UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE TECNOLOGIA E GEOCIÊNCIAS

DEPARTAMENTO DE ELETRÔNICA E SISTEMAS

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

RODRIGO BARROS BERNARDINO

ANÁLISE DE QUALIDADE E TEMPO DE PROCESSAMENTO DE ALGORITMOS DE BINARIZAÇÃO PARA DOCUMENTOS TEXTUAIS

Recife

2018

RODRIGO BARROS BERNARDINO

ANÁLISE DE QUALIDADE E TEMPO DE PROCESSAMENTO DE ALGORITMOS DE BINARIZAÇÃO PARA DOCUMENTOS TEXTUAIS

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

Área de Concentração: Comunicações

Orientador: Prof. Dr. Rafael Dueire Lins

Recife

2018

Catalogação na fonte
Bibliotecária Valdicéa Alves, CRB-4 / 1260

B523a Bernardino. Rodrigo Barros.
Análise de qualidade e tempo de processamento de algoritmos
de binarização para documentos textuais / Rodrigo Barros Bernardino - 2018.
109folhas, Il.; e Tabs.

Orientador: Prof. Dr. Rafael Dueire Lins.

Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia Elétrica, 2018.
Inclui Referencias e Apêndices.

1. Engenharia Elétrica. 2. Binarização. 3. Síntese de textura.
4. Processamento de imagens. I. Lins, Rafael Dueire (Orientador). II.Título.

UFPE

621.3 CDD (22. ed.) BCTG/2018 - 217

Universidade Federal de Pernambuco

Pós-Graduação em Engenharia Elétrica

PARECER DA COMISSÃO EXAMINADORA DE DEFESA DE DISSERTAÇÃO DO MESTRADO ACADÊMICO DE

RODRIGO BARROS BERNARDINO

TÍTULO

"ANÁLISE DE QUALIDADE E TEMPO DE PROCESSAMENTO DE ALGORITMOS DE BINARIZAÇÃO PARA DOCUMENTOS TEXTUAIS"

A comissão examinadora composta pelos professores: RAFAEL DUEIRE LINS, DEINFO/UFRPE; VALDEMAR CARDOSO DA ROCHA JÚNIOR, DES/UFPE E GABRIEL DE FRANÇA PEREIRA E SILVA, UACSA/UFRPE, sob a presidência do primeiro, consideram o candidato **RODRIGO BARROS BERNARDINO APROVADO**.

Recife, 08 de fevereiro de 2018.

MARCELO CABRAL CAVALCANTI Coordenador do PPGEE	**RAFAEL DUEIRE LINS** Orientador e Membro Titular Interno
GABRIEL DE FRANÇA PEREIRA E SILVA Membro Titular Externo	**VALDEMAR CARDOSO DA ROCHA JÚNIOR** Membro Titular Interno

AGRADECIMENTOS

Em primeiro lugar, agradeço ao professor Dr. Rafael Dueire Lins, que me orientou e apoiou durante todo o percurso. Serei eternamente grato por cada conselho, cada cobrança, a paciência e energia dedicadas a mim. O prof. Rafael faz jus ao posto de orientador e, ainda mais, de um verdadeiro educador. Sua cultura, conhecimento e experiência muito me serviram de inspiração e motivação.

Agradeço aos colegas de curso, em especial ao Darlisson Marinho, que deu grandes contribuições e um forte apoio na realização deste projeto.

Agradeço aos meus pais, Salvador e Suely, que não mediram esforços ao me apoiar e guiar nessa jornada. Todo o carinho e suporte, em todos os sentidos, que me propiciaram, foram de fundamental importância para que eu chegasse onde cheguei.

Por fim, agradeço à Elisângela, minha companheira, que me aturou, apoiou, e permaneceu sempre ao meu lado, seja nas noites mal dormidas, momentos de tristeza e em cada vitória. A sua paciência e perseverança sempre me inspiraram a fazer mais e ser cada vez melhor.

RESUMO

A binarização de imagens digitais é uma técnica amplamente utilizada, uma vez que documentos monocromáticos necessitam de menor espaço de armazenamento e banda de transmissão em redes de computadores. Além disso, a binarização é etapa usual em muitos processos complexos de processamento de imagens, tais como a transcrição automática de documentos. Esta dissertação de mestrado propõe uma metodologia para análise da qualidade das imagens resultantes de algoritmos de binarização baseada em imagens sintéticas. Tais imagens são geradas a partir de um conjunto de imagens binárias de referência com a adição de características extraídas de documentos reais, tais como textura do papel e escrita, interferência frente-verso, etc. As imagens sintéticas são, então, binarizadas e comparadas com as imagens de referência. Quanto mais próximo no número de *pixels* brancos e pretos da imagem de referência, considera-se melhor o desempenho do algoritmo. Os tempos de processamento também são coletados. Um total de $2.083.200$ documentos representativos do universo de documentos textuais foram sintetizados e binarizados. Visando uma ampla divulgação, os resultados obtidos foram disponibilizados numa plataforma web, na qual o usuário escolhe os parâmetros, a plataforma gera o documento sintético e apresenta os resultados para cada algoritmo testado.

Palavras-chave: Binarização. Síntese de textura. Processamento de imagens.

ABSTRACT

Binarization of digital images is a technique widely used, as monochromatic documents require less storage space and transmission bandwidth in computer networks. Besides that, binarization is applied in many complex image processing applications, such as automatic document transcription. This M.Sc. dissertation presents a methodology for assessing the performance of binarization algorithms based on synthetic images. Such images are generated from a set of *ground truth* binary images with the addition of features extracted from real documents, such as paper and writing textures, back-to-front interference, etc. The synthetic images are then binarized using several algorithms and compared with the *ground truth* images. The closer the number of black and white *pixels*, the better is considered the performance of the algorithm. The processing times are also collected. A total of 2,083,200 documents, representative of the universe of textual documents, were synthesized and binarized. Aiming at a wider dissemination, the results obtained were made available on a web platform, in which the user chooses the parameters, the platform generates the synthetic document and then presents the binarization results for each of the tested algorithms.

Keywords: Binarization. Texture synthesis. Image processing.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A comunicação é um fator crucial para que os seres humanos possam conviver em grupo. Conforme evoluiu, o homem tornou sua comunicação cada vez mais complexa e sofisticada, passando da linguagem oral e gestual para registros físicos. Desde a pré-história, período que antecede a escrita, é possível encontrar pinturas feitas em pedras, cavernas e pequenos objetos, que davam lugar à arte e à conservação de informações. Mais tarde, com o aparecimento da escrita, as informações passaram a ser representadas por meio de sinais ou símbolos que permitem registrar, com grande precisão, uma ideia humana, em especial, a linguagem falada.

A primeira forma de escrita desenvolvida, a escrita cuneiforme, originou-se na antiga Mesopotâmia, há cerca de 4.000 a.C. [1], devido à necessidade de guardar registros de contas e transações comerciais. Usava-se uma espécie de caneta de madeira com a ponta em forma de cunha para gravar suas formas abstratas em placas de argila molhada, que, quando aquecidas, endureciam e preservavam as informações. Desde então, os meios utilizados para os registros evoluíram bastante na busca de mais facilidade no registro, transporte e durabilidade. Vários materiais já foram utilizados para a escrita: pedra, cerâmica, madeira, fibras vegetais e, finalmente, o papel como o conhecemos.

Por volta de 2.500 a.C., os egípcios desenvolveram um material para escrita prático, leve e sofisticado: o papiro. Feito à base de uma planta aquática abundante no rio Nilo, o papiro, precursor do papel, facilitou a mobilidade da informação e armazenamento dos registros. Para confeccioná-lo, corta-se o miolo esbranquiçado e poroso do talo da planta em lâminas finas que, através de um processo que dura vários dias, são tratadas e prensadas até formar uma espécie de papel amarelado. Então, era enrolado a uma vareta de madeira ou marfim para criar o rolo que seria usado na escrita.

O papel, tal como conhecemos hoje, surgiu na China por volta do ano 150 D.C. Era fabricado com fibras de árvores e trapos de tecidos cozidos e esmagados, sendo sua massa resultante exposta ao sol para secagem. Sua fórmula era mantida em segredo, até que, cerca de 1.000 anos depois, foi introduzido na Espanha e espalhou-se pelo Ocidente. Devido à sua praticidade, custo de produção, portabilidade e qualidade, este meio de armazenamento de informações se tornou extremamente popular, substituindo todas as outras formas anteriores e sendo usado até os dias atuais [2].

Até a segunda metade do século XX, o papel havia se tornado o principal meio de armazenamento e transmissão do conhecimento da humanidade. Apesar de sua importância, é de difícil preservação e armazenamento. Pode ser facilmente rasgado, amassado, dobrado ou molhado e, com o passar do tempo, torna-se frágil. Caso não seja conservado em um ambiente apropriado, sofrerá com a ação de fungos, insetos e umidade, acarretando em perda da informação

contida nele. Além disso, devido ao seu peso e espaço que ocupa, o armazenamento e transporte de grandes quantidades de papel torna-se uma tarefa difícil e custosa.

Em empresas públicas e privadas, é comum o uso de estantes e salas de arquivos para armazenar e organizar os documentos mais importantes, como por exemplo, contratos, notas fiscais, relatórios, manuais, etc. Empresas de grande porte, que precisam armazenar grandes quantidades de papel, terceirizam o serviço de armazenamento e organização dos documentos. O motivo pelo qual o papel ainda ocupa uma função essencial no processo de comunicação é o fato de ser barato, portátil, fácil de anotar e, principalmente, de ler.

Com o advento e disseminação do computador, surgiram formas mais eficientes de armazenamento, preservação e transmissão da informação. Passou a ser possível criar, alterar e armazenar dados textuais virtualmente. Então, com a criação de dispositivos como *scanners*, câmeras digitais e, mais recentemente, *smartphones*, tornou-se possível transpor documentos manuscritos ou impressos para imagens em formato digital. Esse processo de transposição ficou conhecido como *digitalização*. Uma vez digitalizados, os documentos podem ser transmitidos sem perda alguma de informação posterior à digitalização. Fisicamente, ocupam muito menos espaço e, conforme o tempo passe, não sofrem degradação alguma [2].

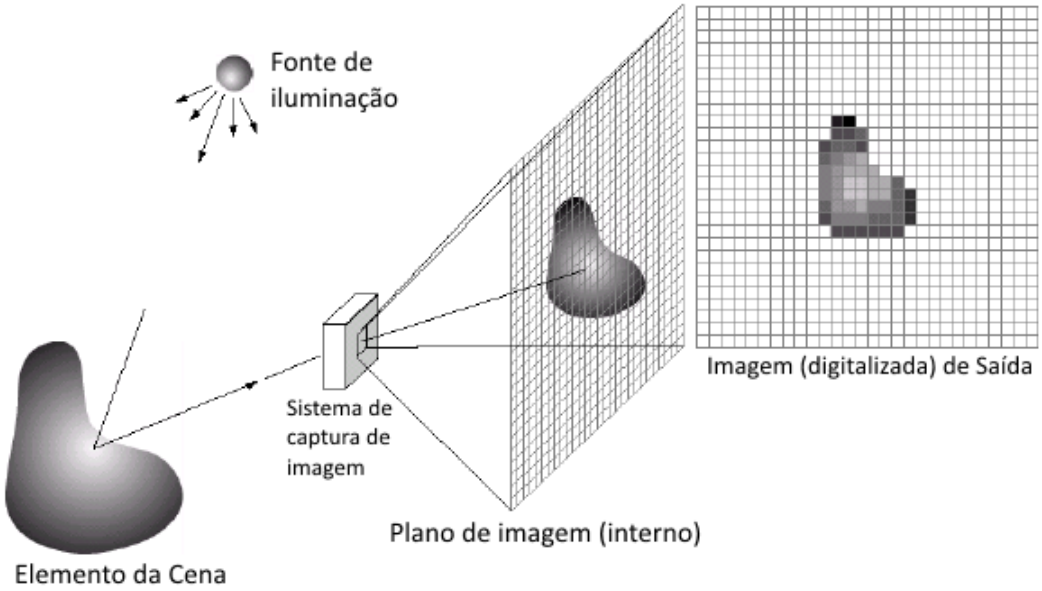
1.1 Digitalização de Documentos

O processo de digitalização de um documento se dá, em essência, como ilustrado Figura 1. Uma fonte de iluminação emite luz sobre o objeto, que é refletida e absorvida por um sistema de captura de imagem. Esse sistema fornece como saída uma imagem digital, composta por elementos gráficos (pontos) chamados *pixels*. O comprimento, em polegadas, que corresponde a 1 *pixel* é determinado pela configuração do dispositivo e é medido em *dpi – dots per inch* ou pontos por polegadas. Esse sistema pode ser tanto um *scanner* de mesa, uma máquina fotográfica digital ou, ainda, um *smartphone*.

Os *pixels* que compõem uma imagem digital geralmente têm valores: *OFF* (0) ou *ON* (1) para imagens binárias, de 0 a 255 para imagens em escala de cinza e 3 canais de 0-255 para imagens coloridas. Uma representação para imagens coloridas muito utilizada em aplicações de processamento de imagens é a RGB – *Red, Green, Blue*, no qual a imagem é dividida em três canais contendo os níveis de tons de vermelho, verde e azul, respectivamente. Cada *pixel* requer 8 bits por canal, totalizando 24 Bits por *pixel* e 16,78 milhões de cores possíveis.

No processamento de imagem com a finalidade de extrair informações ou identificar padrões, é comum converter os três canais em apenas um, transformando de RGB para a representação em 8 bits, com 256 níveis de tons de cinza (em inglês, *grayscale*) [3]. O principal método utilizado para realizar a conversão, também usado nesta dissertação, é o cálculo da

Figura 1 – Processo de Aquisição de Imagem Digital



Fonte – Adaptado de Gonzalez, Woods e Masters [3]

luminância, dado pela equação (1.1):

$$Y(x,y) = 0,2126R(x,y) + 0,7152G(x,y) + 0,0722B(x,y), \tag{1.1}$$

sendo R, G e B as matrizes de cada canal, Y é a matriz em escala de cinza (luminância) resultante e (x,y) indica as coordenadas de cada ponto na matriz.

