {n};
        end if
      end for
    end while
  end if
end for

```

Once the regions are identified, it is needed to select the one that represents the document background. As no assumptions are made about the image structure, two criteria are used to select the region as the document background: the percentage of NGBs of the region; quantity of NGBs in the center of the image, which is defined as a rectangle with dimensions  $\left(\frac{img\_width}{3}, \frac{img\_height}{3}\right)$  centered on the intersection of the diagonals of the image.

If there exists a region with more than 15% of all NGBs and most blocks in the rectangle of image center, this region is set as document background seed (DBS). Otherwise, the region with most NGBs is set as the DBS.

The computational complexity for the first part was found to be proportional to the number of blocks as each NGB are visited exactly one time. Considering memory usage, in the worst case the queue contains all the blocks in the image. A more efficient flood-fill and

component labeling procedures could also be used but they are more complex to implement, an example can be found in [7]. Region statistics for the second part can be computed while executing pseudo-code presented, which has a computational complexity proportional to the number of blocks.

### 2.4 Interpolation of non-background blocks

Once the DBS is found, the color values of the non-DBS blocks must be estimated. These NGBs are arbitrarily located, so a classical interpolation method (bilinear, bicubic etc) cannot be used.

A new approach is proposed similar to iterative dilation; where at every iteration the background regions are expanded, the color of the blocks are computed by the weighted average of its neighboring blocks. This process stops when all blocks have their color defined. The pseudo-code with this process is presented below; Figure 6 illustrates an example of it.

```

BGs: set of background with color set
B: set of currently expanding blocks
Q: set of expanding blocks of next iteration
BGs <= all blocks in DBS
//initial fill of B set
for every element n of BGs do
  for every m 8-neighbor of n and not in BGs
    B <= B + {m};
  end for
end for
//algorithm iteration
do
  Q <= empty set;
  for every element b of B do
    Set color value of b to weighted sum
    of BGs 8-neighbors, where the weight
    equals neighbor distance inverse;
    For every n that are 8-neighbor of b
      and is not in Q or BGs
        Q <= Q + {n};
    end for
  end for
  BGs = BGs + B; B <= Q;
until B is empty

```

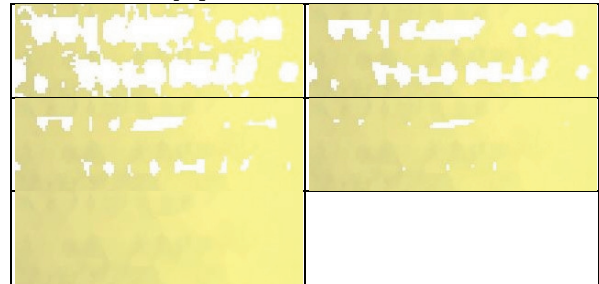


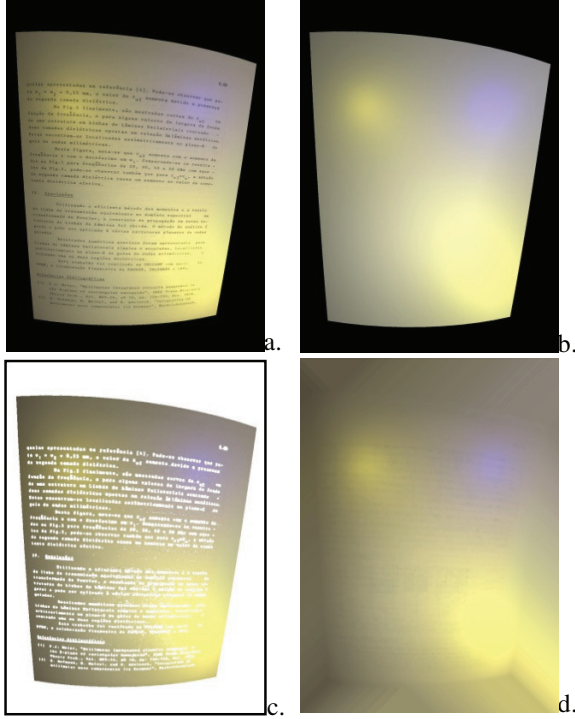
Figure 6. Interpolation iterations process

Observe that each non-DBS block is filled once, thus the computational complexity is proportional to the number of blocks.

Figure 7.a presents a synthetic image, Figure 7.b its lighting pattern ground truth. The result of the



processing described in section 2.3 is presented in Figure 7.c, with pure white blocks as non-DBS blocks; Figure 7.d shows the predicted background, note that the background is estimated for the whole image.



**Figure 7.** Background estimation: synthetic image (a); ground truth (b); DBS (c); estimated (d)

### 3. Shading Removal

Observing equation (3),  $I_{us}(i, x, y)$  values for the document background color should be constant for every RGB component, hence  $I_{us}(i, x, y) = I_{us}(i)$ . It is estimated by calculating the component-wise mean of the DBS blocks and locating the closest DBS block to this mean.

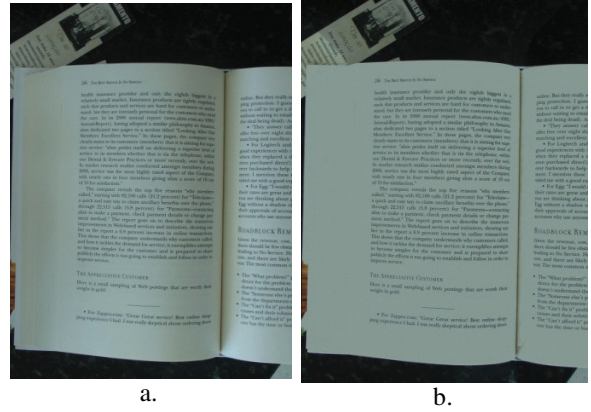
The only unknown value is  $I_{us}(i, x, y)$  for every pixel in the image. It is calculated using equation (5) or (6), where  $BG_o(i, x, y)$  and  $BG_{us}(i)$ , denotes  $I_o(i, x, y)$  and  $I_{us}(i, x, y)$  of estimated document background, respectively;  $I_{us}$ ,  $I_o$ ,  $BG_{us}$  and  $BG_o$  are in  $[0, 255]$ .

Eq. (5) is only applied for the case when  $I_o(i, x, y)/BG_o(i, x, y)$  is less than 1. Whenever greater, the ratio is difficult to represent as it is in  $(1, \infty]$ , negatives are used instead, so  $\overline{I_o(i, x, y)/BG_o(i, x, y)}$  is in  $[0, 1)$ . When  $I_o(i, x, y) = BG_o(i, x, y)$ , both eqs. (5) and (6) yields  $BG_{us}(i)$ , thus it is used in this case. No floating point is required as numerator may be computed first followed by an integer division.

$$I_{us}(i, x, y) = \frac{BG_{us}(i)}{BG_o(i, x, y)} I_o(i, x, y) \quad (5)$$

$$I_{us}(i, x, y) = \left( \frac{BG_{us}(i)}{BG_o(i, x, y)} I_o(i, x, y) \right) \quad (6)$$

$$\overline{C(i, x, y)} = 255 - C(i, x, y) \quad (7)$$



**Figure 8.** Shade removal: original image (a); result (b)

### 4. Document Binarization

Once shade is removed, binarization can be done by a global threshold approach [4]. Although, by applying it to the grayscale enhanced image purely yields poor results as camera documents may contain undesired objects around document disrupting the histogram, which is input for global threshold algorithms.

In Figure 7(c) the non-DBS blocks (in white) can be separated in two types: text blocks (TB) and image bordering blocks (IBB). Regarding these categories one may see that most TBs are surrounded by DBS blocks gathered on section 2.3. A process to identify IBBs is described in the pseudo-code below, and it can be summarized by a search starting at every block in the image border towards the opposite direction looking for a DBS block, until it is not found the intermediary blocks are set to be IBBs. Observe that the computational complexity in the worst case is proportional to the number blocks, but in the cases where the document borders are close to the image borders the complexity approaches the sum of the dimensions of the image.