Apesar das vantagens de se digitalizar documentos, o espaço necessário para armazenar as imagens é um problema crítico. Por exemplo, hoje, para uma configuração de digitalização típica, de 300 dpi, imagem colorida de 24 Bits e uma folha de tamanho A4 (210×297 mm), será gerada uma imagem contendo 2.480×3.508 *pixels*, totalizando, aproximadamente, 24,89 MBytes, se armazenada sem compressão. Se a imagem for composta apenas de texto, a mesma informação pode ser armazenada com menos de 100 Kbytes no formato de texto digital. Sendo assim, a conversão de imagens de documentos impressos e tipografados para formato de texto provê uma diminuição no espaço de armazenamento, além de facilitar a implementação de outras operações comuns, como busca por palavras-chaves.

A técnica para transcrever imagens de documentos textuais em texto digital editável chama-se OCR (*Optical Character Recognition*) [4], que engloba as etapas de análise de características e análise textual. O termo OCR é muito mais comumente usado para designar o processo de reconhecimento de caracteres impressos, tipografados ou datilografados, porém, como descrito por Reshma [5] e Kasturi [6], também designa o reconhecimento de caracteres *manuscritos*. Isso porque, independente do tipo de escrita (impresso ou manuscrito), as principais etapas do OCR são, em geral, as mesmas: pré-processamento, segmentação, representação, treinamento e

reconhecimento e pós-processamento, nessa ordem [5]. Porém, as técnicas utilizadas em cada etapa variam de acordo com o tipo de documento.

Para documentos impressos e datilografados recentes, de boa qualidade e sem degradação, os principais problemas de OCR já possuem soluções quase ótimas [7]. No entanto, caso a imagem apresente qualquer tipo de ruído, o OCR se torna um desafio. Especialmente no caso de documentos manuscritos, o processo de transcrição em documento textual é ainda mais problemático, pois não há uma padronização na grafia. Sendo assim, sistemas de reconhecimento de caracteres manuscritos requerem um processo de aprendizagem e processamento diferentes e mais complexos.

Em se tratando de imagens de documentos históricos, o problema é ainda mais complexo, pois além da falta de padronização, documentos históricos são caracterizados pela sua degradação. Devido à ação natural de agentes externos, ao longo do tempo, variados tipos de ruídos surgem nos documentos. Esse fator acarreta em uma alteração na estrutura original do documento, gerando problemas como a aparição de manchas, desgaste do papel, perda de informações de texto, realce de informações contidas no verso da folha, regiões amassadas, etc. A Figura 2 exibe uma imagem de documento histórico com alguns desses problemas.

1.2 Binarização de Documentos

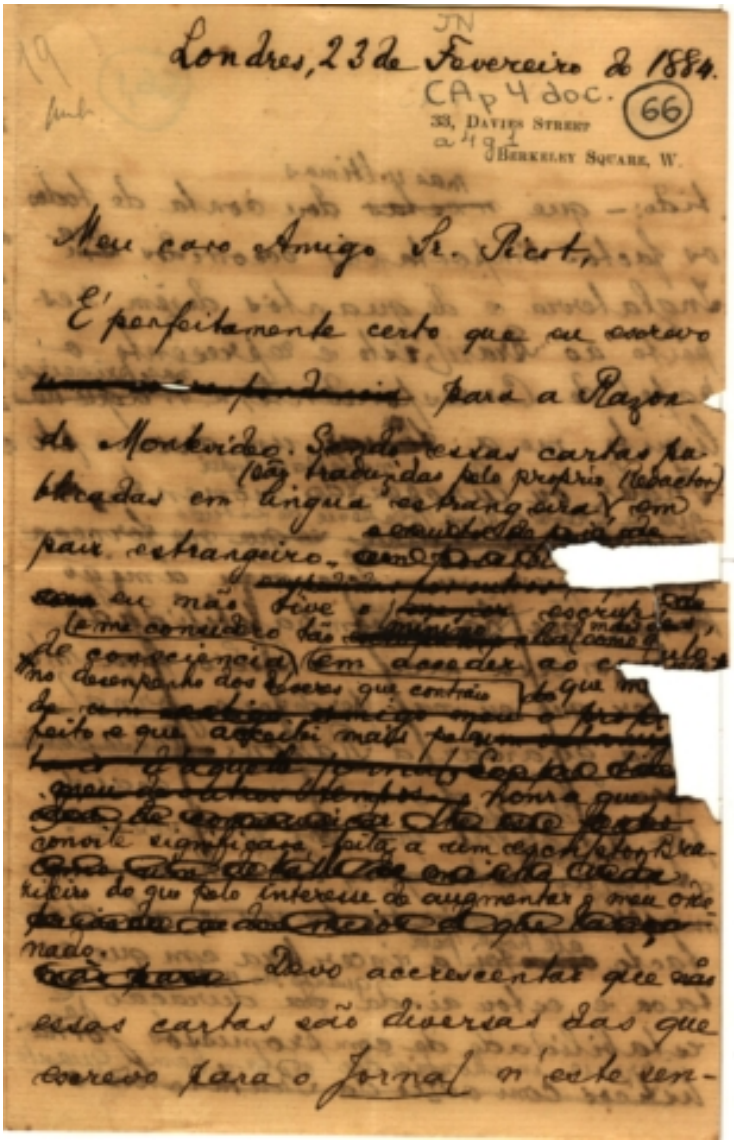
A Binarização é o processo pelo qual uma imagem digital é convertida da representação colorida ou monocromática para binária. Para tal, os *pixels* são divididos em duas categorias: texto (frente ou *foreground*) e fundo, ou tudo que não for texto (*background*). A binarização também é chamada de limiarização (*thresholding*), pois o processo mais comum é selecionar um limiar para dividir os valores da representação em escala de cinza nos dois grupos de *pixels* mencionados. Em geral, caso a imagem seja colorida, ela é primeiro convertida para a representação em escala de cinza. Para exemplificar, na Figura 3 é apresentada uma imagem de documento histórico, manuscrito, juntamente com a versão binária gerada utilizando-se o clássico algoritmo de Otsu [9]

Diversos fatores influenciam a binarização e cada método de binarização se comporta diferente de acordo com as características do documento. Sendo assim, é importante realizar estudos centrados nas imagens, que buscam encontrar qual o melhor algoritmo no contexto de cada imagem. Nesta dissertação, estudou-se o desempenho de algoritmos de binarização quando aplicados a imagens sintéticas. Um processo automatizado foi desenvolvido para gerar e processar milhões de imagens sintéticas e os resultados em termos de qualidade e tempo de processamento foram registrados para cada imagem.

1.3 Estrutura desta Dissertação

O conteúdo desta dissertação está organizado da seguinte forma:

Figura 2 – Exemplo de Documento Histórico



Fonte – Projeto Nabuco [8]

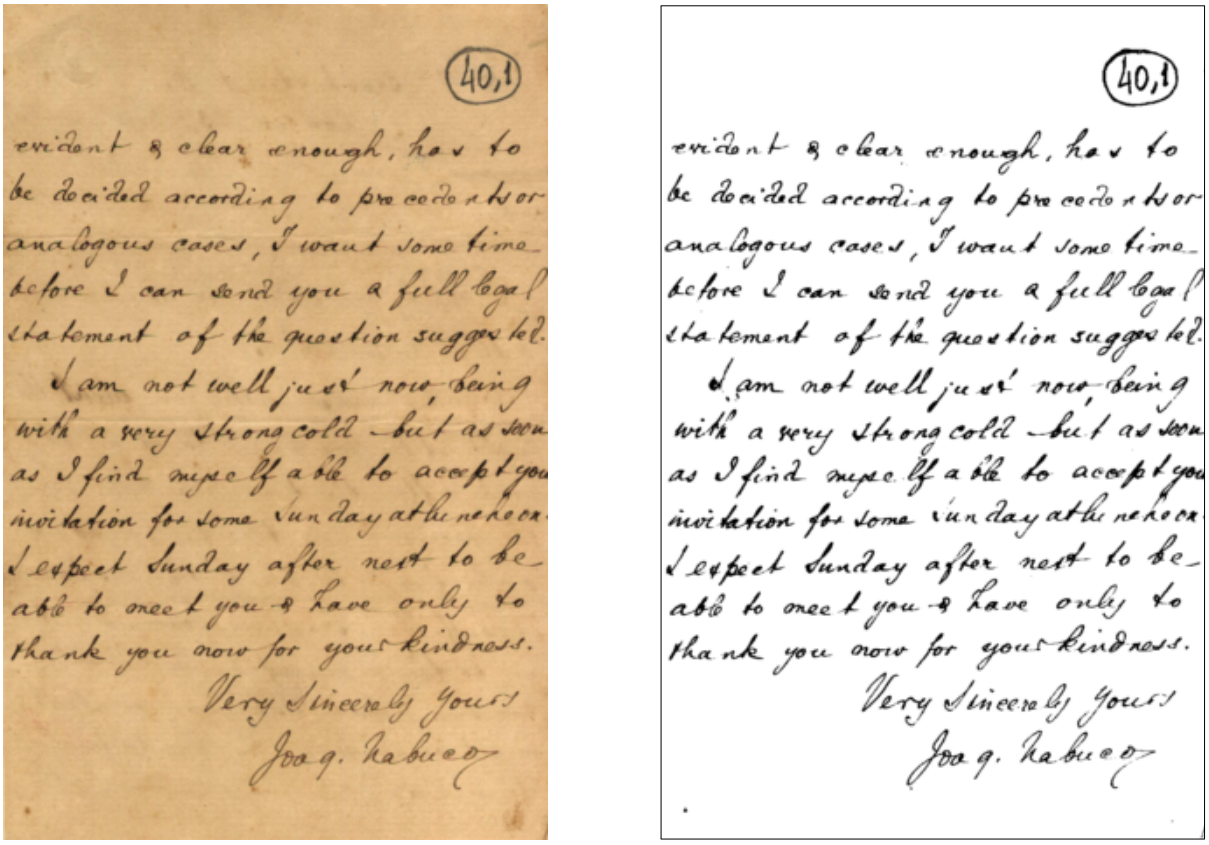
O Capítulo 1 é a presente introdução.

No Capítulo 2, um novo método de síntese de imagens de documentos é apresentado. A síntese se divide em geração dos *pixels* do texto, interferência frente-verso, adição de textura do papel sintética e, por fim, mesclagem das três imagens.

No Capítulo 3, um novo método de avaliação de algoritmos de binarização é apresentado e é feita breve discussão sobre os algoritmos considerados.

No Capítulo 4, são apresentadas as considerações finais e apresentadas sugestões de trabalhos futuros centrados na plataforma desenvolvida.

Figura 3 – Exemplo de Imagem Binária



(a) Imagem Original **(b)** Imagem Binarizada com Otsu

Fonte – Projeto Nabuco [8]

2 SÍNTESE DE DOCUMENTOS PARA BINARIZAÇÃO

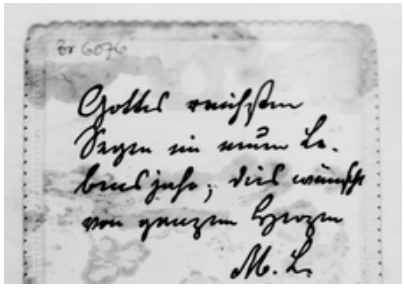
A binarização de documentos é um passo importante no processo de análise e reconhecimento de imagens de documentos. Os concursos internacionais de algoritmos de binarização são uma evidência da importância dessa área. O concurso mais tradicional é, possivelmente, DIBCO – *Document Image Binarization Competition*, que foi organizado pela primeira vez na ICDAR – *International Conference on Document Analysis and Recognition* em 2009 e, desde então, tem ocorrido todos os anos. A metodologia utilizada pelo DIBCO é avaliar algoritmos com medidas de desempenho de avaliação estabelecidas na literatura. Os algoritmos são aplicados a um conjunto de imagens de teste, composto por imagens coloridas e monocromáticas, e as imagens binárias produzidas são comparadas com imagens binárias de referências, chamadas *ground truth* (GT).

Na Figura 4, é apresentado o conjunto de imagens de teste usado no DIBCO 2016 [11]. Como se pode observar, o conjunto de dados do concurso de 2016 é formado apenas por documentos manuscritos, tanto representação em escala de cinza quanto colorido. Alguns desses documentos apresentam manchas (1, 4, 8, 9 e 10) e marcas de envelhecimento (4, 9 e 10). A ferramenta de avaliação utilizada no DIBCO fornece como saída as medidas *F-Measure*, *pseudo F-Measure*, PSNR, DRD, *Recall*, *Precision*, *pseudo-Recall* e *pseudo-Precision*. Detalhes sobre estas medidas fogem ao escopo desta dissertação, mais informações podem ser encontradas nas referências [12, 13].

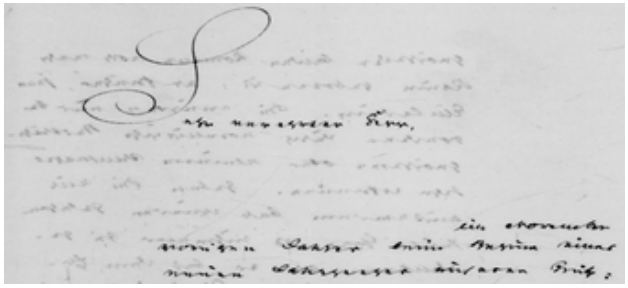
Um detalhe importante a se notar nas imagens do DIBCO 2016 é a presença do ruído de interferência frente-verso em todas as imagens, exceto na primeira (Figura 4a, Imagem 1). Ou seja, no caso do documento ter sido escrito dos dois lados do papel, a opacidade do papel é tal que permite que o conteúdo escrito no verso do papel possa ser visto na frente (em detalhe na Figura 5 para duas outras imagens). Em imagens coloridas e até mesmo em tons de cinza, o cérebro humano é capaz de, facilmente, filtrar esse tipo de ruído e manter a legibilidade do documento, mas esse não é o caso de ferramentas automatizadas como OCRs. A aplicação direta de certos algoritmos de binarização pode gerar um documento completamente ilegível, uma vez que a tinta proveniente do verso pode ser sobreposta como parte do texto da frente, gerando imagens binárias completamente ilegíveis para humanos e máquinas.