```

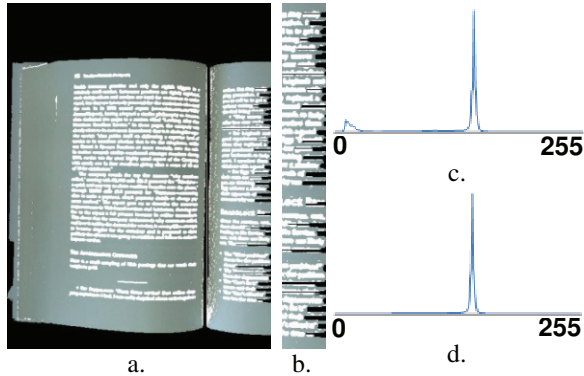
I: Number of block columns in the image
J: Number of block lines in the image
B: bidimensional array of image blocks. B(0,0)
  means upper left block; B(I,J) lower right
Initially all blocks are NOT IBBs
for every i varying from 1 to I
  execute findBorderBlocks(i, 0, right);
  execute findBorderBlocks(i, J, left);
end for
for every j varying from 1 to J
  execute findBorderBlocks(0, j, down);
  execute findBorderBlocks(I, j, up);
end for
procedure findBorderBlocks(i, j, dir)
  //while stops when first DBS block is found
  while ((B(i, j) is not DBS block)

```

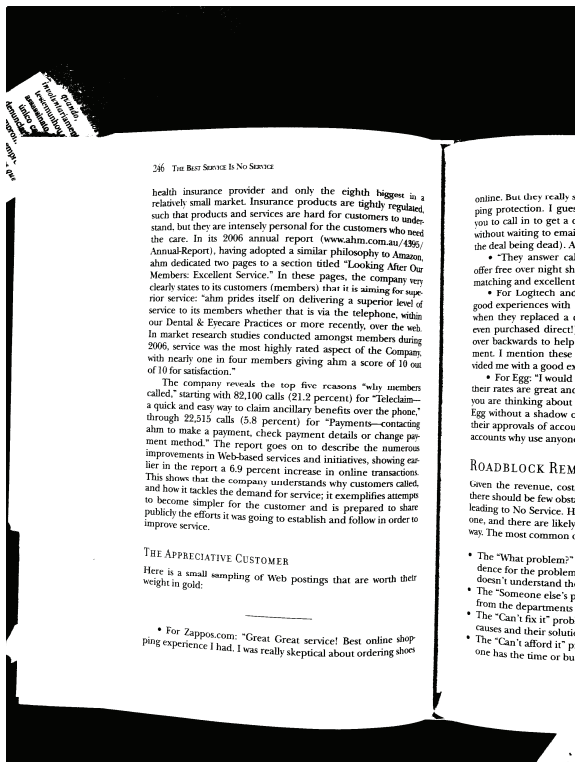
```

B(i,j) is set as IBB
//move i or j to given direction
if dir = left then j = j - 1;
if dir = right then j = j + 1;
if dir = up then i = i - 1;
if dir = down then i = i + 1;
end while
end procedure

```



**Figure 9.** Rough boundary definition: block classification (a); some misclassification (b); enhanced image histogram (c); modified histogram (d)



**Figure 10.** Proposed binarization using Otsu's

Figure 9.a shows the rough estimation with right border details on Figure 9.b, where IBBs are in black, TBs are in white and DBS blocks with its corresponding color. It is observable that the boundary estimation can be set to some text blocks as image

border, but it does not affect the global threshold performance as this fact is not statistically significant. Figure 9.c shows the histogram of the grayscale version of the whole enhanced image showed in Figure 8.b, where Figure 9.d shows the histogram without IBBs. Figure 10 shows Figure 8.a's binarized version.

## 5. Results

To compare the method proposed here, Savoula [10] approach was implemented with [2] optimizations. The memory cost is measured by the second moment of the whole image with two 64-bit arrays of size  $W \times H$  are allocated at once. Parameters were adjusted for the test images of this work yielding to a  $21 \times 21$  window,  $k = 0.2$  and  $0.5$  and  $R$  set to 128.

The method presented in reference [4] could not be implemented due to its large computational time of 5 Mpixel images polynomial surface calculation, with 5 million points interpolation.

The complexity of the proposed work was outlined in all sections to be proportional to the image dimensions or the number of blocks. The latter depends on the block area and its expansion factor. Needed memory for the execution is at least  $(3 \times W \times H + k \times W \times H / \text{block\_area})$  bytes: one 24-bit array for the enhanced image; an array of size  $W \times H / \text{block\_area}$  with block information and queue lists with at most  $W \times H / \text{block\_area}$  elements (where  $k$  block information size plus queue element, our implementation required  $k$  equal to 30).

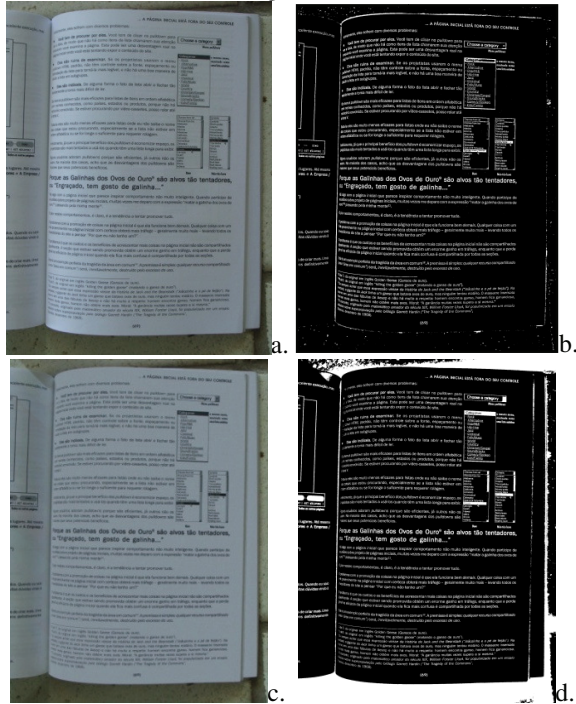
Table 2 shows the processing time statistics using Java 1.5, DELL D531; Turion TL56 1.80GHz; 3Gb RAM on Windows Vista Business. The first column presents mean and standard deviation of images sizes in Mpixels followed by the total processing time average in ms. Other columns show the percentages of the execution times for each part; Section 3 part and the modified histogram computation were implemented together. Times do not include file loading and screen refresh. Floating points were only used on global thresholds, as section 2.4 neighbor distances can only be 1 or  $\sqrt{2}$ , an integer approximation was used.

**Table 2 – Processing time statistics**

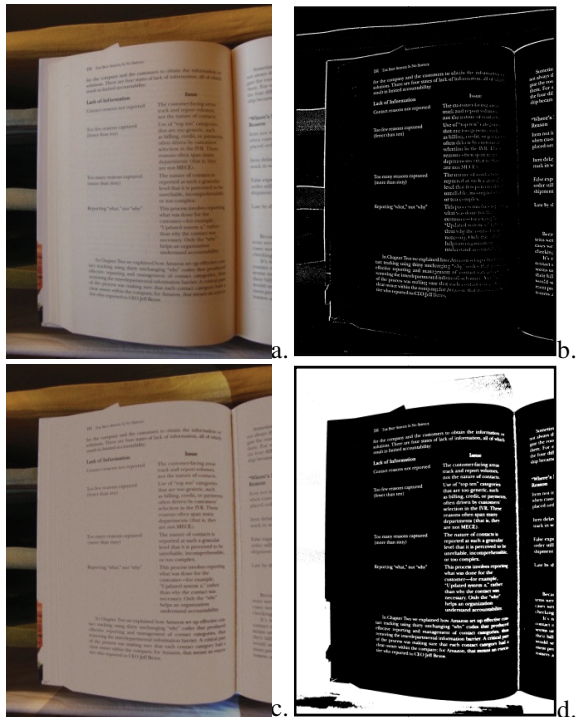
	Size	Time (ms)	Sec. 2.2	Sec. 2.3	Sec. 2.4	Sec. 3 + modified histo.	Otsu
Mean	6.69	4325	69.3%	3.9%	2.6%	23.4%	0.8%
Std.	1.82	1096	1.7%	1.1%	0.7%	1.4%	0.1%

A visual evaluation was performed with 300 photo documents. Figure 11 shows that the results with both binarizations (b and c) had similar outcome. Figure 11.c shows shading completely removed. While Figure

12.a Savoula's algorithm does not binarizes properly where in 12.c new method does. Removed shade version is illustrated in Figure 12.c.

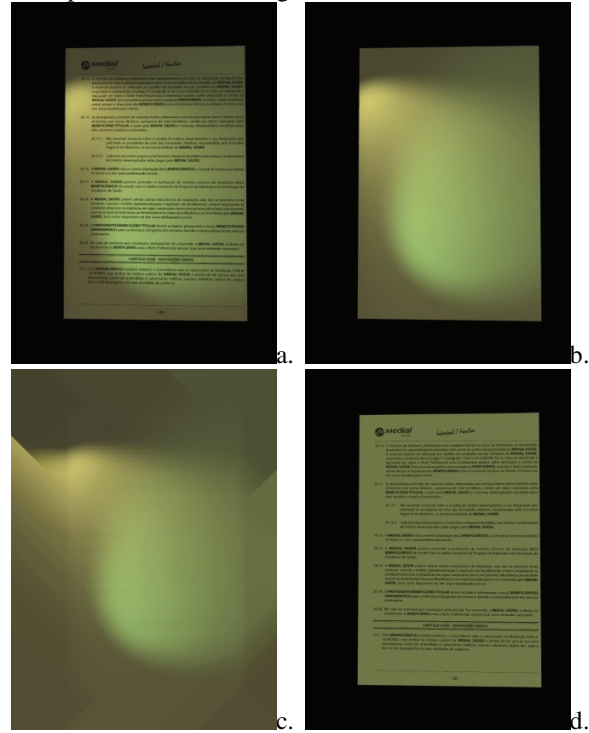


**Figure 11.** Comparison: original (a); Savoula (b); new shade removal (c); new binarization with Otsu (d)



**Figure 12.** Processing results: original (a); Savoula (b); new shade removal (c); new binarization with Otsu (d)

A quantitative comparison of the estimated lighting pattern with its ground truth was done with 20 synthetic images covering different scenarios generated by Adobe After Effects CS4 [14]. The mean error was about 0.0566 and its standard deviation of 0.0024, with component values scaled in the interval [0,1]. Examples are shown in Figures 7 and 13.



**Figure 13.** Background estimation: synthetic image (a); ground truth (b); predicted (c); shading removed (d)

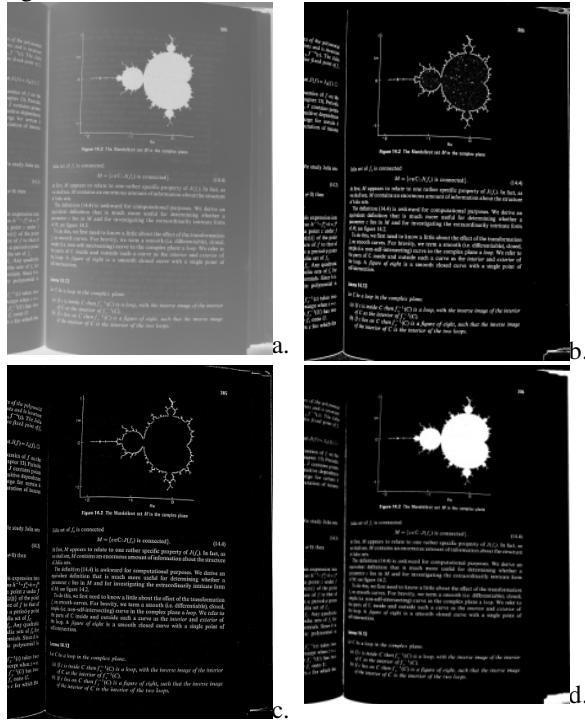
The binarization was compared with CBDAR 2007 dewarping contest [3], as they contain a 102 grayscale images and the corresponding binary image which it was used as a ground truth, this comparison is more straightforward than OCR as no dewarping is done for the binary image. Three global threshold algorithms were used: Otsu's [9], Mello-Lins [6] and Silva et al [11]. All those algorithms were tested with the original and enhanced image with the modified histogram computation described in section 4. Savoula's [10] adaptative approach was also compared. Four different metrics were calculated, the same metrics were used in DIBCO 2009 contest [15]:

- Error (E) =  $(FP + FN)/TOTAL$
- Recall (RC) =  $TP/(FN+TP)$
- Precision (PR) =  $TP/(FP+TP)$
- F-measure (FM) =  $2*RC*PR/(RC+PR)$

Where FP denotes false positives, FN false negatives, TP true positives, TN true negatives and TOTAL the number of pixels in the analyzed area.

Notice that for all presented metrics, greater values means better performance, except for error rate.

It was found that in CBDAR2007 dewarping dataset, 3 binary images (img\_1179, img\_1203, img\_1235) has a region with uniform black color classified as white pixels, one example is shown in Figure 14.



**Figure 14.** CBDAR2007 image img\_1179: original (a); binarized (b); Savoula (c); proposed binarization with Otsu (d)

Another issue regards document boundary, both ground truth and Savoula's approach set to white pixels external to the document. These pixels are marked as "don't care" pixels, as is not important whether they is classified as white or black due to they not belonging to text area in the document. For the algorithm comparison, the documents were cropped to encompass only the text information.

Table 3 shows the statistical metrics for the global threshold algorithms for the original image and the proposed modification of the histogram calculation. The metric shows a performance improvement when the proposed approach is used. Among all global threshold methods, Otsu's showed to provide more consistent good results than the others.

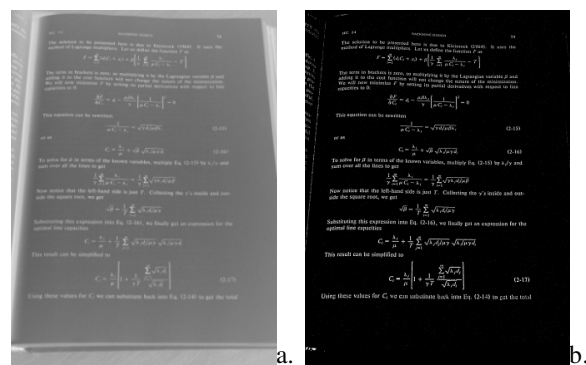
Table 4 shows the metrics with Savoula's approach using original image. The performance of the proposed method yielded better results than Savoula's and handle a wider range of "scenarios" as the one showed in Figure 12. Another result can be seen in Figure 15.

**Table 3 – Comparison between binarization approaches using the original images and the proposed approach**

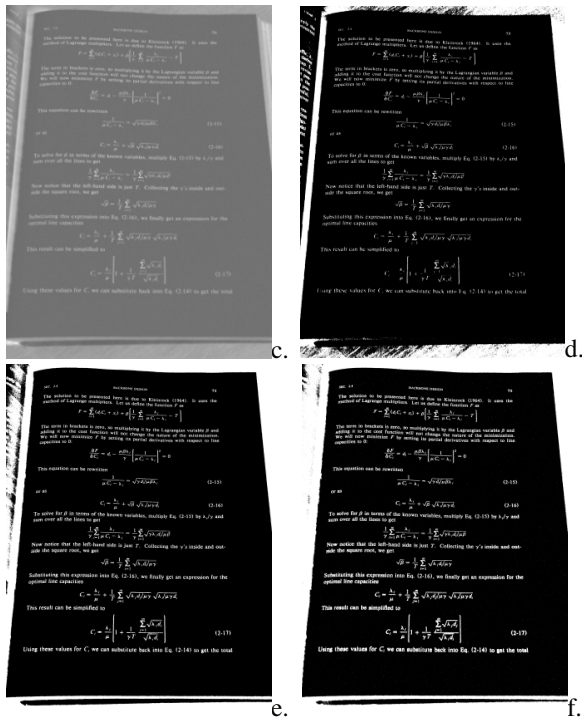
		Min	Max	Mean	Std.
Otsu (original)	E	0.5%	35.8%	3.9%	5.1%
	RC	55.7%	100.0%	82.5%	9.4%
	PR	2.7%	100.0%	80.1%	25.1%
	FM	5.3%	94.1%	77.6%	18.3%
Otsu (proposed)	E	0.2%	5.4%	1.4%	0.8%
	RC	69.5%	99.7%	83.1%	5.5%
	PR	72.8%	100.0%	95.9%	6.2%
	FM	77.2%	94.7%	88.8%	3.4%
Silva-et-al (original)	E	0.6%	9.0%	3.2%	1.6%
	RC	28.7%	100.0%	69.4%	16.0%
	PR	11.6%	100.0%	86.1%	19.9%
	FM	20.7%	91.4%	73.5%	13.1%
Silva-et-al (proposed)	E	0.3%	10.5%	2.4%	1.5%
	RC	37.0%	100.0%	77.5%	16.2%
	PR	23.7%	100.0%	89.9%	16.5%
	FM	38.3%	97.1%	80.4%	10.8%
Mello-lins (original)	E	0.9%	98.5%	61.9%	35.1%
	RC	52.8%	100.0%	98.8%	5.1%
	PR	1.5%	99.7%	18.5%	22.4%
Mello-lins (proposed)	E	0.1%	93.6%	19.5%	29.3%
	RC	66.7%	100.0%	98.2%	5.1%
	PR	5.7%	99.9%	56.8%	31.5%
Mello-lins (proposed)	E	10.7%	97.6%	65.8%	28.3%
	RC				
	PR				

**Table 4 – Savoula's approach Metrics**

		Min	Max	Mean	Std.
Savoula k=0.5	E	17.2%	47.4%	23.2%	5.0%
	RC	38.6%	96.3%	70.8%	13.6%
	PR	2.5%	34.6%	18.9%	7.0%
	FM	4.9%	48.9%	29.2%	9.4%
Savoula k=0.2	E	0.4%	4.4%	1.7%	0.7%
	RC	60.8%	91.2%	78.5%	6.8%
	PR	75.5%	99.1%	94.9%	4.0%
	FM	71.4%	94.9%	85.7%	4.7%



**Figure 15.** dsc00626 Binarization: original (a); Savoula (b)



**Figure 15. (cont.)** without shading (c); new binarization with Otsu (d); new binarization with Mello-Lins (e); new binarization with Silva et al (f)

## 6. Conclusions and Lines for Further Work

This paper showed new schemes for color shading removal and binarization of documents captured with portable digital cameras. The performance of the shading removal algorithm was compared the lighting pattern with ground truth and with more than 300 images, which provided good and fast (does not require floating point operations) results for a wide variety of scenarios using only the captured image. The output of the binarization algorithm introduced was compared with one of the most used local algorithm with the same 300 images and CBDAR2007 dewarping dataset, it proves to cover more scenarios than Sauvola's method.

Some of the images in CBDAR 2007 dewarping dataset exhibit a light back-to-front interference [6][11] (bleeding). The binarization of photo documents with strong back-to-front interference is left a line for further work.

## 7. Acknowledgements

Research reported herein was partly sponsored by CNPq – Conselho Nacional de Pesquisas e

Desenvolvimento Tecnológico, Brazil. We want also to thank Rafael Lellys for providing synthetic model.

## 8. References

- [1] M. S. Brown et al. "Restoring 2D Content from Distorted Documents". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007. pp 1904-1916.
- [2] F. Shafait, D. Keysers, and T. M. Breuel, "Efficient implementation of local adaptive thresholding techniques using integral images," in *Document Recognition and Retrieval XV*, San Jose, USA, Jan. 2008.
- [3] F. Shafait and T. M. Breuel. "Document Image Dewarping Contest", 2nd Int. Workshop on Camera-Based Document Analysis and Recognition, CBDAR 2007, Brazil. Sep. 2007. pp 181-188.
- [4] S. Lu and C. L. Tan. Binarization of Badly Illuminated Document Images through Shading Estimation and Compensation. *ICDAR 2007*, Volume 1, 23-26 Sept. 2007, Brazil, Curitiba, 2007, pp 312 – 316.
- [5] R. A Maronna, D. R. Martin and V. J. Yohai. *Robust Statistics: Theory and Methods*. John Wiley & Sons, Ltd, England, 2006. ISBN: 0-470-01092-4.
- [6] C. A. B. Mello and R. D. Lins. "Image segmentation of historical documents". *Visual 2000*, Mexico City, Mexico. 2000.