Diversos algoritmos foram desenvolvidos especialmente para tratar a interferência frente-verso [15, 16, 17, 18, 19, 20]. Dependendo da opacidade do papel, permeabilidade e o tipo e grau de fluidez da tinta usada na escrita, a dificuldade em obter uma segmentação capaz de filtrar esse tipo de ruído aumenta drasticamente. Uma vez que essas e outras características variam muito entre os documentos, a binarização se torna uma tarefa extremamente desafiadora. Não há como um algoritmo específico fornecer o melhor resultado possível para todo tipo de documento [21]. Além disso, é improvável que um pequeno conjunto de teste possa fornecer

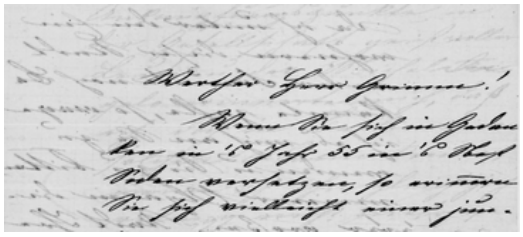
Figura 4 – Imagens de Teste do DIBCO 2016



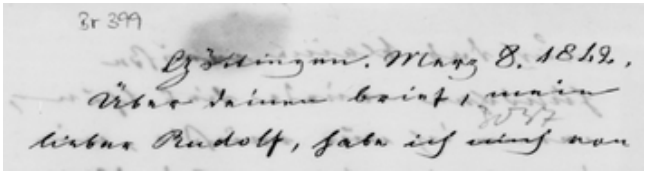
(a) Imagem 1



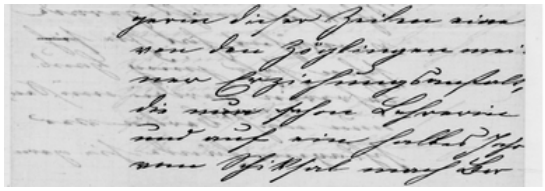
(b) Imagem 2



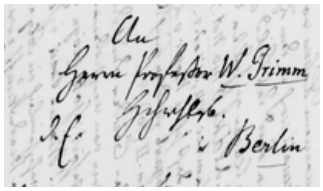
(c) Imagem 3



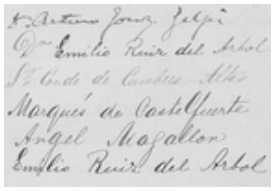
(d) Imagem 4



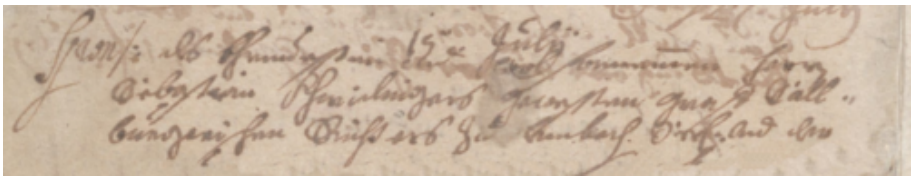
(e) Imagem 5



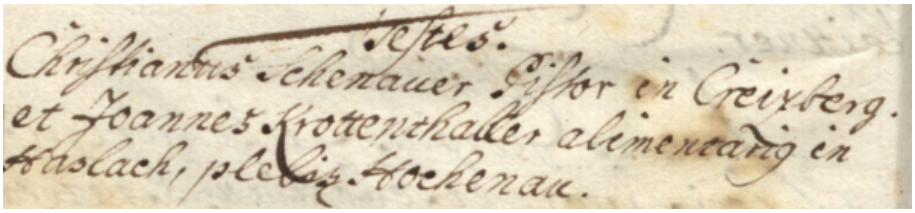
(f) Imagem 6



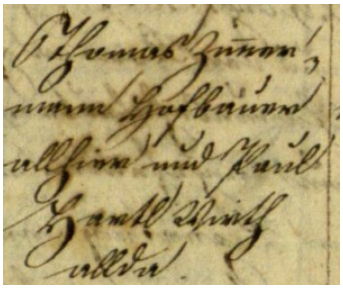
(g) Imagem 7



(h) Imagem 8



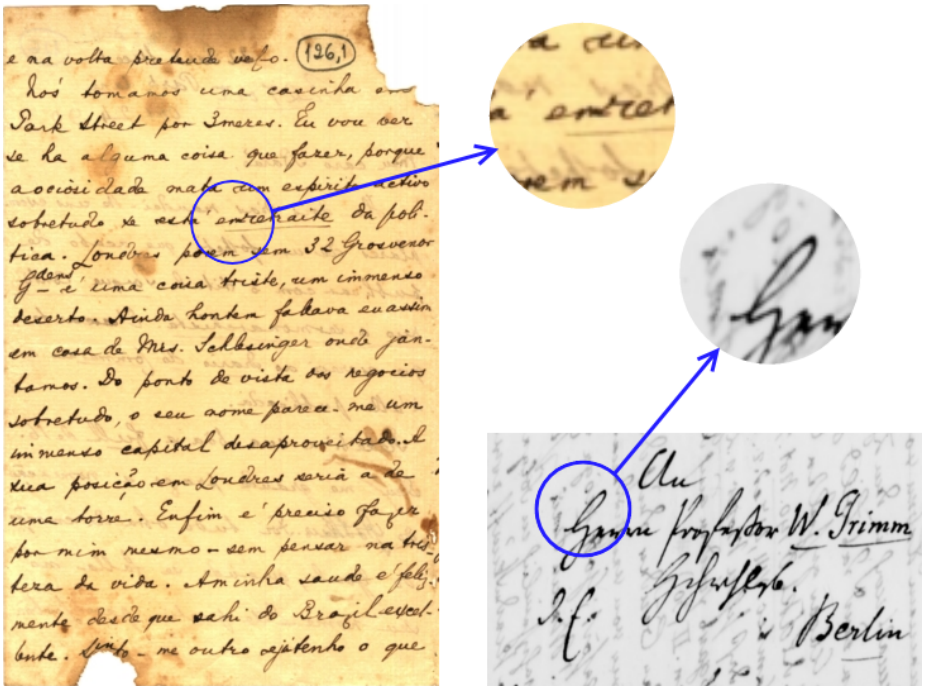
(i) Imagem 9



(j) Imagem 10

Fonte – Website do Concurso DIBCO 2016 [14]

Figura 5 – Documentos com Interferência Frente-Verso



Fonte – Esquerda: Projeto Nabuco [8]; direita: DIBCO 2016 [14]

uma análise representativa dos algoritmos de binarização. É importante ter a possibilidade de verificar a eficácia de cada método em um vasto conjunto de imagens sintéticas representativas do universo de documentos textuais.

Outro detalhe pouco investigado nos estudos da literatura [22, 23, 11], é o tempo de processamento requerido para binarizar as imagens. Os ambientes de execução em que algoritmos de binarização são utilizados variam de sistemas embarcados, com severas limitações de hardware, computadores pessoais com capacidade mediana a até servidores para computação de alto desempenho. Especialmente nas situações em que haja limites na capacidade de processamento e/ou memória, a complexidade computacional se torna um fator decisivo na escolha do algoritmo a se usar. Por exemplo, no caso em que dois algoritmos forneçam resultados com qualidade semelhante, mas um tenha um custo computacional muito mais elevado que o outro, caso haja limitações de *hardware* na aplicação, o de custo menor e com resultados aceitáveis será a melhor escolha.

O novo método de síntese de imagens de documentos proposto inicia-se por extrair os *pixels* correspondentes ao texto de uma imagem de documento e gerar duas imagens: texto da frente e texto do verso. Então, uma imagem de folha de papel é sintetizada a partir de uma amostra de textura. O texto do verso é deslocado, verticalmente, por 10, 20 e 30 *pixels* para que a imagem do verso não coincida com a da frente. A seguir, um borramento é aplicado à imagem do verso, simulando o efeito de filtro passa-baixa da translucidez do papel. Finalmente, as imagens da frente, verso e textura de papel são combinadas, sendo que a imagem do verso é esmaecida

por um coeficiente α (alfa), que é variado entre 0 e 1 a passos de $0{,}1$.

Na Figura 6, página 20, o processo completo de síntese é ilustrado resumidamente. Um recorte de uma imagem de exemplo é apresentado em cada passo. Primeiramente, são necessários uma imagem de documento real, o seu *ground truth* e uma textura de papel. Então, gera-se a imagem da frente e da interferência frente-verso. Em seguida, a imagem do verso é colada sobre a textura e, então, aplica-se a transparência aos *pixels* do texto da interferência frente-verso, bem como o *blur*. Por fim, uma operação especial, chamada "operação *darker*" (detalhada na seção 2.4), é aplicada para acrescentar os *pixels* da frente à imagem de fundo ruidoso. Neste capítulo, todos esses passos do processo de síntese são descritos em detalhes.

2.1 Geração dos *Pixels* do Texto

O primeiro passo na geração de imagens sintéticas foi a geração de um banco de imagens que fornecessem uma boa representação do universo de documentos textuais:

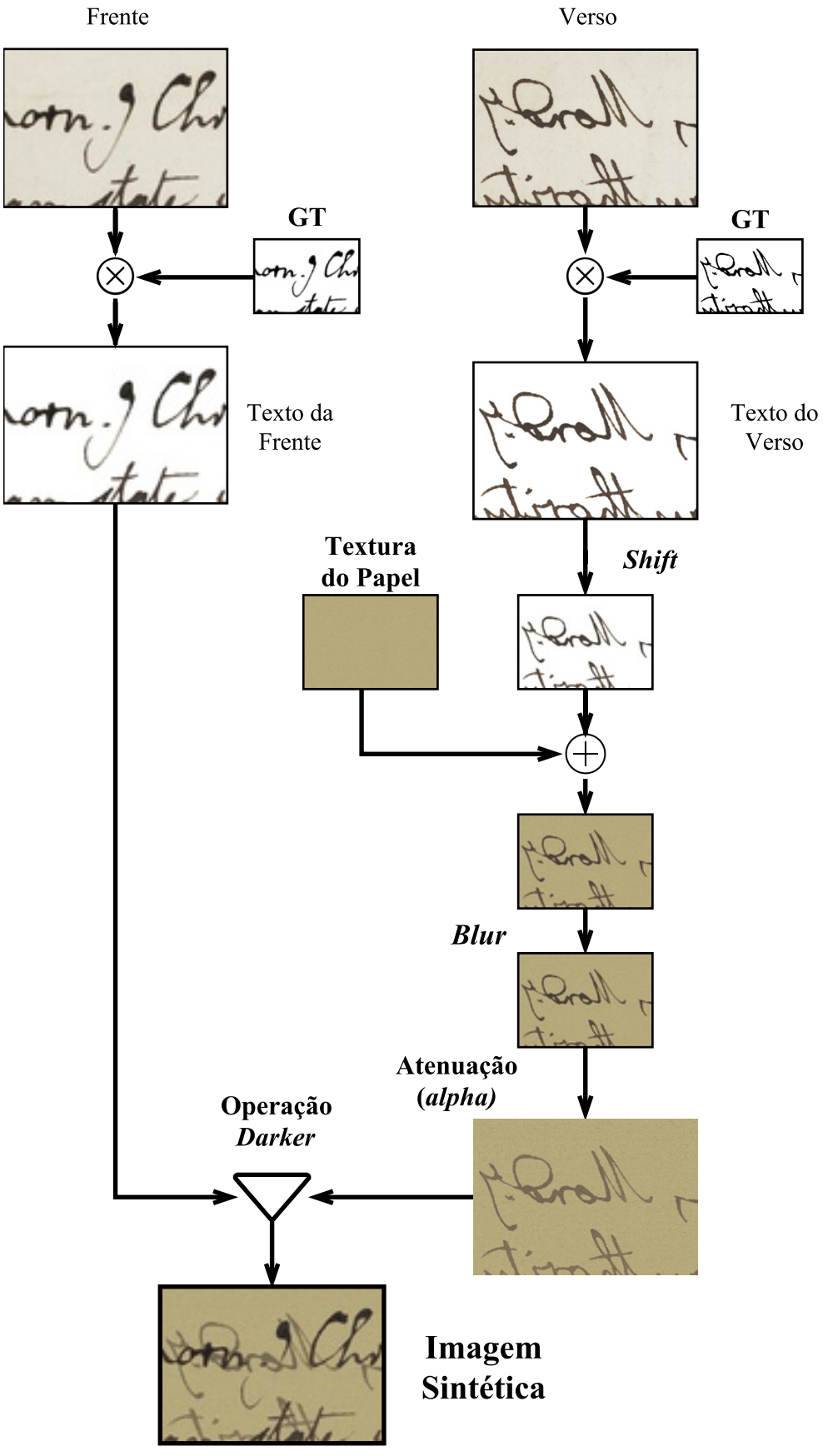
- Documentos datilografados;
- Impressos com impressoras de jato de tinta, laser e *offset* nas cores mais comuns (preto, azul e vermelho);
- Manuscritos com diferentes tipos de caneta (*fountain*, esferográfica e pena), de variados fabricantes, usando tinta *preta* e azul, além de três caligrafias diferentes.

Os documentos impressos são provenientes de anais de congressos do SBrT, digitalizadas no projeto *LiveMemory* [24], coordenado pelo orientador desta dissertação. Para compor as imagens geradas neste trabalho, documentos com as características citadas foram impressos/digitados/escritos em folhas de papel brancas, tamanho A4, de boa qualidade. As imagens foram, então, digitalizadas utilizando um *scanner* de mesa configurado para uma resolução de 300 dpi, *True Color* (imagem RGB de 24 bits), gerando imagens de 2.480×3.508 *pixels*. Ao final do processo, foram geradas imagens de 43 documentos manuscritos e 88 impressos.

Para que possam ser usadas no processo de síntese, foi preciso gerar as imagens de *ground truth* dessas imagens. Para tanto, elas foram binarizadas com o algoritmo de Otsu [9], que, para documentos de boa qualidade (sem ruídos), apresenta excelentes resultados. Então, passaram por uma filtragem de ruído de sal e pimenta. Esse ruído remove pontos pretos isolados, ou seja, rodeados por *pixels* brancos, invertendo a sua cor. *Pixels* brancos isolados também são invertidos. Por fim, os *pixels* invertidos na imagem de *ground truth* foram também trocados de "*pixels* do texto" para fundo e "*pixels* do fundo" para texto. Todo este processo é ilustrado na Figura 7.

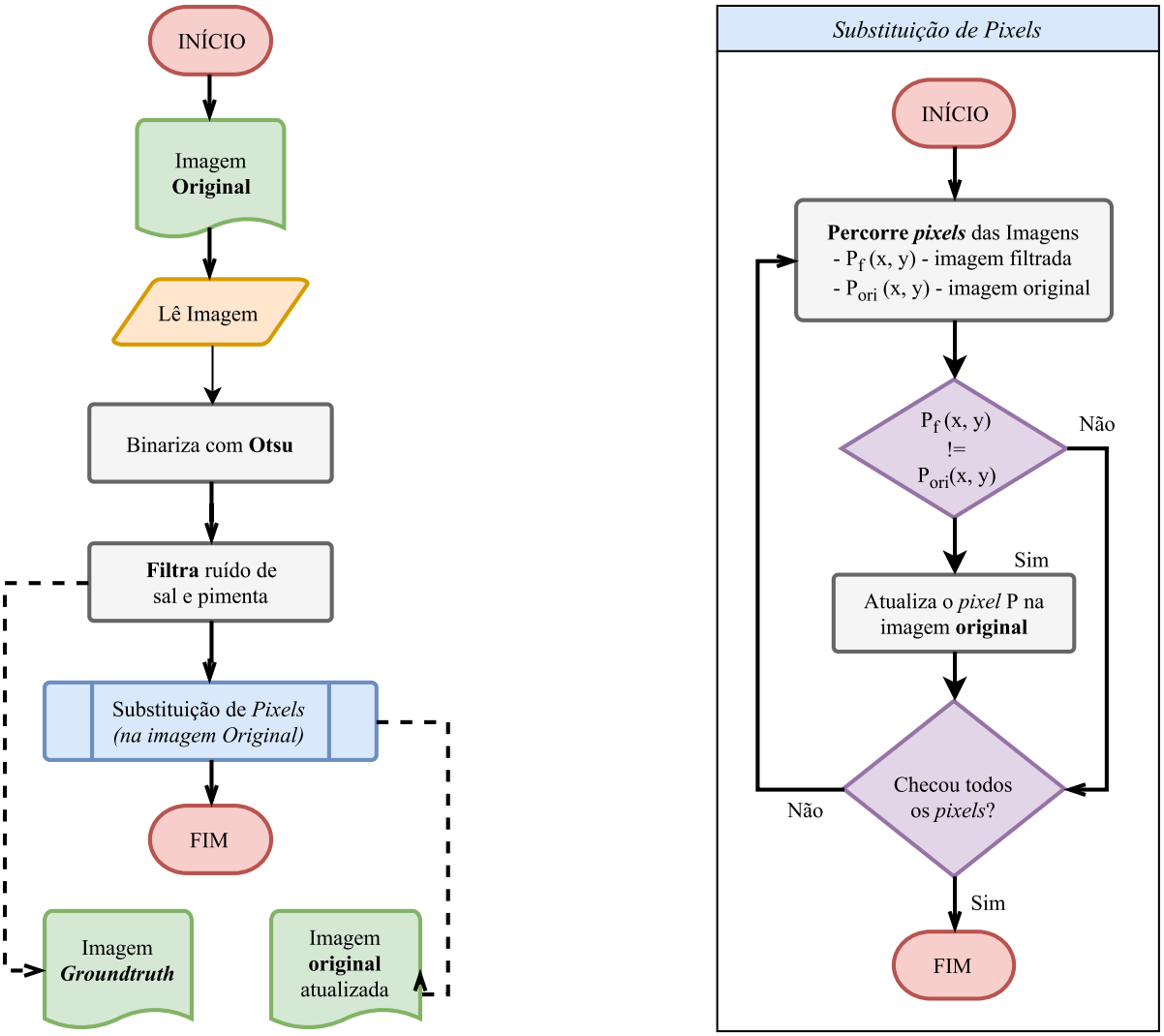
Além dos documentos gerados no contexto deste trabalho, as imagens do concurso DIBCO, juntamente com suas respectivas imagens de referência (*ground truth*), também foram

Figura 6 – Processo de Síntese



Fonte – Próprio Autor

Figura 7 – Geração de Imagens de Referência (*Ground Truth*)



(a) Processo Completo

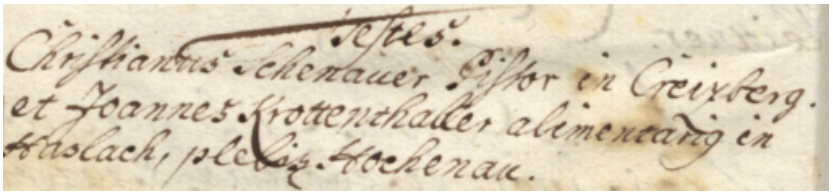
(b) Substituição de *Pixels*

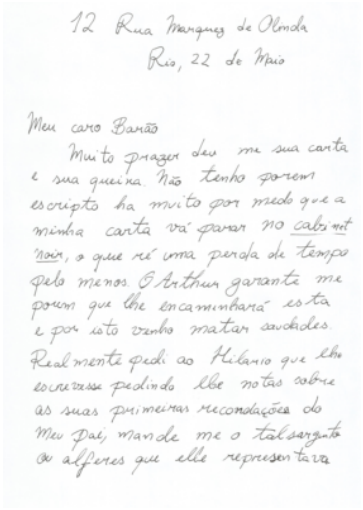
Fonte – Próprio Autor

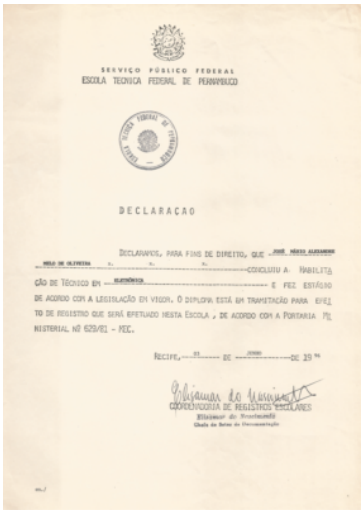
utilizadas [11, 25, 26, 27, 10]. As imagens de toda a série de eventos foram consideradas, totalizando 61 imagens de documentos manuscritos e 25 tipografados. Não há uma padronização quanto à qualidade, resolução ou cor da imagem e as imagens são recortes de alta resolução de documentos reais. Sendo assim, ao final, 104 documentos manuscritos e 113 impressos, ou seja, 217 documentos compõem o banco de imagens utilizado na síntese de imagens da frente (*pixels* do texto). Na Figura 8 algumas dessas imagens são apresentadas.

No momento da síntese de documentos, uma dessas imagens e seu GT são fornecidos como entrada para o processo. Inicialmente, três abordagens para geração dos *pixels* do texto foram consideradas: cópia do GT, cópia da imagem original e subtração adaptativa. Na cópia do GT, os *pixels* pretos da imagem GT foram utilizadas diretamente, porém os resultados mostraram que qualquer algoritmo era capaz de facilmente preservar o texto, uma vez que tinha valor

Figura 8 – Exemplos de Imagens Usadas na Síntese da Imagem do Texto







Fonte – DIBCO [14] e Próprio Autor

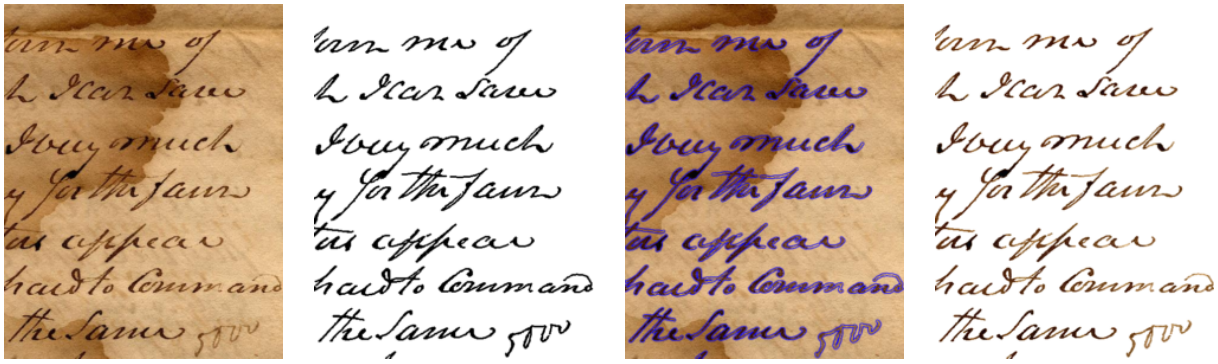
mínimo (0, preto), totalmente diferente do restante da imagem. Além disso, não há como uma imagem real ter os *pixels* do texto 100% pretos.

A segunda abordagem foi a cópia dos *pixels* da imagem original. Nesta estratégia, a distribuição de cores do documento real é preservada, o que leva à geração de imagens mais realistas. A abordagem de cópia dos *pixels* consiste em usar o GT como máscara de recorte sobre a imagem original para gerar uma nova imagem contendo apenas os *pixels* do texto, como ilustrado na Figura 9. Os resultados com essa abordagem se mostraram satisfatórios, uma vez que as imagens geradas eram visualmente semelhantes a documentos reais e os algoritmos se comportam de forma variada, como se espera de documentos reais.

Porém, sabe-se que a cor do papel influencia na cor final do texto, uma vez que a tinta é absorvida pelo papel. Sendo assim, caso a textura do papel, de onde os *pixels* do texto foram extraídos, seja muito diferente da textura sintética onde serão inseridos, o resultado pode não ser tão natural quanto se espera. Portanto, foi desenvolvido um outro método de geração de *pixels* do texto que adapta os *pixels* do texto à textura onde serão aplicados. Neste método, os histogramas da representação em escala de cinza da textura da imagem da frente e dos *pixels* do texto são alinhados. Então, as cores da imagem da frente são invertidas e as duas imagens são subtraídas.

Para algumas imagens, o método adaptativo apresentou bons resultados, no entanto, na maioria dos casos, ocorreu a saturação de alguns ou muitos *pixels*. A saturação se caracteriza

Figura 9 – Exemplo de Cópia dos *Pixels* do Texto



(a) Imagem Original **(b)** *Ground Truth* **(c)** Texto Selecionado **(d)** Texto Extraído

Fonte – Próprio Autor

pela conversão (após a subtração) de um *pixel* colorido para 100% preto ou branco. Sendo assim, o método que fornece melhores resultados, dentre os considerados, é a cópia direta da imagem original. Na Figura 10, um exemplo de resultado para cada método é exemplificado.

2.2 Geração da Interferência Frente-Verso

Como já foi mencionado, documentos com interferência frente-verso são muito mais complexos para se binarizar. Diversos fatores influenciam para que a tinta do verso do papel seja vista mais ou menos embaçada e esmaecida na frente do papel: espessura; textura; permeabilidade; idade; condições de armazenamento (temperatura, umidade, exposição à luz do sol); tipo de tinta usada no processo de impressão ou caneta, no caso de manuscritos.

Esse efeito foi modelado com a aplicação de um filtro gaussiano, com núcleos de 3×3 e 5×5 *pixels*. Para simular o esmaecimento (transparência), a imagem do verso é aplicada à imagem da textura tendo cada um dos seus *pixels* ponderados por um coeficiente α, utilizando a equação:

$$I_m(x, y) = \alpha * I_f(x, y) + (1 - \alpha) * I_v(x, y), \tag{2.1}$$

sendo I_m a imagem mesclada, I_f a imagem da frente e I_v a imagem do verso. O coeficiente *alpha* varia por 10 níveis, logo $0 < \alpha < 1$, a passos de 0,1. Ao realizar uma inspeção visual em algumas imagens com diferentes valores para o coeficiente α e núcleo do filtro, os ruídos de interferência frente-verso nas imagens sintéticas apresentaram efeito similar aos encontrados em documentos reais, como ilustrado na Figura 11.

Figura 10 – Exemplo de Geração de *Pixels* do Texto

(a) Imagem Original **(b)** *Ground Truth* **(c)** Textura Sintética

(d) Cópia do *Ground Truth* **(e)** Cópia da Imagem Original **(f)** Subtração Adaptativa

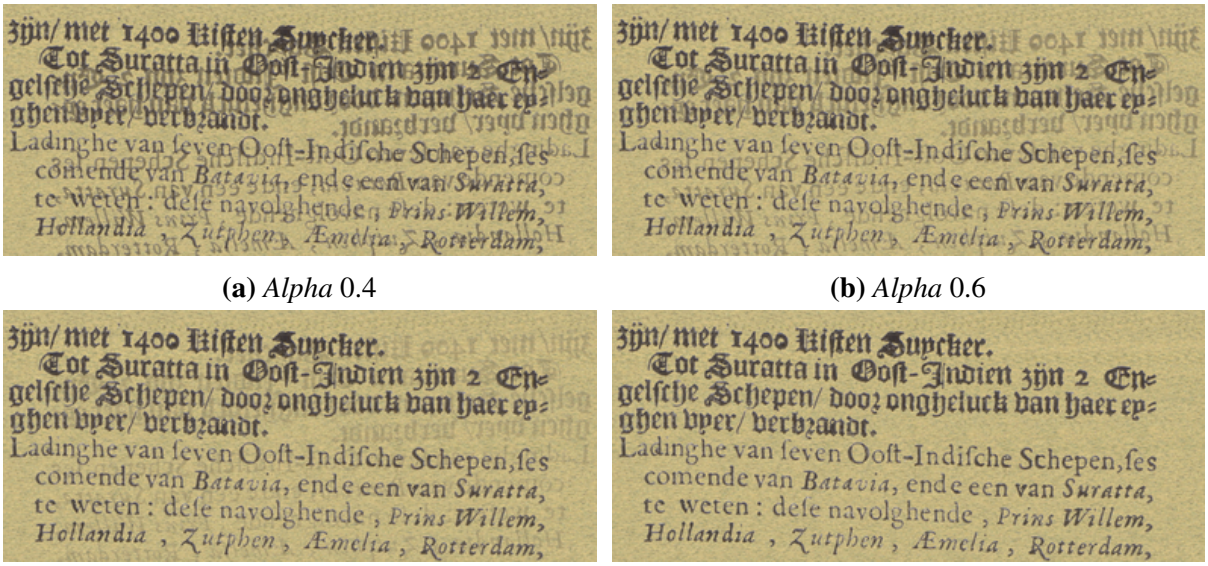
Fonte – Resultados da Pesquisa

2.3 Amostras de Textura de Papel

A síntese de textura do papel consiste em transformar um recorte de textura, extraída de um documento real, em uma nova imagem com dimensões maiores, preservando a aparência da textura original. Na Figura 12, página 25, é apresentado um exemplo de uma textura de 100×100 *pixels*, que foi extraída e processada para gerar uma nova textura, com o mesmo padrão visual, mas com dimensões de 500×700 *pixels*. Logo, para gerar os documentos sintéticos, foi necessário obter um conjunto de amostras de texturas de papel.