- [7] W. Kesheng, O. Ekow, and S. Arie, "Optimizing connected components labeling algorithms," in *SPIE Int. Symposium on Medical Imaging*, San Diego, CA, USA, Feb. 2005.
- [8] W. Niblack. *An Introduction to Digital Image Processing*. Prentice-Hall, Englewood Cliffs, New Jersey, 1986.
- [9] N. Otsu. A threshold selection method from graylevel histogram. *IEEE Transactions on System, Man, Cybernetics*, 19(1):62–66, January 1978.
- [10] J. Sauvola and M. Pietikainen. Adaptive document image binarization. *Pattern Recognition*, 33(2):225–236, January 2000.
- [11] J. M. M. da Silva, R. D. Lins and V. C. da Rocha Jr., "Binarizing and Filtering Historical Documents with Back-to-Front Interference". *Proceedings of SAC 2006*. New York : ACM Press, 2006. p. 853-858.
- [12] C.L. Tan, L. Zhang, Z. Zhang and T. Xia. Roberts, "Restoring Warped Document Images through 3D Shape Modeling", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, New York, Vol. 28, No. 2, Feb. 2006, pp. 195-208.
- [13] Y.C. Tsoi and M.S. Brown. "Geometric and Shading Correction for Images of Printed Materials: A Unified Approach Using Boundary". *Proc. IEEE CVPR* 2004, vol. 1, pp. 240-246, 2004.
- [14] Adobe. Adobe After Effects CS4. <http://www.adobe.com/products/aftereffects/>.
- [15] B. Gatos. DIBCO 2009 – Evaluation. <http://www.iit.demokritos.gr/~bgat/DIBCO2009/Evaluation.html>, accessed on 1<sup>st</sup> may of 2009.
- [16] J. Kemp. File:Standard deviation diagram.svg. [http://en.wikipedia.org/wiki/File:Standard\\_deviation\\_diagram.msvg](http://en.wikipedia.org/wiki/File:Standard_deviation_diagram.msvg)

D. M. Oliveira e R. D. Lins. *Shading Removal of Camera Documents with Natural Neighbor Interpolation*. (Submetido em 10 de Junho de 2011 ao IJDAR).

# Shading Removal of Camera Documents with Natural Neighbor Interpolation

Daniel Marques Oliveira · Rafael Dueire Lins

**Abstract** Portable digital cameras have become one of the most widely used devices for document acquisition due its low price, portability and image quality. Often, photos of documents have non-uniform illumination causing shading which may yield to bad quality image for human visualization and unsuitable for some image processing algorithms, such as global binarization ones. The non-uniformity may be circumvented by some methods using a window around every pixel that estimates its predominant color. This approach may not work well with documents with large non-uniform regions (e.g. photographs, illustrations). The algorithm proposed here removes the shading of this kind of photographed documents printed in non-glossy paper yielding better quality images. First, it identifies the background pixels. Then, the shading for non-background pixels are estimated using Natural Neighbor Interpolation. The proposed algorithm was tested on 33 synthetic images using SSIM and PSNR as performance measures. Several other “real world” images were also tested in order to confirm the algorithm effectiveness both in quality and processing time.

## Keywords

Shading removal, Illumination normalization, Document enhancement.

## 1 Introduction

Portable digital cameras were developed for “family” photographs. Their fast decreasing price, high portability, and rapid increase in image quality made of them a pervasive good in modern life. Such availability gave rise to an originally unforeseen application: document digitalization.

The development of algorithms for photographed textual documents is more complex than the ones aimed at processing document images captured with flatbed scanners, due to uneven illumination caused by the short distance between the document and the camera strobe-flash and all the uncontrolled sources of lightning in the environment, the perspective projection introduced by holding the camera in a non-orthogonal plane to the document, variable image resolution as the distance between the camera and the document also varies, and irregular illumination.

Figure 1 presents two examples of photographed documents which encompass drawings and pictures, in which one may observe some shades. Due to its intrinsic uneven nature, shading removal is not a straightforward process. One strategy is to see the original document image as the result of the addition of two images. One of them is a blank sheet of paper in which the variation of intensity is due exclusively to the non-uniform illumination, called the background image. The second image is a uniformly illuminated image which is added to the background one. The idea of shading removal is to get rid of the background image yielding only the uniformly illuminated one,

which would resemble the scanned version of the physical document.

Often, the methods in the literature identify shading of the whole document image followed by its removal. The estimation of the shading can be classified into: shading interpolation and shading fitting. For the former approach, the resulting estimation passes exactly in the known points. For the shading fitting process, the final estimation is close to the known shading values. The next subsections overview these techniques.

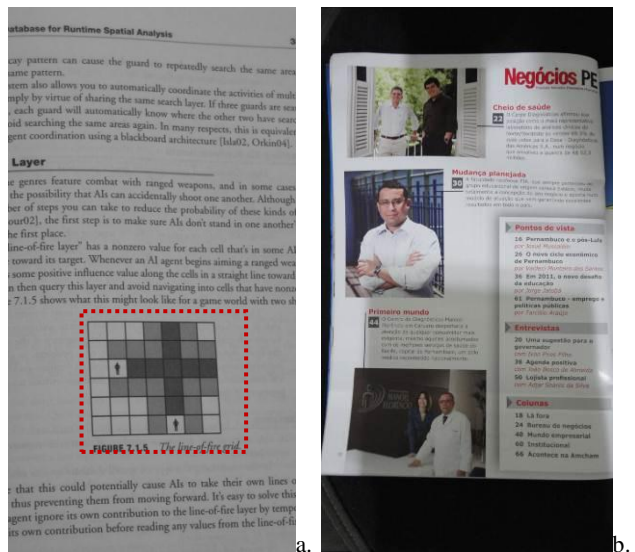


Figure 1. Camera documents with drawings and pictures

### 1.1 Shading Interpolation

The technique of shading interpolation was proposed by Gatos and his colleagues in reference [9] as a binarization algorithm for grayscale historical documents. The algorithm steps follow:

1. Apply a low-pass Wiener filter to remove the noisy background and increase the contrast between the background and the character strokes.
2. Identify the background regions with Niblack’s binarization algorithm [13].
3. Calculate the surface on the foreground region by interpolating neighboring background intensities.
4. Apply thresholding by combining the calculated background surface with the original image.
5. Perform a post-processing step that removes the small connected components and fills small gaps on the foreground.

The interpolation in step 3 is described by equation (1), where  $I(x, y)$  is the image obtained in step 1,  $N(x, y)$  is the Niblack’s binarization of  $I(x, y)$  and  $B(x, y)$  the interpolated background image. The window of size  $dx \times dy$  should cover at least two characters [9].

Daniel Marques Oliveira · Rafael Dueire Lins  
Universidade Federal de Pernambuco, Recife, Brazil  
+55 81 2126-8430 ext. 4305  
e-mail {daniel.moliveira, rdl}@ufpe.br

$$B(x, y) = \begin{cases} I(x, y) & \text{if } N(x, y) = 0 \\ \frac{\sum_{i=x-dx}^{x+dx} \sum_{j=y-dy}^{y+dy} I(i, j)(1 - N(i, j))}{\sum_{i=x-dx}^{x+dx} \sum_{j=y-dy}^{y+dy} (1 - N(i, j))} & \text{if } N(x, y) = 1 \end{cases} \quad (1)$$

In reference [10], Gatos and colleagues substitute Niblack's by Savoula's binarization algorithm [20].

Another approach which uses shading interpolation was proposed by the authors of this work in reference [16]. It differs from Gatos *et al* [9] [10] approaches by the use of overlapping blocks, instead of pixelwise analysis. It takes into account that the number of points in the image is at the order of millions (in an image of several Mpixels). The overlapping window allows a coarser and faster image processing technique, without losing the continuity. This approach reduces the number of processing points by a factor of the size of the block area.

Another difference between Gatos *et al* approaches [9] [10] and the one described in reference [16] regards the background identification process that is divided into two steps:

1. Identify blocks with uniform color by a histogram analysis;
2. Define the uniform region which characterizes the document background by neighborhood block comparison

The first item has the purpose to eliminate the blocks that do not belong to the paper background. The second step does a neighbor block comparison instead of a set of blocks (e.g. border detection). The work [17] extends the one in [16] to black, green and white boards.

The shading of the foreground is obtained by an iterative procedure that starts on the boundary of the background identified at step 1 and stops when no background color is left to be estimated.

## 1.2 Shading Fitting

Zhang and his colleagues [26] identify the background region by edge detector and morphological operations on the value component of the HSV model. The Radial Basis Function (RBF) is used to estimate the shading of the entire image.

Lu and Tan [23] estimate the shading by applying iteratively the Savitzky-Golay filter, which is a local polynomial fitting. Lee and his collaborators [27] remove shading of grayscale document images with pictures and illustrations obtained by scanner. First, they obtain background and foreground using Söbel edge detector in several directions. Then, the background values for the foreground regions are estimated by vertical line segments. Finally, the shading is removed from the original image. This method is further improved by Athimethphat and Patanavijit [28] with the addition of a polynomial fitting.

This paper presents an efficient algorithm to remove shading of color document images with pictures, drawings and photos such as the one in Figure 1.b. It extends the work presented by the authors of this paper in reference [16] by changing the shading estimation of the foreground to the Natural Neighbor Interpolation (NNI). This type of interpolation is claimed by Amidror [2] to be the best method when the distribution of known points varies highly, which is the case of the illustrations boundaries in the image of a document.

This paper is organized as follows: section 2 gives the background needed for understanding the proposed algorithm. Section 3 describes the new algorithm. Section 4 shows test

results of the approach, while section 5 presents the conclusions of this work.