A textura do papel tem forte influência no desempenho dos algoritmos de binarização. Sendo assim, é fundamental ter amostras de texturas de papel que sejam representativas do universo de documentos que se pretende modelar, que datam desde o século 19 até os dias atuais. Para tal, 3.351 imagens de documentos reais foram utilizadas, das quais 1.048 são cartas escritas

Figura 11 – Interferência Frente-Verso Sintética

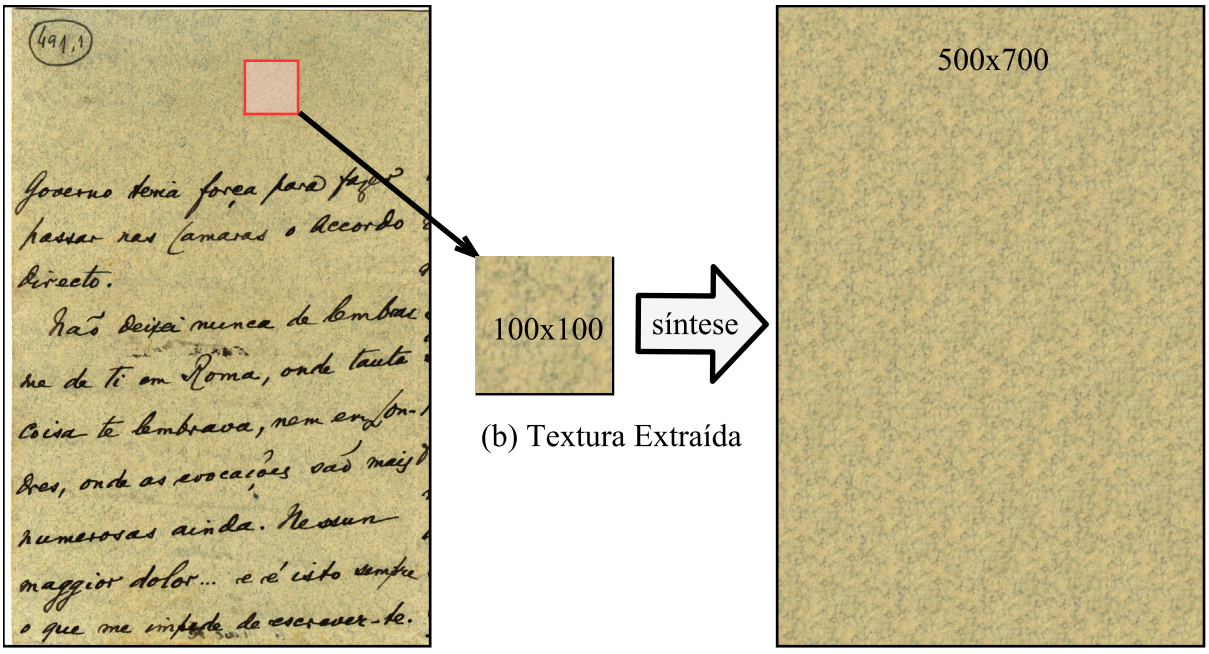


(a) *Alpha* 0.4 **(b)** *Alpha* 0.6

(c) *Alpha* 0.8 **(d)** *Alpha* 1.0

Fonte – Resultados da Pesquisa

Figura 12 – Exemplo de Síntese de Textura



(a) Documento Original (b) Textura Extraída (c) Textura Sintética

Fonte – Próprio Autor

por Joaquim Nabuco, importante intelectual, político e diplomata brasileiro do século XIX, obtidas como parte do Projeto Nabuco [8], e 2.303 obtidas do projeto *LiveMemory* [24], no qual foi gerada uma biblioteca digital com todos os anais das conferências da Sociedade Brasileira de Telecomunicações.

As imagens do Projeto Nabuco foram geradas com um *scanner* de mesa, resolução de 200 dpi, e salvas no formato JPEG com taxa de compressão de 1%. As cartas, que datam dos anos de 1860 a 1910, foram escritas em diferentes tamanhos de papel e várias sofreram degradações. As imagens possuem diferentes resoluções: 900×1.400, 1.500×1.800, 1.600×2.000 e 1.100×1.800 *pixels*. Já os anais do SBrT, foram digitalizados com um *scanner* de mesa a 300 dpi, sendo salvos no formato JPEG com taxa de compressão de 2%. Compõem um *dataset* de 2.303 imagens dos anais de 5 anos distintos: 355 imagens de 1988; 17 imagens de 1991; 542 imagens de 1994; 660 imagens de 1998; 729 imagens de 1999. As dimensões das imagens são padronizadas: 2.480×3.507 (anos 88, 91 e 94) e 1.653×2.238 *pixels* (98 e 99).

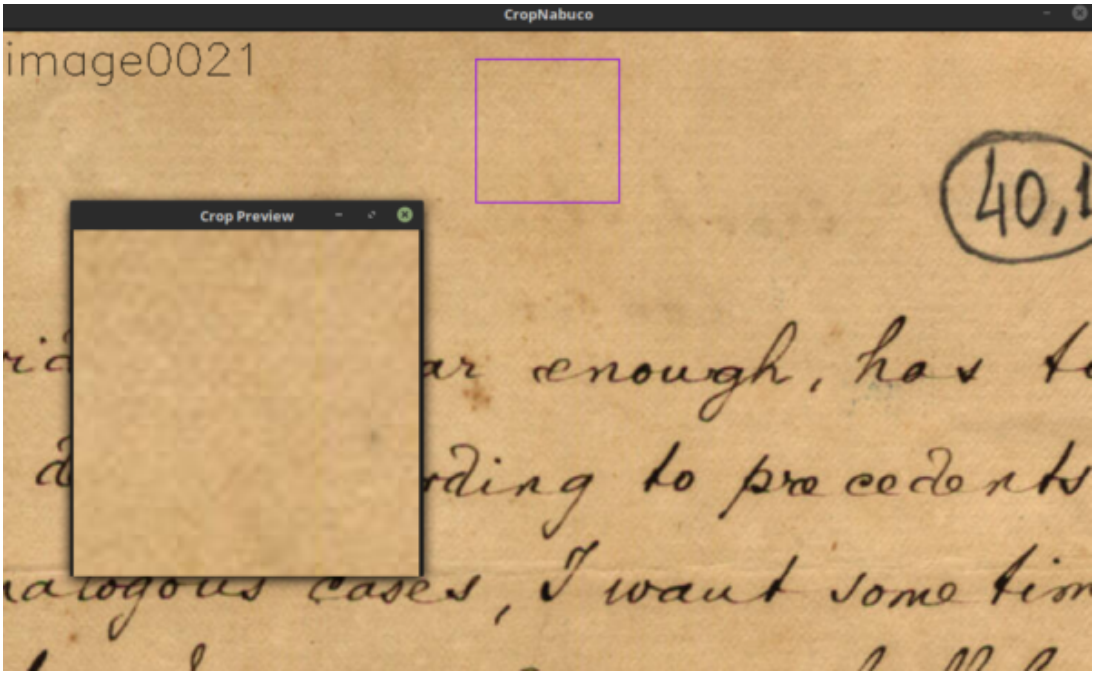
Essas imagens foram recortadas de forma a obterem-se amostras de textura do papel com dimensões de 50×20 *pixels*. É importante que a região capturada não contenha nenhum traço de tinta ou qualquer outro tipo de ruído. Para isso, a região superior dos documentos foi escolhida para o recorte, exceto em alguns documentos do Projeto Nabuco, que continham pouco espaço livre de ruído. Um *software* foi desenvolvido para facilitar o recorte das mais de 3 mil imagens. Este, recebe como entrada um conjunto de imagens e as exibe, uma a uma, juntamente com um retângulo indicando a área a ser recortada. O retângulo é inicialmente posicionado na região superior e, então, manualmente ajustado, com diversas configurações de visualização possíveis. Na Figura 13, uma captura de tela do *software* em execução é apresentada.

Várias dessas amostras de texturas possuem características semelhantes entre si. A binarização de texturas semelhantes irá prover resultados semelhantes, portanto elas foram agrupadas de forma que as texturas dentro de um mesmo grupo fossem semelhantes dentro do grupo, mas diferentes das texturas dos outros grupos. Ao final, uma imagem representativa de cada grupo foi escolhida para formar o conjunto final de texturas de papel. Esse tipo de processo é conhecido na literatura como *clustering*, sendo amplamente estudado com aplicações em várias áreas do conhecimento [28]. Na próxima seção, esse tema é abordado em mais detalhes.

2.3.1 *Clustering*

Clustering é o processo de classificação não supervisionado onde um conjunto de objetos de dados (observações, ponto de dados ou vetor de características) são agrupados em subconjuntos baseados nas informações que os descrevem. Os grupos devem ser compostos de tal modo que os objetos dentro de um mesmo grupo sejam similares entre si e diferentes dos objetos de outros grupos. Grupos construídos dessa maneira são chamados de *clusters*. Quanto maior for a similaridade dentro dos *clusters* e maior for a diferença entre os *clusters*, melhor terá sido o *clustering* [29, 30].

Figura 13 – *NabucoCrop*: Ferramenta de Recorte de Textura



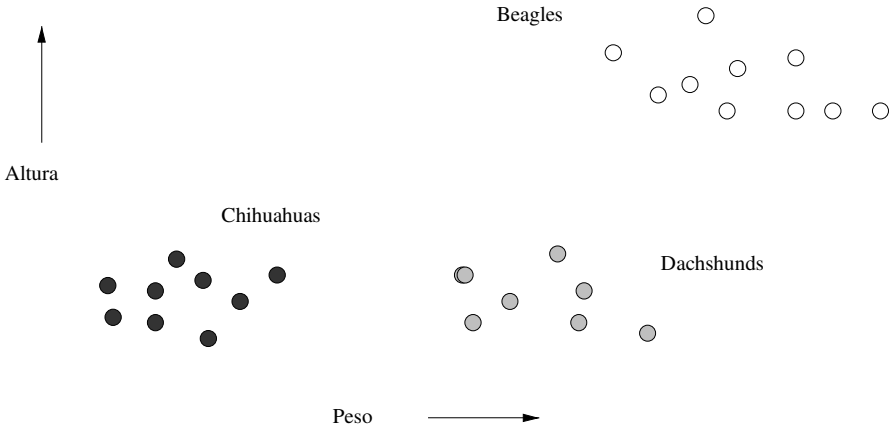
Fonte – Próprio Autor

Os objetos a serem agrupados são frequentemente representados como pontos em um espaço euclideano. Cada ponto é um vetor de números reais, onde cada número indica uma característica ou medida do objeto correspondente. Para que seja possível realizar o *clustering*, é necessário definir uma medida de distância entre os pontos, que varia de acordo com a técnica utilizada. Aplicações clássicas de *clustering* envolvem uma representação em espaços de 2 ou 3 dimensões e a distância Euclideana entre os pontos é usada como medida.

Na Figura 14, um caso simples de *clustering* é ilustrado, onde cachorros de diferentes raças são agrupados por altura e peso. Ao observar o gráfico, fica claro a existência de três *clusters*. Ao identificar as raças correspondentes de cada ponto, verificou-se que, de fato, cada *cluster* corresponde a uma raça específica. Quando existem classes bem definidas e o espaço de dados possui 2 ou 3 dimensões, o *clustering* não é tão complexo, porém, frequentemente, esse não é o caso, pois muitas vezes as classes não são conhecidas e os dados têm *clusters* difíceis de definir [29].

Na Figura 15, um outro conjunto de pontos é agrupado de 3 formas diferentes. Apenas considerando a distribuição visual, pode-se enxergar duas formas de agrupar: na primeira (b), os pontos são separados em dois grandes grupos, já em (d), 6 pequenos grupos podem ser identificados. Porém, essas divisões podem ser nada mais que um efeito causado pela percepção visual humana, sendo possível separá-los também em 4 grupos, como mostrado em (c). Tudo vai depender do objetivo a alcançar e da interpretação dos dados. Sendo assim, o processo de *clustering* não é uma tarefa trivial e requer uma série de procedimentos para garantir uma boa

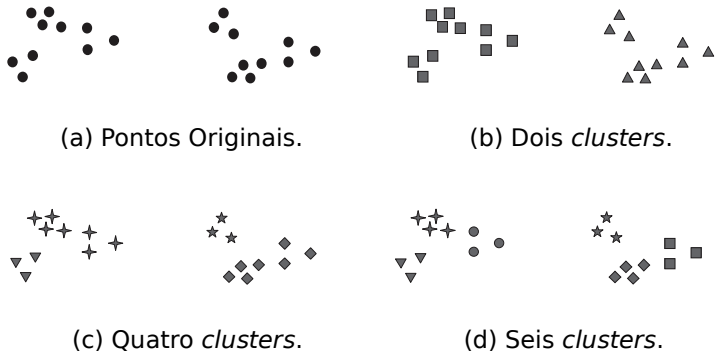
Figura 14 – Exemplo de *Clustering* Simples



Fonte – Rajaraman e Ullman [29]

qualidade nos agrupamentos [30].

Figura 15 – Diferentes Formas de *Clustering*

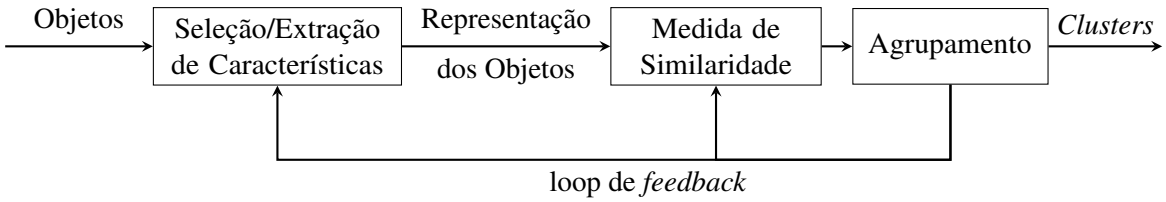


Fonte – Tan, Steinbach e Kumar [30]

Um processo típico de *clustering* é representado na Figura 16. Geralmente envolve os seguintes passos [28] :

1. **Seleção/Extração de Características** – gera uma representação dos objetos estudados para ser usada no *clustering*;
2. **Definição da medida de similaridade** – deve estar de acordo com o domínio dos dados, bem como a dimensão do espaço de dados;
3. ***Clustering*** ou Agrupamento – aplicação de uma das técnicas de *clustering* nos dados definidos no passo 1. Após uma avaliação, pode ser necessário refazer desde o passo 1 (o "loop" representado na figura);

Figura 16 – Etapas do Processo de *Clustering*



Fonte – Adaptado de Jain, Murty e Flynn [28]

A primeira etapa, de "Seleção ou Extração de Caraterísticas", é o processo de identificação do subconjunto de características mais relevantes ao *clustering*. Pode envolver um processo de "Extração de Características", onde são realizadas uma ou mais transformações sobre os dados de entrada para gerar novas características com uma distribuição que forneça informações mais significativas para o *clustering*. Essa etapa é fundamental para se obter *clusters* de qualidade e úteis.