## 2 Background

In order to present the new algorithm, z-order filling curve and natural neighbor interpolation are explained in this section. Both of them are of great importance to the quality and computational performance of the algorithm proposed.

### 2.1 Z-order pixel iteration

A spacing filling curve maps an interval into an N-dimensional cube without missing any point in the cube. These filling curves are defined by an order function between two points. The z-order filling curve is used herein [1]. This path resembles a "Z" letter which can be seen in various dimensions (see Figure 2.a). Reading points in this order has the advantage of increasing memory locality [6].

The z-order defines a z-code (see Figure 2.b) that is obtained by interleaving bits of the Cartesian coordinates (see Figure 3.b for a 2D bit interleaving). Thus, the z-code is a projection mapping of N-dimension to one-dimension. If the code is used as a key of a tree, one can obtain a self-balancing multi-dimensional data structure [1].

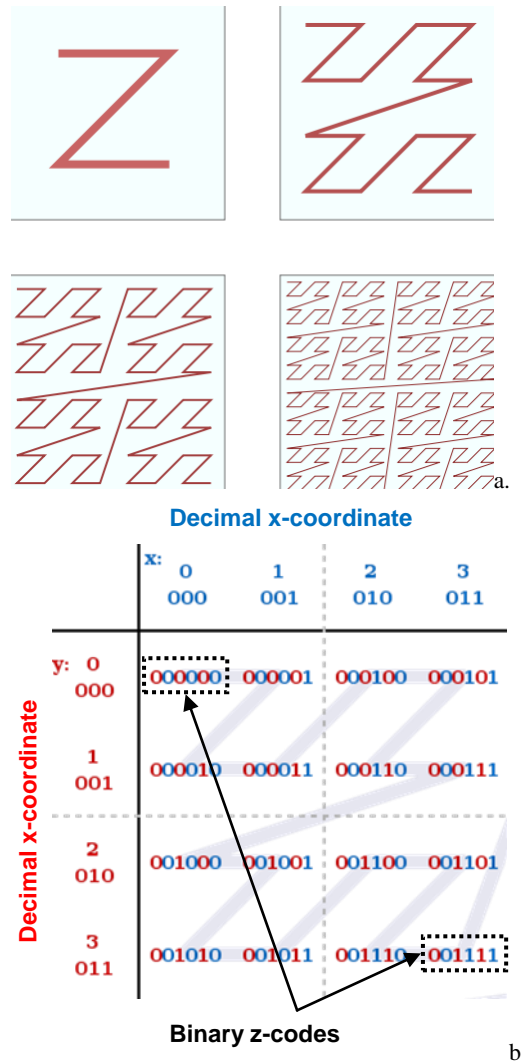
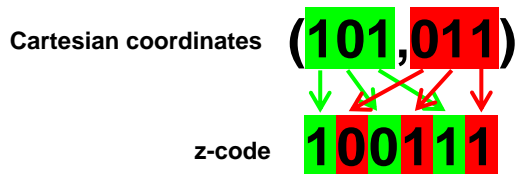
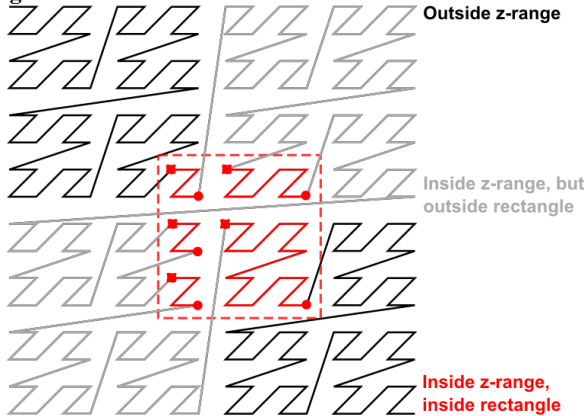


Figure 2. Z-order filling curve: several levels [30] (a); decimal coordinates and binary z-codes [31] (b)





**Figure 3.** Conversion between Cartesian coordinates to z-code.



**Figure 4.** Search interval in red with squares and circles as begin and end of z-segments; segments not in the rectangle are in black (outside z-range) and gray (inside z-range)

One may notice that first and last points of the z-shape define a square. As the proportion of most images dimensions is not 1:1, a procedure must be defined in order to iterate the pixels within a rectangle in the z-order.

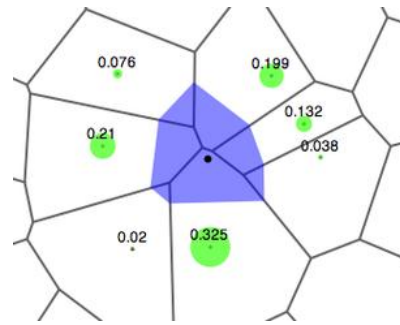
Tropf and Herzog [24] provide an algorithm with that purpose. Their paper does not provide enough information to allow the reader to implement their procedure. Ramsak [18] [19] presents another algorithm for the same operation, which is also poor in details. This paper suggests one implementation of such methods which may be found at the publisher website.

## 2.2 Natural Neighbor interpolation

The estimation of values between non-uniform points is known as “scattered data interpolation”. Amidror [2] provides a survey on this subject and compares methods as Radial Basis Function (used by Zhang and his colleagues [26]), polynomial fitting (exploited by Athimethphat and Patanavijit [28]), natural neighbor interpolation and others. That study claims that the most efficient approach when the density of points has a high variation is the natural neighbor interpolation.

The first step of natural neighbor interpolation is to setup a Voronoi diagram of the known points. A Voronoi diagram defines areas of the closest points. Figure 5 shows one example with solid black lines as region limit between sites.

An unknown value will be calculated by the weighted sum of neighboring sites. The weight is the area “borrowed” from each known point if the diagram had this point. Figure 5 represents this by the blue region around the new point; the interpolation weights are represented by the green circles at each corresponding point.



**Figure 5.** Natural neighbor interpolation [32]

The natural neighbor interpolation provides continuity everywhere except on known sites, therefore the interpolation is C0 continuous. If the first derivative is known for the sites, is possible to provide a C1 interpolation. Thus, if the N-derivatives are known for the points, is possible to obtain a CN continuous interpolation. This study will use only C0 interpolation.

The dual of a Voronoi Diagram is the Delaunay Triangulation [3]. In this triangulation no point is inside the circumcircle of any triangle. The Voronoi diagram is obtained by connecting the circumcircle centers of the triangles.

Delaunay triangulation can be used to compute the Natural Neighbor Interpolation; insertions and removals in it are easier to implement than using Voronoi Diagram directly [8].

The triangulation is built as points are inserted. If points are inserted according to their z-order, the triangulation creation has linear complexity in time [5]. If no specific order is used, the lower bound in time is log linear [3].

Natural neighbor interpolation can only be done inside the convex hull defined by Delaunay triangulation. In order to estimate the values outside the hull, [4] the extrapolation method is used.

## 3 Shading removal

Figure 1 presents two documents: the left one is a grayscale image obtained on CBDAR 2007 dewarping dataset [21]; the other was taken with an amateur Sony Cybershot W210 camera. As the only input for this method is the document image, is reasonable to assume that:

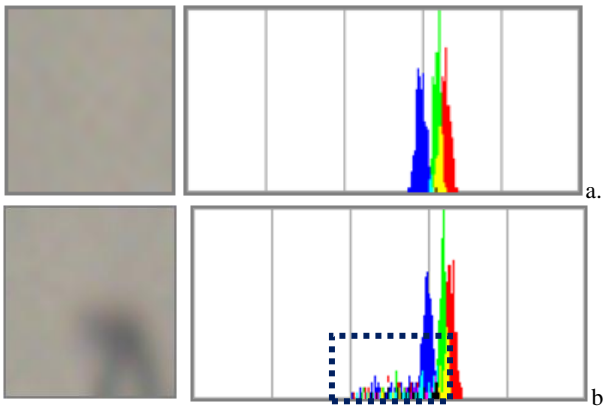
- The paper has a uniform color;
- There is a high contrast between the foreground and background.

The proposed method is summarized below and is presented in the next sections.

1. Recognize blocks with uniform color (BUC);
2. Identify paper BUCs;
3. Estimate the paper color of the foreground blocks using Natural neighbor interpolation.

### 3.1 Blocks with uniform color (BUC)

Blocks located on paper background will present histogram components concentrated around paper intensities. Figure 6.a shows the histogram of a block of paper background pixels only. Figure 6.b presents the color histogram of the block between the paper background and upper part of letter “A”.



**Figure 6.** Paper background histogram with each color representing the respective RGB component: block totally located on background (a); block partially on background(b)

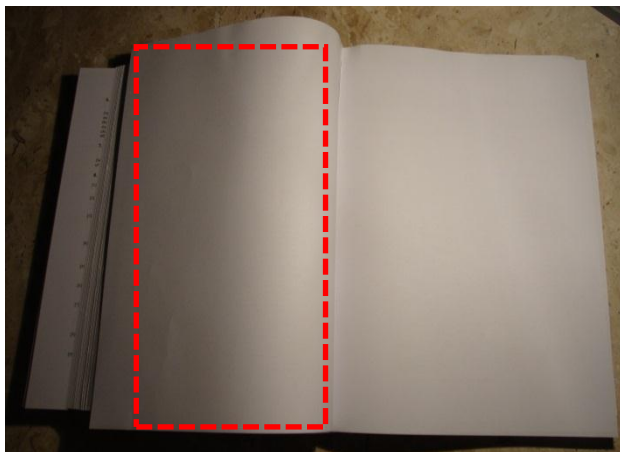
Reference Maronna *et al* [12] claims that the presence of noise affects more strongly the linear statistical measures (mean, standard deviation, etc) than the robust ones (mode, median, etc). Figure 6.b shows that the modes did not changed in presence of letter "A" upper part. Therefore, the modes are used to represent the uniform color of a block.