A "Medida de Similaridade" é definida, normalmente, entre pares de objetos, que dependerá fortemente do tipo de dados, dimensões e a abordagem utilizada. Para métodos que consideram os vetores como pontos num espaço euclideano (*K-means* [31]), a medida mais popular é a distância euclideana entre pontos. Enquanto que para métodos que utilizam teoria dos grafos, é usada a distância entre nós. Outro tipo de medida é a distância semântica, como a Distância de *Hamming* [32], usada em aplicações com textos.

Concluído o agrupamento, pode ser necessário processar os *clusters*, ou grupos, gerados de forma a facilitar a sua interpretação. Nesse caso, o processo de *clustering* envolve mais dois passos:

4. **Abstração de dados** – sumariza os resultados em tabelas, gráficos, entre outros;
5. **Avaliação dos resultados** – aplica métodos de avaliação de *clustering* para determinar a qualidade do agrupamento.

A saída do algoritmo de *clustering*, independente do método utilizado, é uma lista de associação entre cada objeto e o número do *cluster* correspondente. É comum que se deseje visualizar os resultados de forma a dar significado aos números. Esta é a etapa de "Abstração de dados", onde os *clusters* são apresentados de uma forma simples e sintética. Caso *clustering* esteja sendo realizado para uma posterior análise por humanos, serão representados de forma intuitiva, pelo uso de tabelas, gráficos ou outra espécie de ilustração. Já no caso de serem utilizados como entrada para um outro programa de computador, serão convertidos para uma representação que facilite o processamento [28].

Por fim, na etapa de "Avaliação dos resultados", os grupos resultantes são avaliados para determinar se o agrupamento foi satisfatório ou não. Caso não o sejam, é necessário

alterar refazer o processo desde a representação dos objetos ou apenas alterando a medida de similaridade e método de *clustering* (loop de *feedback* indicado na Figura 16). Nesse contexto, "bom" ou "de qualidade" são conceitos difíceis de definir e diferentes medidas de validação existem [33]. É comum que essas técnicas utilizem um critério específico ao problema, porém, para se determinar esse critério, muitas vezes uma inferência subjetiva é feita. Não existem padrões ótimos, universais, ou uma "regra de ouro" nessa questão e diferentes parâmetros precisam ser testados até encontrar a solução mais adequada ao problema [28].

2.3.2 *Clustering* de Texturas

O primeiro passo para realizar o *clustering* das texturas de papel foi identificar quais dados poderiam ser gerados para formar o vetor de características. Inicialmente, consideraram-se 35 características das texturas, extraídas de 5 diferentes representações da imagem, a saber: cada canal RGB individualmente, escala de cinza e média $(R + G + B)/3$. Para cada uma destas versões da imagem, sete medidas estatísticas foram aplicadas: média, desvio padrão, moda, valor mínimo e máximo, mediana e curtose.

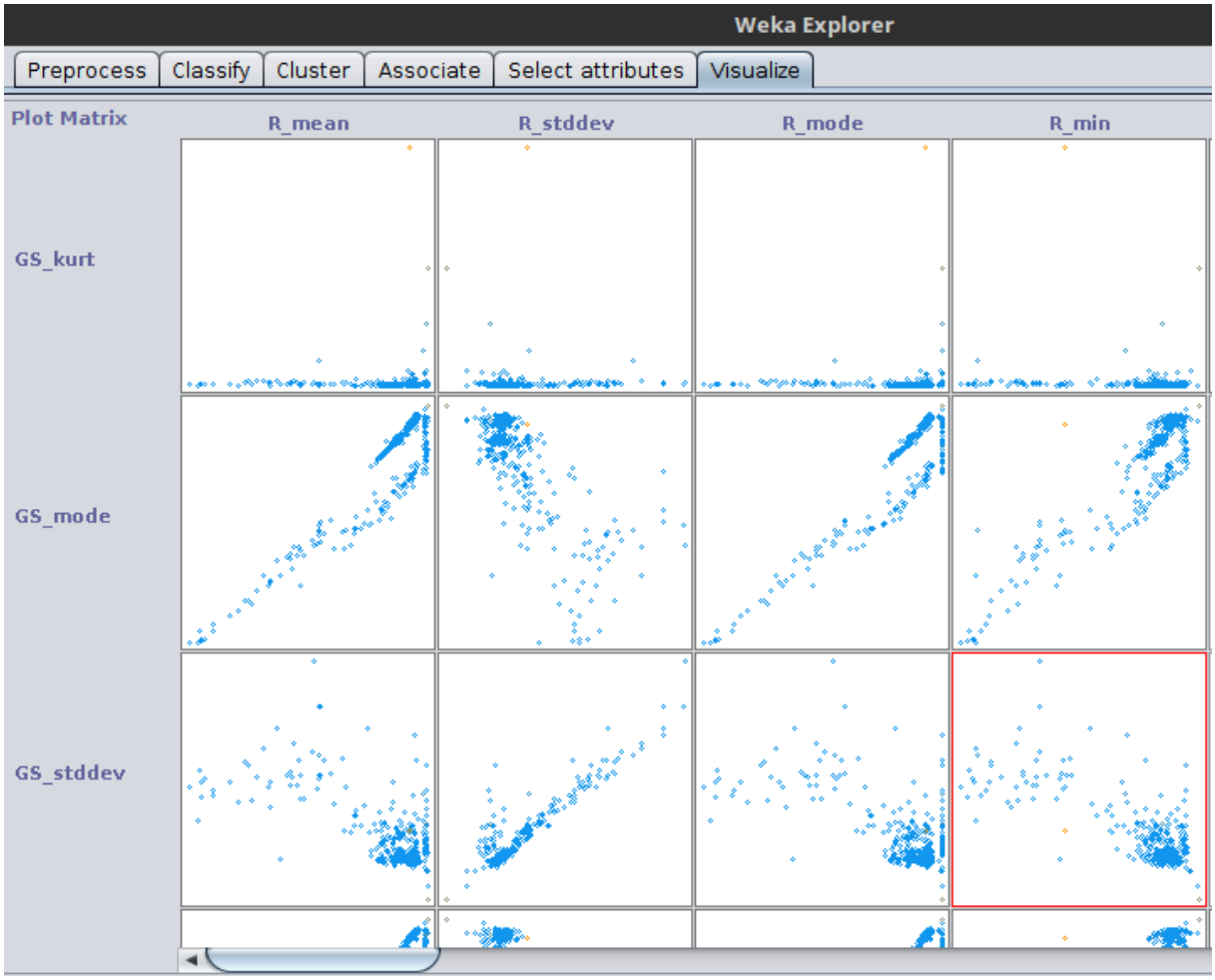
Para decidir quais características utilizar, os recursos do *software* Weka [34] foram explorados. Esse *software* permite variadas visualizações e experimentação com vetores de características. Na Figura 17, é mostrado uma parte da matriz com a plotagem de todas as possíveis combinações, duas a duas, das 35 características consideradas. Apenas por inspeção visual, é possível identificar as medidas que apresentaram uma distribuição muito condensada (como no caso da curtose) ou muito dispersa (como no caso do desvio padrão), para todos os casos em que aparecem, sendo, portanto, descartadas. Além disso, sabe-se, a priori, que existem dois grandes grupos: SBrT e Nabuco, logo as medidas que não apresentaram uma divisão clara em dois conjuntos de pontos também foram descartadas.

Nesse processo de inspeção visual, observou-se que os gráficos cujos componentes continham a moda estatística apresentaram uma distribuição mais favorável ao *clustering*, especialmente a moda dos canais RGB. Na Figura 18, a plotagem dessas três medidas $(R_{mode}, G_{mode}, B_{mode})$ – moda estatística do canal R, G e B, respectivamente, é ilustrada, onde cada ponto representa uma das 3.351 amostras de textura e os valores indicam o nível de cor, que varia de 0 a 255. Como se pode observar, a distribuição é concentrada em dois grandes grupos, um formado por cores mais claras (acima) e outro logo abaixo, formado por cores mais escuras. Além disso, o formato longitudinal indica uma variação gradual na cor, o que implica em diversos subgrupos de textura. Por essas razões, esse vetor foi utilizado.

Uma vez que existem alguns conjuntos mais densos de pontos nas regiões superior e inferior no formato longitudinal, o agrupamento foi divido em duas etapas:

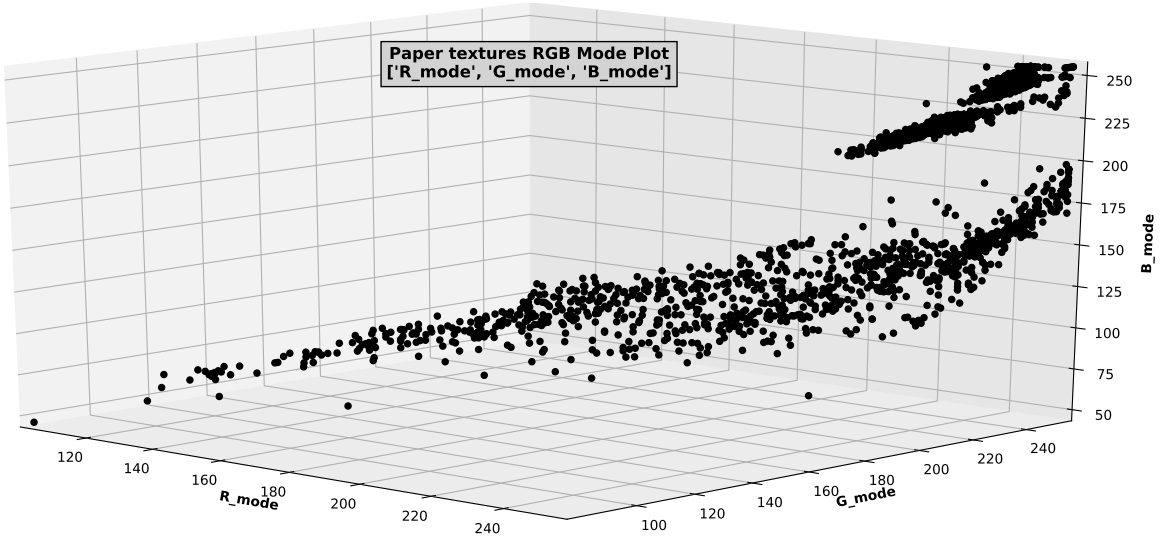
(a) aplicar um método de *clustering* que considera a conectividade dos pontos;

Figura 17 – Matriz de Características com Weka



Fonte – Dados da Pesquisa

Figura 18 – Plotagem da Moda RGB



Fonte – Dados da Pesquisa

(b) identificar os maiores *clusters* resultantes e subdividi-los em *clusters* menores (*re-clustering*), com técnicas que consideram a média da distância entre os pontos.

Os critérios para a escolha das técnicas foram a amplitude de uso e se o tipo de medida de similaridade era apropriado para a etapa. Sendo assim, na primeira etapa, (a), o método hierárquico – *hierarchical cluster analysis* (HCA) [29] foi usado e, para a segunda etapa, (b), tanto o HCA quanto o K-means [31] foram usados.

O HCA consiste em gerar uma hierarquia de *clusters*, onde, a partir de uma divisão inicial, os *clusters* são divididos ou unidos até atingir um critério determinado. Durante o processamento, para decidir se dois *clusters* serão unidos ou separados, é aplicada uma medida de similaridade entre os pontos dos *clusters*. A escolha da medida e do critério que se deseja atingir vai determinar o formato final dos *clusters*. A medida utilizada neste trabalho foi a distância euclideana entre os pontos, por ser a mais popular e ter apresentado resultados satisfatórios, além de ser mais adequada à representação tridimensional dos pontos. Já quanto ao critério, foram considerados o *Average linkage* [35] e *Ward Linkage* [36].

A estratégia *Average Linkage* busca minimizar as distâncias médias entre todos os pares de pontos dos *clusters*, buscando unir os *clusters* que estão mais próximos um do outro. Sendo assim, os *clusters* precisam ser inicializados e, então, o algoritmo é aplicado. A distância entre dois *clusters* A e B é dada pela Equação (2.2). Ou seja, considerando $|A|$ e $|B|$ como o número de elementos presentes em cada *cluster*, a distância é definida pela média das distâncias $d(x, y)$, com $x \in A$ e $y \in B$, entre todos os pares de objetos presentes em A e em B, daí o nome *Average* (média). É indicada para uma distribuição onde se sabe que os *clusters* devem ser formados pelos "pontos mais conectados" e a uniformidade quanto ao tamanho dos *clusters* não é prioridade.

$$\frac{1}{|A||B|} \sum_{x \in A} \sum_{y \in B} d(x, y) \tag{2.2}$$

Uma vez que se sabe que dois grandes grupos existem: Nabuco e SBrT, o HCA com *Average Linkage* foi utilizado para a etapa *(a)*. Para a inicialização dos *clusters*, o grafo dos k-ésimos vizinhos mais próximos é construído. O número k de vizinhos para construção do grafo foi determinado variando-se o valor k de 20 a 80, a passos de 5, observando-se o formato dos *clusters* gerados. Concluiu-se que um número k próximo ao número de *clusters* apresentou resultados mais satisfatórios.

Além do número k para o grafo de conectividade, esse método recebe como parâmetro a quantidade de *clusters* que se deseja gerar. A estratégia de análise utilizada para determinar o número de *clusters* foi observar o comportamento da soma de quadrados dentro dos *clusters* (WSS - *Within Sum of Squares*), à medida que se aumenta o número de *clusters*. O WSS é uma forma de medir a qualidade de um conjunto de *clusters*. Quando o aumento no número de

clusters implicar em uma diminuição pequena no WSS, significa que o aumento não melhorou muito a qualidade e, então, o número de *clusters* é definido.