In order to study how the illumination changes in a document with uniform color, blank sheets of white papers were placed in the middle of the volume 1 of the proceedings of ICDAR 2007. Six pictures were taken with a Sony Cybershot W210 (amateur digital camera) and Nokia N95 (cell phone). Three different setups were done with: natural illumination ("natu" prefix); incandescent ("incan" prefix) and fluorescent ("flour" prefix) lights. These setups can be seen in Figure 8.

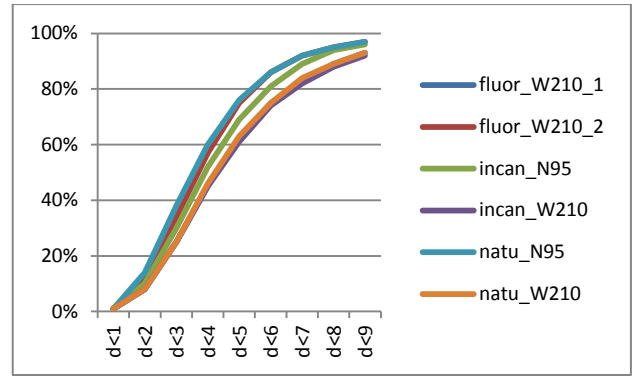
In order to analyze the worst case illumination, the regions with greater variance were selected, e.g. Figure 8 left rand side. The selected image regions were divided into blocks of 17x17 pixels. For each block, the mode was computed in all components; then the Tchebychev distance is calculated between each pixel in the block and the mode as in equation (2).

$$d(\mathbf{R}, \mathbf{G}, \mathbf{B}) = \max(|\mathbf{R} - \mathbf{R}_{mode}|, |\mathbf{G} - \mathbf{G}_{mode}|, |\mathbf{B} - \mathbf{B}_{mode}|) \quad (2)$$

Figure 8 shows a graph with the percentage of pixels that the distance in equation (2) is less than the value specified in the x-axis, with the color values specified in the [0,255] interval. Each line in the graph identifies the experiment done in a different document.



**Figure 7.** Blank sheet of white paper sample region (incan\_W210 picture)



**Figure 8.** Percentage of pixels in blocks that have Tchebychev distance to the block mode less than the value specified in the x-axis

Two parameters are defined: DBUC and PBUC. The first is a threshold of  $d(\mathbf{R}, \mathbf{G}, \mathbf{B})$ . The second is a limit of the percentage of pixels in a block that  $d(\mathbf{R}, \mathbf{G}, \mathbf{B}) < DBUC$ . These two parameters are used to define if a block has uniform color or not, as defined in Listing 1.

**Listing 1**

```

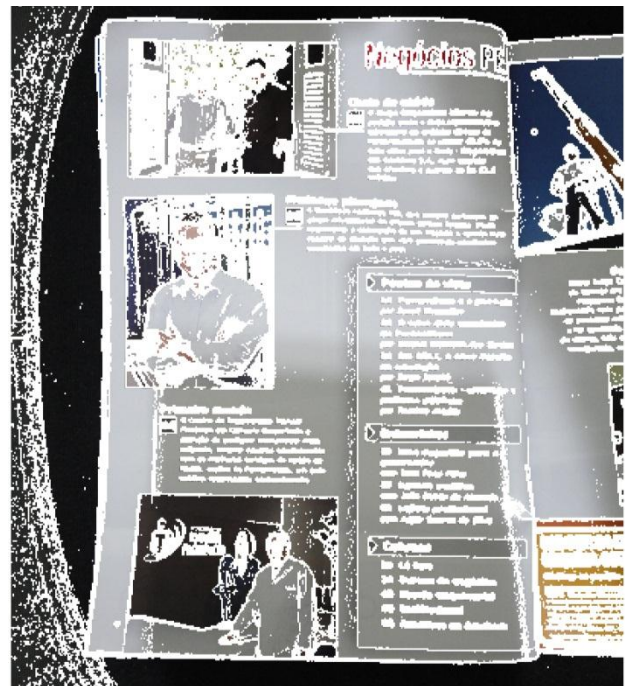
Input: B - A block
Output: Has uniform color or not

FUNCTION IS_BUC(B)
RETURN # of pixels in B with  $d(\mathbf{R}, \mathbf{G}, \mathbf{B}) < DBUC$  > PBUC
# of pixels in B

END FUNCTION

```

Observing the graph in Figure 8, two reasonable values for DBUC and PBUC are 5 and 62%, respectively. Figure 9 shows the BUCs identified with the color defined by the block mode while the non-uniform blocks are presented in pure white.



**Figure 9.** Blocks with uniform colors (BUC) (non-uniform colors blocks in white)

### 3.2 Identifying paper BUCs

Once the BUCs are defined, not all blocks belong to the paper background. Two k-neighbors BUCs of the same region (paper, picture, graph etc.) have a low contrast. This relation is defined by if the Tchebychev distance of the colors of two blocks (described in equation (3)) is below than the DBUC threshold. Then, if two k-neighbors BUCs have this relation, they belong to the same region. This rule is applied until there are no more blocks to process. The region with most elements is returned as the paper background region. In order to identify the neighbors, the all-k-nearest neighbors [6] algorithm was used herein with the value of K set to 4. **Listing 2** summarizes this procedure and Figure 10 shows an example of such estimation.

$$DIST(A, B) = \max_{k \in \{red, green, blue\}} |A_k - B_k| \quad (3)$$

#### Listing 2

Input: Blocks with uniform color  
Output: Set of blocks of document background

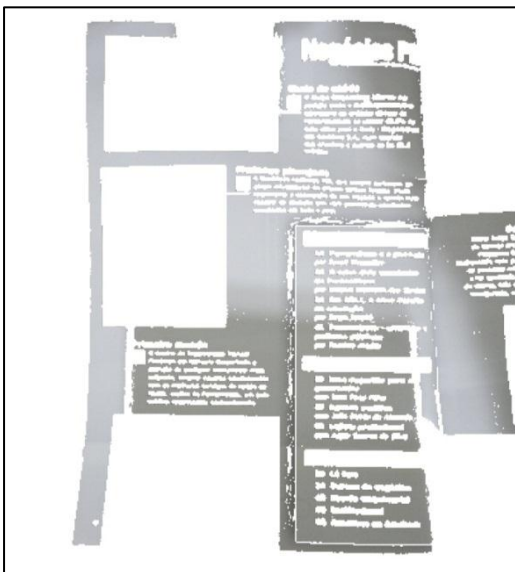
```
FUNCTION GET_MAIN_CLUSTER(BUCS)

Calculate k-nearest neighbor of BUCs [6];
CLUSTERS = empty set;

WHILE
exists a B ∈ BUCS that does not belong to any
set in CLUSTERS
DO
QUEUE = [B];
C = empty set;
WHILE QUEUE is not empty DO
B = first element of QUEUE;
QUEUE = QUEUE - {B};
C = C + {B};
FOR each neighbor N of B DO
IF DIST(N, B) ≤ DBUC THEN
Append N to QUEUE;
END IF
END FOR
END WHILE
END WHILE

RETURN the set in CLUSTERS with largest
cardinality;

END FUNCTION
```



**Figure 10.** Blocks identified as paper background (non-paper blocks are in white)

### 3.3 Shading estimation for non-background blocks

The paper background blocks were obtained in the previous step. One may notice in Figure 10 that there is a high density variation of background blocks close to the illustrations. Document images has several Mpixels (i.e. millions points), which is a severe scenario for the estimation of unknown values. As presented earlier on in this paper, the Natural Neighbor Interpolation is the best estimation method for such case [2], thus it is used herein.

As explained in section 2, before calculating the natural neighbor a Delaunay triangulation is built. The points are inserted in the z-order, thus the computation is linear in time [5]. Figure 11 shows the shading estimated using Figure 10 as basis.



**Figure 11.** Shading pattern blocks estimated using natural neighbor interpolation

### 3.4 Shading Removal

Once the shading is obtained for the entire picture, it must be removed. Two assumptions can be made for document images [23]:

- the book surface has lambertian reflection (i.e. the specular index is too low), and
- it has non-uniform albedo distribution.

With these assumptions, the image of the paper background has a constant value of intensity ( $I_{us}$ ), independently of the location of the viewer if illuminated by the same amount of light.

As the image is the result of reflected light ( $I_o$ ) at arbitrary levels, one can express the light variance by equation (4):

$$\frac{I_o(i, x, y)}{I_{us}(i, x, y)} = L(i, x, y); \quad i \in \{r, g, b\} \quad (4)$$

Observing equation (4),  $I_{us}(i, x, y)$  the values for the document background color should be constant for every RGB component, hence  $I_{us}(i, x, y) = I_{us}(i)$ . It can be set two ways:

- Assuming as being a particular color, most times this value is set as pure white [7].
- Inferring the uniform paper value from the image. Herein a component-wise mean was calculated of the paper cluster. The color of a block closest to the mean is set as  $I_{us}(i)$ .

The value without shading is  $I_{us}(i, x, y)$  for every pixel in the image. It is calculated using equation (5), where  $BG_o(i, x, y)$  and  $BG_{us}(i)$ , denotes  $I_o(i, x, y)$  and  $I_{us}(i, x, y)$  of shading pattern, respectively.  $I_{us}$ ,  $I_o$ ,  $BG_{us}$  and  $BG_o$  are in the interval  $[0,255]$ .

$$I_{us}(i, x, y) = \frac{BG_{us}(i)}{BG_o(i, x, y)} I_o(i, x, y) \quad (5)$$

Figures 12 and 13 present the result of correcting images with final background set to pure white and inferred paper value, respectively.



Figure 12. Shading removing setting  $I_{us}(i)$  to pure white



Figure 13. Shading removing inferring  $I_{us}(i)$  from the image

## 4 Results

The results are divided in two sections. The first shows the objective performance metric comparison using synthetic images. The second shows some examples of processing in real world images. All tests were done using a Dell Inspirin 1525 Laptop with 3GB RAM Intel Celeron 550 2.0GHz running in Windows 7 professional.

### 4.1 Tests with Synthetic Images

The algorithm performance measurement is done using synthetic images. These were generated by three illumination models combined with 11 images without shading. For each output image, one with the same structure but without distortion was created. Figure 14 illustrates how these images were produced.

This work used Processing 1.2.1 [33] library to build this test set. All images will be made publically available for non-commercial use after this paper publication.

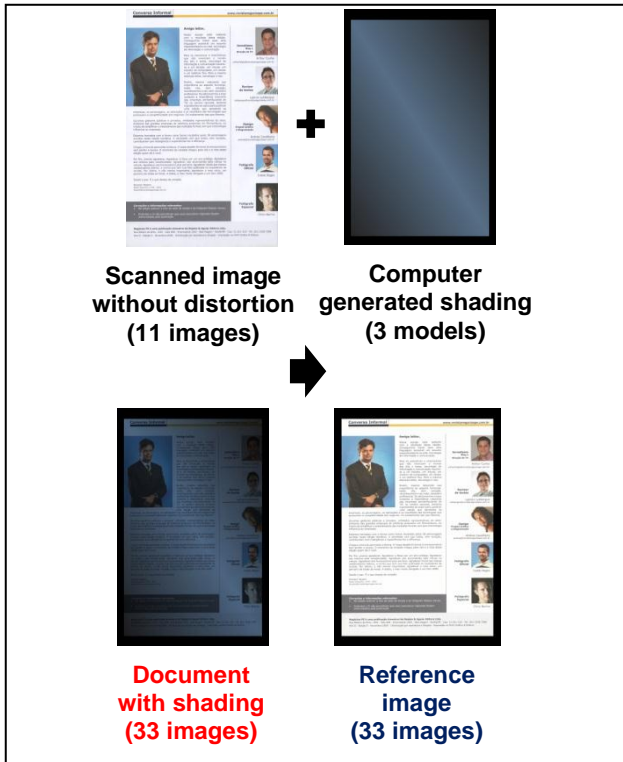


Figure 14. Computer generated images

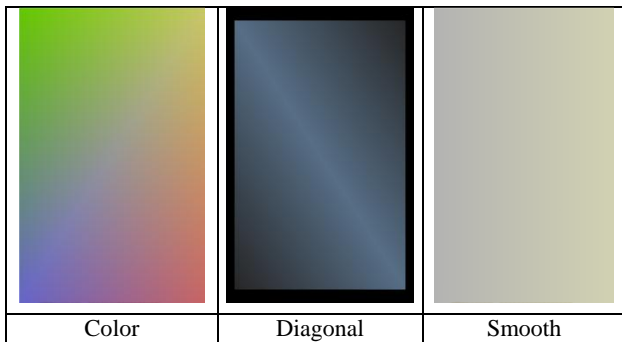


Figure 15. Illumination models with different shading levels

Figure 15 shows the illumination models cover various scenarios. “Color” is seldom seen in real images, but it is used to test the robustness of the algorithm. The “Diagonal” model represents when the illumination which is not orthogonal to the plane of the paper. The “Smooth” illumination model provides a soft change in shading.

The images without shading were obtained from the LiveMemory dataset [11] and two issues of the Negócios PE Magazine [14][15], which are scanned images. The colors of the paper were extracted as they are not pure white in the RGB space. Figure 16 shows some of these images which range from textual documents to ones that has small text areas; paper color for each picture are below the respective image.

The distorted and reference images were produced by applying the pattern with and without shading to the corresponding image, respectively.

Small text areas Paper color #FFFFFF	Small images Paper color #EEE7D5
Single image Paper color #F8F8FF	Images without borders Paper color #F9F9FF
Grayscale Paper color #F5F4F9	Text and old paper Paper color #EEE7D5

Figure 16. Scanned images with uniform illumination

#### 4.1.1 Quality measures

The last section explained how the test database was built. The quality of the proposed method is obtained by calculating the “similarity” between references and processed images. This computation alone does not provide enough information regarding the algorithm performance, as the processed image can be less clear than its input. If this happens, the image information was “lost”. Thus, the degree of “similarity” between input and reference image is obtained, then the percentual gain can be calculated.

Figure 17 illustrates how the “similarity” of input and processed images is obtained related to its reference. Equation (6) shows

the gain percentage; herein it is assumed that the measure is higher when two images are more equal. Therefore, if the processed image is closer to the reference one than its input, then the gain will be positive.

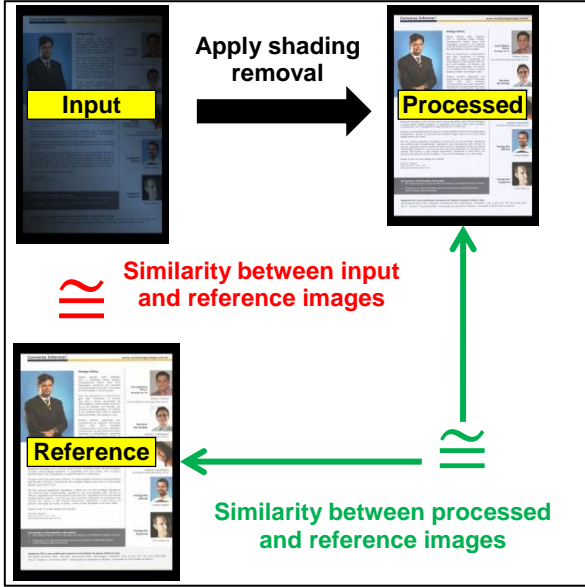


Figure 17. “Equalities” computation

$$\text{Gain} = \frac{\text{Similarity between processed and ref.}}{\text{Similarity between input and ref.}} - 1 \quad (6)$$

In this work two similarity measures were used: PSNR and SSIM [25]. The PSNR (Peak signal-to-noise ratio) is a classical similarity measure between two images which is calculated according to equation (7), where  $MAX_I$  are the maximum possible intensity and  $MSE$  the mean square error between the two images. In this context, the intensities are in the [0;1] scale, thus  $MAX_I$  is equal to 1.

$$\text{PSNR} = 20 \times \log_{10} \left( \frac{MAX_I}{\sqrt{MSE}} \right) \quad (7)$$

The SSIM was proposed by Wang and his colleagues, which stands for Structural SIMilarity [25]. It is an objective measure that compares how two images are different from a human eye perceptual model. It is widely used for JPEG compression quality measurement. This measures ranges between 0 and 1, with 1 representing if two images are equivalent and 0 if they are totally different. This work used the Matlab code provided by the authors [25] with the default parameters.

As the input images are in color, the values of the PSNR and SSIM were computed for each RGB component and the grayscale version of the images.

Table 1 shows the SSIM measure between the input and reference images. The minimum and maximum values of the measure as well its average and standard deviation are presented. Table 2 provides the same information for the PSNR metric. Both tables show that “Diagonal” and “Color” models changed the structured of the document images. The input images tested cover a wide range of different scenarios of the SSIM distribution.

Table 1. SSIM statistics between input and reference images

Illumination model	SSIM		
	Min.	Avg. ± std.	Max.
Diagonal	0.471	0.572 ± 0.048	0.682
Color	0.714	0.814 ± 0.018	0.862
Smooth	0.932	0.951 ± 0.006	0.962

Table 2. PSNR statistics between input and reference images

Illumination model	PSNR		
	Min.	Avg. ± std.	Max.
Diagonal	4.1	5.742 ± 1.504	8.2
Color	6.4	9.320 ± 1.628	12.1
Smooth	12.5	14.813 ± 1.653	17.2

Table 3 provides the statistics of the measures of the SSIM and PSNR between the processed and reference images. The proposed algorithm yields same quality images independently of the level of shading.

Finally, Table 4 shows the average and standard deviation of the gain for the SSIM and PSNR.

Figure 18 provides examples of the processing of some of the test images with of the algorithm proposed in this paper. This figure includes the input, estimated shading and output images, together with the reference image and the calculated SSIM and PSNR for each RGB component.

It is possible to see that the proposed method estimates the shading in illustrations have smooth transitions according to their colors at the borders. The corrections in those areas are similar to the respective reference image, and are confirmed by the SSIM and PSNR values.

One important fact is that the illustrations do not change the quality of correction of textual areas as the interpolation always passes at the known points. This does not happen for the fitting methods, which are used by [26] [23] [27] [28].

Table 3. Quality measures of the proposed algorithm

Illumination model	SSIM avg. ± Std.	PSNR avg. ± Std.
Diagonal	0.952 ± 0.014	32.4 ± 8.6
Color	0.970 ± 0.011	33.8 ± 11.1
Smooth	0.981 ± 0.004	38.5 ± 5.1

Table 4. Quality gain provided by proposed algorithm

Illumination model	Avg. SSIM Gain ± Std.	Avg. PSNR Gain ± Std.