Tabela 1 – Análise do WSS para Agrupamento de Texturas

i	Número de *Clusters*	WSS	Redução da WSS
9	38	67.954,027	2.551,527
10	39	67.752,015	202,011
11	40	66.915,620	836,396
12	41	64.967,801	1.947,818
13	42	64.778,798	189,003
14	43	64.375,135	403,663
15	44	61.462,224	2.912,911
16	45	61.681,413	-219,189
17	46	63.614,565	-1.933,152
18	47	50.633,003	12.981,562
19	48	50.505,905	127,098
20	49	59.970,019	-9.464,113
21	50	60.490,212	-520,193
22	**51**	**50.153,539**	**10.336,673**
23	52	48.970,865	1.182,674
24	53	49.931,624	-960,758
25	54	49.600,848	330,776
26	55	49.035,433	565,415

Fonte – Dados da Pesquisa, 2017

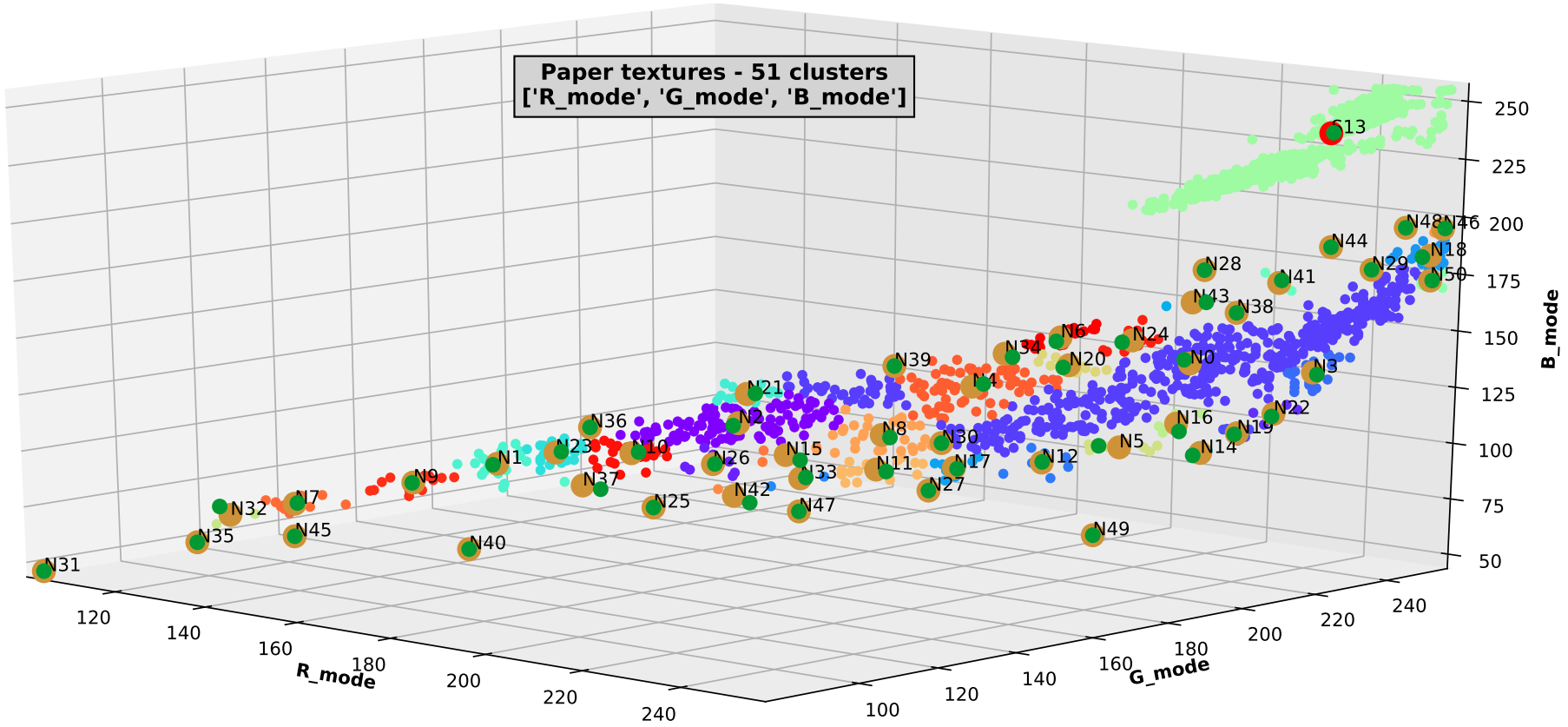
A Tabela 1 apresenta parte das iterações para determinar o número de *clusters* da etapa (a). A coluna "Redução do WSS" indica $WSS_{i-1} - WSS_i$, onde i é número da iteração, de 1 a 50, sendo que o número de *clusters* testados variou entre 30 e 80. Um número negativo na redução indica que a WSS aumentou. Como se pode observar, ao atingir o número de 47 *clusters*, uma grande redução (de $12.981{,}562$) no WSS ocorreu, porém, com 51 *clusters*, uma outra grande redução (de $10.336{,}673$) ocorreu. Observou-se que, ao aumentar o número de *clusters* para valores acima de 51, uma redução tão elevada não ocorreu novamente, mesmo para números acima de 55 (não mostrados na tabela). Portanto, o número de 51 *clusters* foi escolhido. Na Figura 19, página 35, os *clusters* resultantes são destacados no gráfico das texturas.

Analisando o gráfico resultante, pode-se observar que as texturas SBrT localizam-se no canto superior direito, *cluster* `S13`, cor verde claro. Isso indica que possuem níveis de vermelho, verde e azul mais próximos de 255, ou seja, cores claras, como esperado das folhas em branco ou levemente amareladas dos anais. O *cluster* `S13` pode ser visivelmente subdividido em alguns outros menores, uma vez que sua forma longitudinal demonstra ter ao menos dois subgrupos, um acima, mais denso, e outro abaixo, distribuído ao longo do eixo da moda do canal verde, `G_mode`. Já para as texturas de Nabuco, os maiores *clusters* são o `N0` (lilás, no meio), `N2` (roxo, à esquerda), e `N4` (laranja). Uma vez que possuem um formato disperso, pouco denso e

longitudinal, é provável que existam texturas com características significativamente divergentes dentro do mesmo *cluster*.

A etapa (b) consiste em realizar o agrupamento nos *clusters* mencionados acima, ou seja, realizar o *reclustering*. Nesta etapa, deseja-se a formação de *clusters* de tamanho uniforme. Foram considerados o K-means e o HCA com a estratégia de *Ward linkage* [36]. Esta última é um método hierárquico onde o WSS é minimizado. Nesse sentido, de minimizar a variância, este método é similar ao K-means, porém com uma estratégia hieráquica de união dos *clusters*. Para determinar qual método (K-means ou HCA com Ward) seria usado para cada *cluster*, aplicou-se cada um deles em cada um dos *clusters* a serem subdivididos. Tomou-se nota do WSS e observou-se qual método mais reduzia o WSS e subdividia os *clusters* mais uniformemente.

Figura 19 – Agrupamento das Texturas por Conectividade



Fonte – Dados da Pesquisa, 2017

Para se obter sucesso na aplicação do K-means, é preciso que, além do número correto (k) de *clusters*, seja feita uma inicialização que favoreça a redução no WSS. Essa inicialização consiste em selecionar uma quantidade k de pontos que, muito provavelmente, estarão presentes em *clusters* separados. Então, cada um desses pontos é associado a um *cluster* e o algoritmo executa adicionando ou removendo pontos em cada um desses *clusters* iniciais [29]. Existem duas abordagens principais para a escolha dos pontos iniciais:

1. Escolher pontos tão distantes entre si quanto possível;
2. Realizar o *clustering* de uma amostra dos dados, talvez por métodos de *clustering* hierárquico, de forma que haja *k clusters* e escolher um ponto de cada *cluster*.

Um processo comum utilizando a primeira forma é escolher aleatoriamente o primeiro ponto de cada *cluster*. Outra forma é selecionar o primeiro, $k = 1$, aleatoriamente e, então, selecionar os $k - 1$ pontos restantes encontrando o ponto cuja distância mínima entre ele e os pontos já selecionados seja a maior possível. Para tal, o algoritmo chamado `k-means++` [37] é utilizado. O `k-means++` realiza a inicialização dos *clusters* de forma otimizada e, então, procede com a execução normal do K-means. Como demonstrado experimentalmente por Arthur e Vassilvitskii [37], este algoritmo otimiza a acurácia e tempo de processamento do K-means original simplesmente pela execução de uma inicialização cuidadosa e inteligente. O método consiste em [37]:

1. Escolha aleatória do primeiro centro de *cluster*;
2. Obter cada um dos centros c_i subsequentes com probabilidade $\frac{D(x)^2}{\sum_{x \in X} D(x)^2}$;
3. Repetir o passo 2 até que todos os k centros tenham sido determinados;
4. Continuar com a execução do K-means padrão adotando com os *clusters* iniciais selecionados.

Sendo assim, a escolha do primeiro centro de *cluster* irá determinar o restante. Por si só, esse algoritmo apresentou excelentes resultados, mas desejava-se fixar os *clusters* iniciais para que fosse possível repetir os experimentos. Além disso, observou-se que, ao variar o centro inicial, o WSS dos *clusters* resultantes também variava. Portanto, uma série de experimentos foram realizados para cada aplicação onde o K-means foi escolhido, a saber: *re-clustering* dos *clusters* `N0` e `S13` (Tabela 2).

A escolha aleatória é determinada pela semente do estimador. Essa semente foi variada de 0 a 1000 e, para cada valor, o WSS foi calculado. Então, a semente com o menor WSS foi escolhida. Esse refinamento tornou possível, então, repetir os experimentos e aumentar ainda mais a eficácia do K-means. Para os *clusters* mais longos e maiores, `S0` e `S13`, o K-means

apresentou uma maior redução no WSS, já para os outros dois, o hierárquico com *Ward Linkage* foi o melhor. O desempenho dos dois, porém, depende fortemente da inicialização e do número de *clusters*, logo, essa escolha foi feita baseada no número de *clusters* escolhido e no formato dos *subclusters* gerados a partir de cada *cluster* maior.

A determinação do número de *subclusters* para cada um dos *clusters* mencionados foi feita com a análise do WSS e a inspeção visual dos *clusters* formados. Para esta inspeção, foi implementada uma interface de software para permitir uma forma de visualização em que as texturas de cada *cluster* são colocadas lado a lado, formando uma grade (vide Figura 20). Dessa maneira, é possível visualizar o nível de similaridade entre as texturas. Na Tabela 2 é indicado o número de divisões e quais os *clusters* formados, bem como o método utilizado em cada caso.

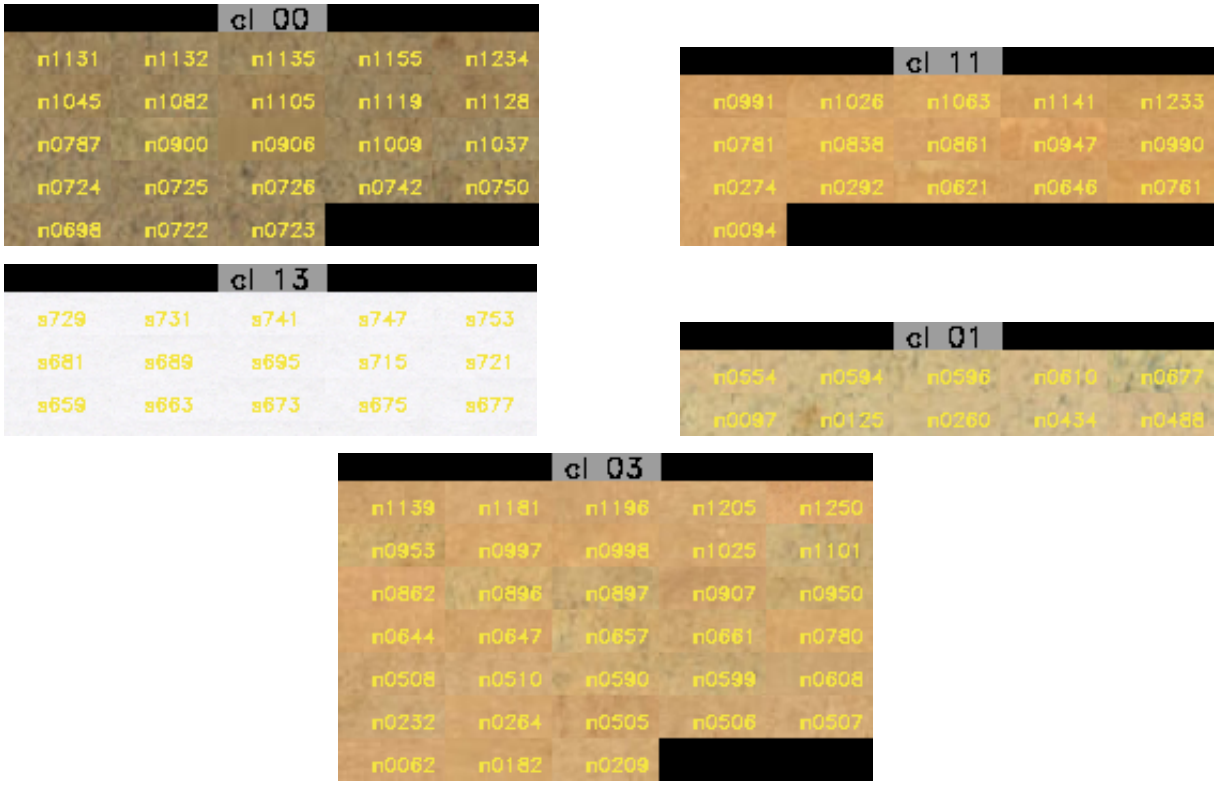
Tabela 2 – *Reclustering* das Texturas

Cluster Original	Número de *Subclusters*	*Subclusters*	Método de *Clustering*
`N0`	20	`N0`, `N51` a `N69`	K-means
`N2`	5	`N2`, `N70` a `N73`	aglomerativo (*Ward Linkage*)
`N4`	5	`N4`, `N74` a `N77`	aglomerativo (*Ward Linkage*)
`S13`	7	`N6`, `N77` a `N83`	K-means

Fonte – Elaboração do autor.