Diagonal	68.0% ± 13.5%	485.4% ± 270.4%
Color	19.5% ± 3.1%	268.7% ± 155.7%
Smooth	3.2% ± 0.6%	165.1% ± 47.6%

Input image	Estimated shading	Output image	Reference image	SSIM	PSNR
				R: 0.942 (+91.4%) G: 0.956 (+66.3%) B: 0.958 (+48.3%)	R: 37.1 (+800.2%) G: 39.0 (+711.0%) B: 39.4 (+632.5%)
				R: 0.944 (+99.3%) G: 0.950 (+71.9%) B: 0.940 (+51.1%)	R: 26.5 (+392.2%) G: 25.7 (+305.2%) B: 25.2 (+251.2%)
				R: 0.967 (+16.8%) G: 0.974 (+17.3%) B: 0.951 (+33.1%)	R: 38.1 (+214.6%) G: 38.9 (+238.6%) B: 28.9 (+188.3%)
				R: 0.978 (+2.2%) G: 0.981 (+2.3%) B: 0.978 (+3.9%)	R: 38.5 (+172.0%) G: 38.9 (+173.3%) B: 41.7 (+231.1%)

Figure 18. Shading and output images of the proposed algorithm

#### 4.1.1 Performance measures

Figure 19 shows the graph that relates the processing time and the image size. One may see that the relation is almost linear for the images in test set. Further testing with more complex documents that may incorporate many different figures of different sizes is needed. Whenever there is a large variation in the density of the points in an image the performance of Delaunay triangulation may degrade [29].

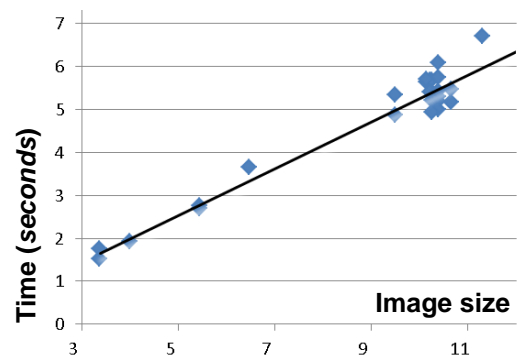


Figure 19. Image size (in Mega pixels) versus Processing Time (in seconds).

## 4.2 Processing “real-world” images

In this section, the efficiency in quality and processing time of the algorithm is presented for processing “real world” images. The images here were acquired with a Sony W380 portable digital camera. Figures 20 and 21 show images with shading followed by after their processing by the proposed algorithm. The worst processing time was on Figure 21 with 11.09s; the size of the input image is 14Mpixels.



Figure 20. Result: (a) input with 3240x4320 pixels; (b) output processed in 7.21s



Figure 21. Result: (a) input with 3240x4320 pixels; (b) result processed in 11.09s



## 5 Conclusions

This paper introduces a new algorithm that efficiently removes shading of documents acquired with portable digital cameras. The algorithm is robust enough to remove shading of documents with pictures, drawings and other illustrations. A widely used perceptual measure – the SSIM – shows that the produces high quality results. Besides that, the algorithm is computationally efficient.

## REFERENCES

- [1] Aluru S (2005). Quadrees and octrees. In D. Metha and S. Sahní, editors, Handbook of Data Structures and Applications, chapter 19. Chapman & Hall/CRC.
- [2] Amidror I (2002). Scattered data interpolation methods for electronic imaging systems: A survey. *J. Electron. Imaging* 11 (2): 157–176.
- [3] de Berg M, van Kreveld M, Overmars M, Schwarzkopf O (2008). Computational Geometry: Algorithms and Applications. Springer-Verlag.
- [4] Bobach T, Farin G, Hansford D, Umlauf G (2009). Natural neighbor extrapolation using ghost points. *Computer-Aided Design* 41(5): 350–365.
- [5] Buchin K, Mulzer W (2009). Delaunay Triangulations in  $O(\text{sort}(n))$  Time and More. 50th Annual IEEE Symposium on Foundations of Computer Science: 139-148.
- [6] Connor M, Kumar P (2010). Fast construction of k-nearest neighbor graphs for point clouds. *IEEE Transactions on Visualization and Computer Graphics* 99 (PrePrints). Source available on <http://sites.google.com/a/compgeom.com/stann/>.
- [7] Fan J (2009). Robust Color Image Enhancement of Digitized Books, Proceedings of 10<sup>th</sup> International Conference on Document Analysis and Recognition: 561-565.
- [8] Flötotto J (2011). 2D and Surface Function Interpolation. In CGAL User and Reference Manual. CGAL Editorial Board, 3.8 edition.
- [9] B. Gatos, I. Pratikakis and S.J. Perantonis, "An Adaptive Binarization Technique for Low Quality Historical Documents", DAS VI, LNCS 3163/2004, pp. 102-113
- [10] B. Gatos, I. Pratikakis and S.J. Perantonis, "Adaptive degraded document image binarization", *Pattern Recognition*, Vol. 39 (3), March 2006, pp. 317-327
- [11] Lins RD, Torreão GS, Silva GFP (2010). Content Recognition and Indexing in the LiveMemory Platform. *Graphics Recognition. Achievements, Challenges, and Evolution. LNCS 6020/2010*: 220-230.
- [12] Maronna RA, Martin DR, Yohai VJ (2006). *Robust Statistics: Theory and Methods*. John Wiley & Sons, Ltd, England. ISBN: 0-470-01092-4.
- [13] Niblack, W.: *An Introduction to Digital Image Processing*. Englewood Cliffs, N. J., Prentice Hall (1986) 115-116.
- [14] Najaim & Aguiar Ltd (2011). *Negócios PE – Empresas, Mercado, Empresários e Executivos*. 19<sup>th</sup> edition. Najaim & Aguiar Ltd, Recife, Brazil.
- [15] Najaim & Aguiar Ltd (2011). *Negócios PE – Empresas, Mercado, Empresários e Executivos*. 18<sup>th</sup> edition. Najaim & Aguiar Ltd, Recife, Brazil.
- [16] Oliveira DM, Lins RD (2009). A New Method for Shading Removal and Binarization of Documents Acquired with Portable Digital Cameras. *Third International Workshop on Camera-Based Document Analysis and Recognition*: 61–65, Barcelona, Spain.
- [17] Oliveira DM, Lins RD (2010). Generalizing Tableau to Any Color of Teaching Boards. 20<sup>th</sup> International Conference on Pattern Recognition, 2411-2414, Istanbul, Turkey.
- [18] Ramsak F, Markl V, Fenk R, Zirkel M, Elhardt K, Bayer R (2000). Integrating the UB-Tree into a Database System Kernel. *Proc. of VLDB2000*, Cairo, Egypt.
- [19] Ramsak F (2002). Towards a general-purpose, multidimensional index: Integration, Optimization, and Enhancement of UB-Trees. Ph.D. Thesis, Technische Universität München.
- [20] Sauvola J, Pietikainen M (2000). Adaptive document image binarization. *Pattern Recognition*, 33(2): 225–236.
- [21] Shafait F, Breuel TM (2007). Document Image Dewarping Contest. 2nd Int. Workshop on Camera-Based Document Analysis and Recognition: pp 181-188, Curitiba, Brazil.
- [22] Skopal T, Kratky M, Snasel V, Pokorny J (2006). A New Range Query Algorithm for the Universal B-trees. *Information Systems* 31(6): 489-511.
- [23] S. Lu and C. L. Tan. Thresholding of badly illuminated document images through photometric correction in *Proc. 2007 ACM Symp. Document Eng., Manitoba, Canada, 2007*, pp. 3–8.
- [24] Tropsch H, Herzog H (1981). Multidimensional Range Search in Dynamically Balanced Trees. *Angewandte Informatik (Applied Informatics)* 2: 71–77, Wiesbaden, Germany.
- [25] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* 13(4): 600-612. Available at <https://ece.uwaterloo.ca/~z70wang/research/ssim/>.
- [26] Zhang L., Yip AM, Tan CL (2007). Photometric and geometric restoration of document images using inpainting and shape-from-shading. 22nd Conference on Artificial Intelligence: 1121-1126, Vancouver, Canada.
- [27] Jung-San Lee, Chin-Hao Chen, and Chin-Chen Chang. 2009. A novel illumination-balance technique for improving the quality of degraded text-photo images. *IEEE Trans. Cir. and Sys. for Video Technol.* 19, 6 (June 2009), 900-905. DOI=10.1109/TCSVT.2009.2017314 <http://dx.doi.org/10.1109/TCSVT.2009.2017314>.
- [28] Athimethphat, M.; Patanavijit, V.. A non-linear illuminations balancing for reconstructed degraded scanned text-photo image. *ISCIT 2010*. 1158–1163. DOI=10.1109/ISCIT.2010.5665163.
- [29] S. Har-Peled. *Data structures for geometric approximation*. American Mathematical Society, 2011.
- [30] Eppstein D (2011). Four levels of the Z curve, showing the square that is eventually filled by the curve. Available at [http://en.wikipedia.org/wiki/File:Four-level\\_Z.svg](http://en.wikipedia.org/wiki/File:Four-level_Z.svg).
- [31] Eppstein D (2011). The Z plane-filling curve formed by interleaving the binary representations of the coordinates (x,y) and sorting the resulting interleaved binary numbers. Available at <http://en.wikipedia.org/wiki/File:Z-curve.svg>.
- [32] Markluffel (2011). File: Natural-neighbors-coefficients-example.png. <http://en.wikipedia.org/wiki/File:Natural-neighbors-coefficients-example.png>
- [33] Processing.org (2011). Available at <http://processing.org>.