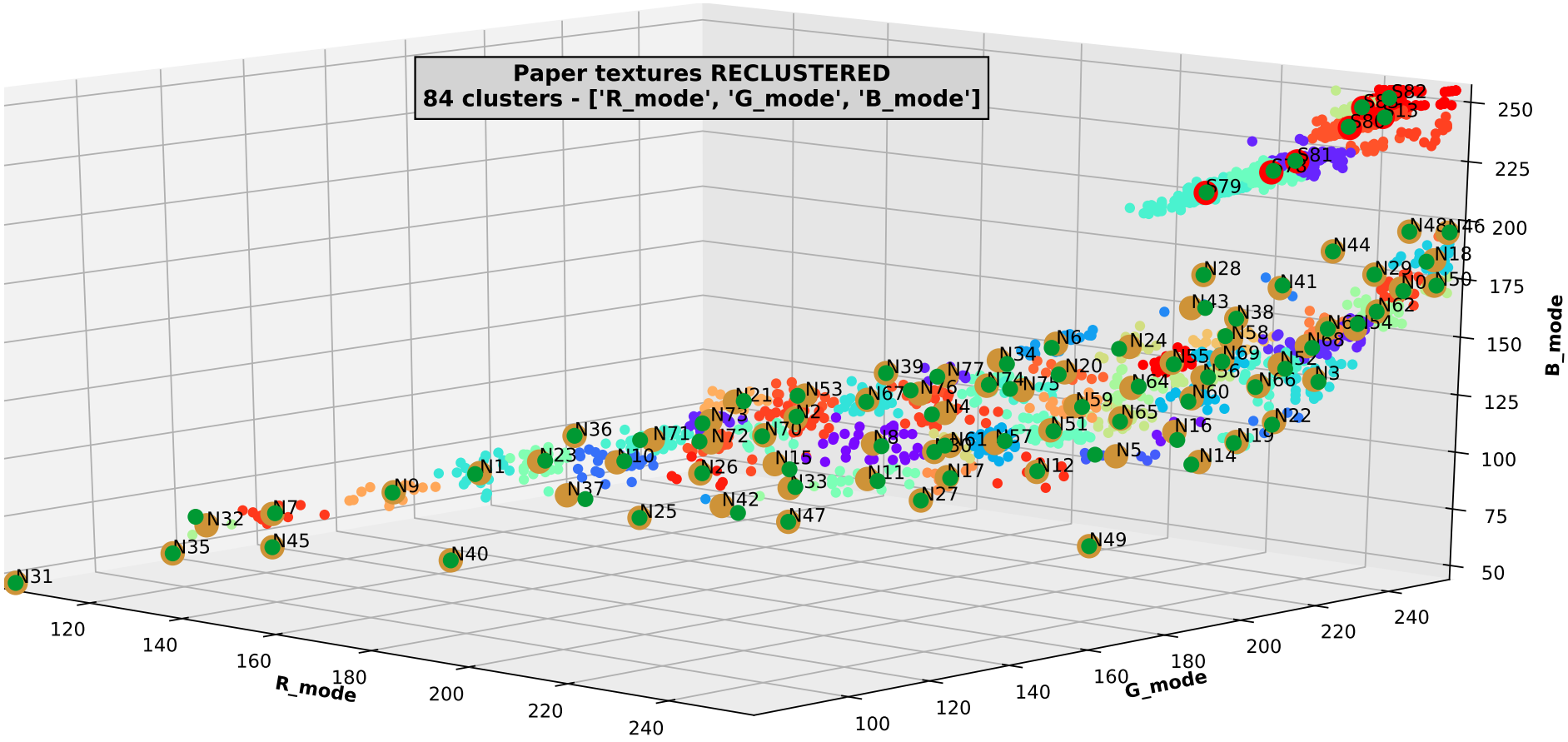
Ao final do processo, 89 *clusters* foram gerados. Na Figura 21, os *clusters* finais são apresentados. Como se pode observar, existem pequenos pontos verde-escuro próximos a pontos dourados ou vermelhos, um pouco maiores. Os pontos maiores, vermelhos e dourados, representam os "centroides", que são os pontos médios dos *clusters*. Já os pontos verdes são os pontos de dados (textura) que mais se aproximam dos centroides. Os centroides dourados são compostos inteiramente de texturas da base Nabuco, enquanto que os vermelhos da base SBrT.

Figura 20 – Grades de Textura de Alguns *Clusters*



Fonte – Dados da Pesquisa

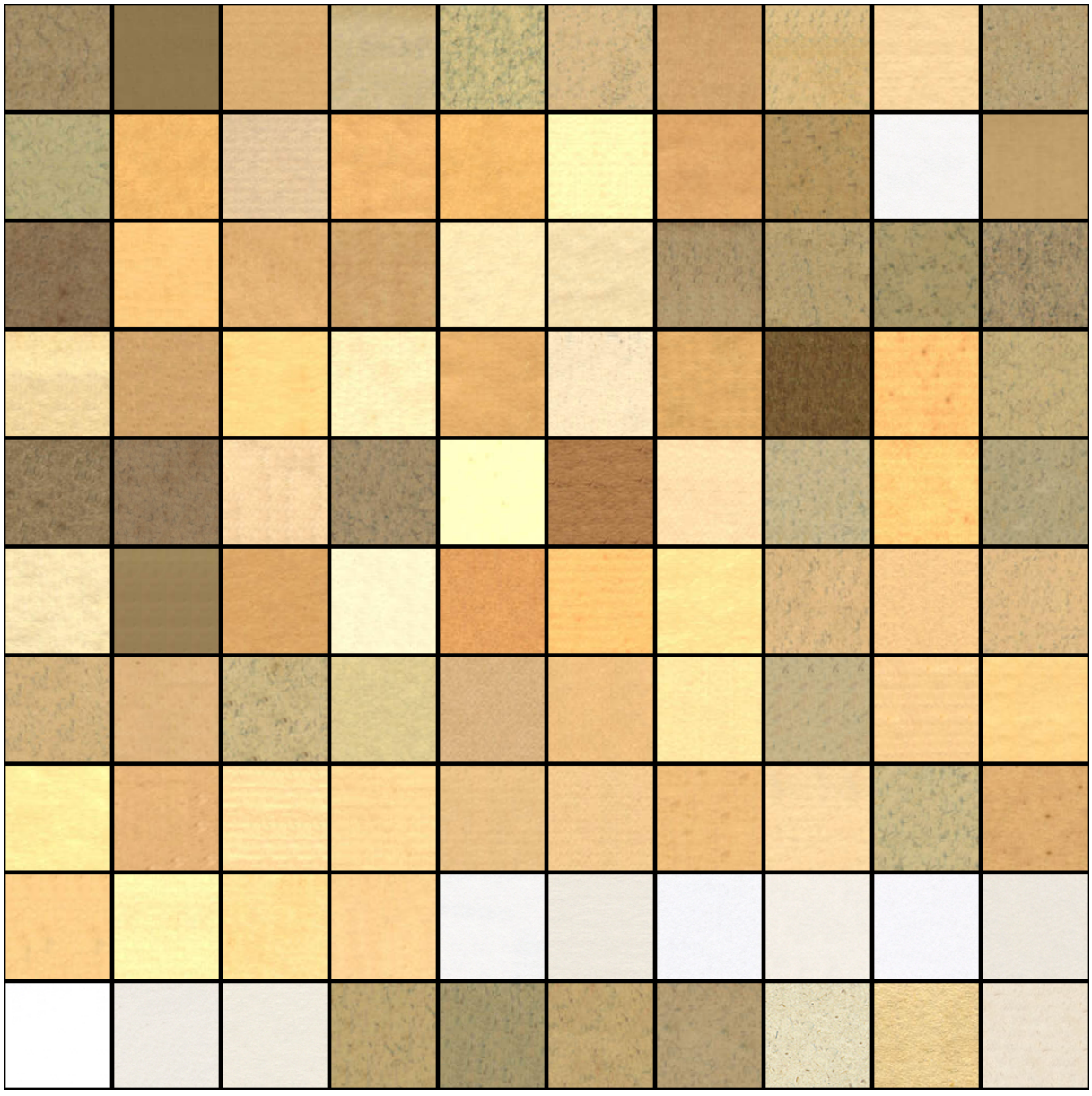
Figura 21 – *Reclustering* por Distância Euclideana



Fonte – Dados da Pesquisa, 2017

Os pontos mais próximos aos centroides formam um conjunto com elementos significativamente diferentes entre si. Sendo assim, as texturas correspondentes a eles são escolhidas para compor o conjunto de texturas base para síntese das folhas de papel. As texturas usadas no *clustering*, porém, possuem as dimensões 50×20 *pixels*, enquanto que os métodos de síntese de textura utilizados se comportaram melhor com amostras de 100×100 *pixels*. Sendo assim, os documentos selecionados foram reamostrados para se adequarem a essa necessidade. Além disso, 11 amostras foram extraídas do conjunto de dados do DIBCO para formar o conjunto final de 100 amostras de texturas, apresentado na Figura 22.

Figura 22 – Texturas de Nabuco, SBrT e DIBCO Usadas na Síntese



Fonte – Próprio Autor

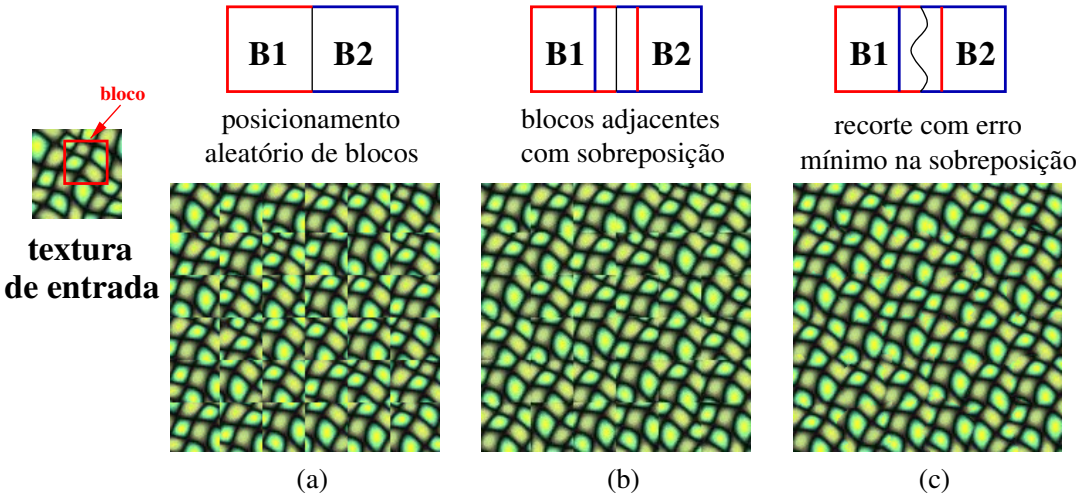
2.3.3 Síntese de Imagens de Folhas de Papel

Cada uma das 100 texturas selecionadas foram utilizadas para gerar uma folha de papel em branco texturizada. Ela é usada para dar cor à imagem de documento sintética, acrescentando o efeito de envelhecimento. Como indicado na Figura 12, página 25, as amostras de textura passam por um processo de síntese que cria uma imagem com características semelhantes, mas de resolução maior. Para cada amostra de textura, duas imagens de folhas de papel de 2.480×3.508 *pixels* foram geradas, uma vez que duas técnicas de geração foram consideradas.

Na primeira abordagem de síntese de textura, a cor de cada *pixel* da textura sintética é aleatoriamente escolhida, dentre os 10.000 *pixels* da textura original (que tem dimensões de 100×100). Isso gera uma imagem que preserva a distribuição de cores da imagem original. O segundo método foi o *Image Quilting* [38], no qual a amostra original é recortada em pedaços que são, por sua vez, encaixados lado a lado até formar uma nova imagem. As imagens são posicionadas de tal maneira que elimina o erro no encaixe.

Na Figura 23, a técnica de *Image Quilting* é ilustrada. Fixando-se um tamanho de bloco S_B (menor que a amostra), considera-se todos os blocos B_i possíveis, variando a posição dentro da textura original. Em (a), os blocos são aleatoriamente escolhidos e encaixados lado a lado até alcançar o tamanho desejado de textura sintética. Em (b), ao invés de sortear uma ordem, uma sobreposição é permitida e o bloco com menor "erro de encaixe" é procurado entre todos os possíveis, a cada novo encaixe. Finalmente, em (c), é permitido que a sobreposição siga uma linha sinuosa e a linha de corte é determinada por um algoritmo de caminho com menor custo, executado sobre a superfície de erro, onde ocorre a junção.

Figura 23 – Síntese por *Image Quilting*

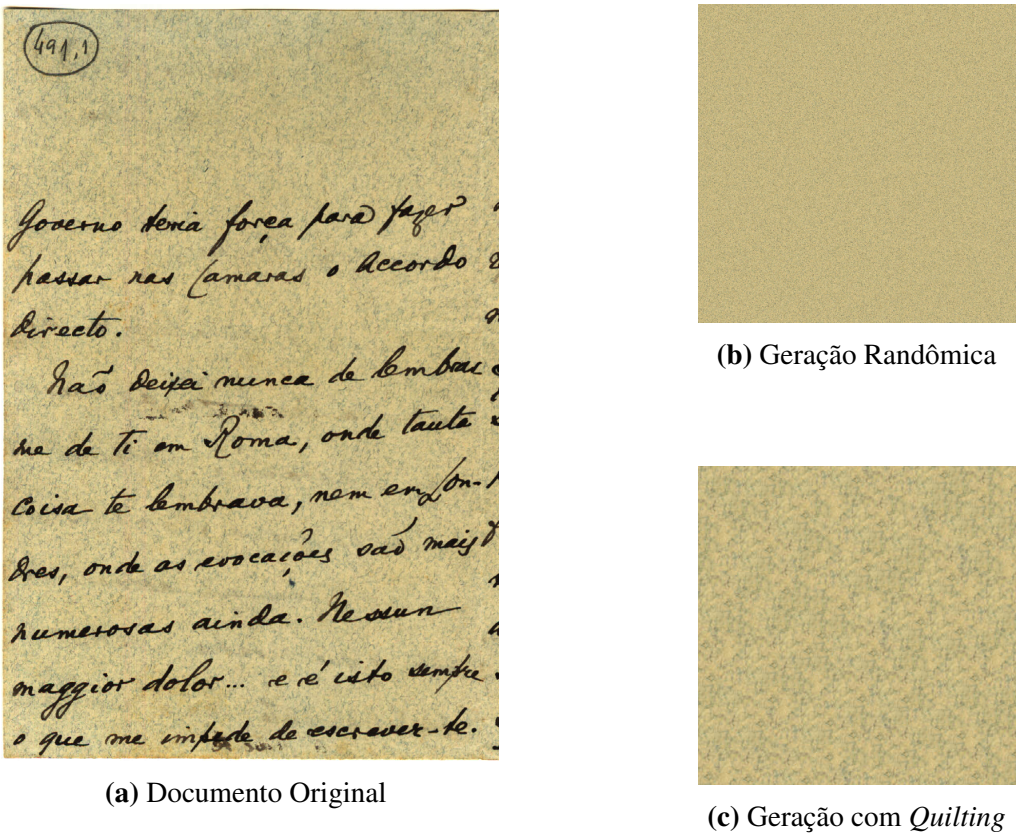


Fonte – Efros e Freeman [38]

Um exemplo de texturas sintéticas geradas utilizando ambos os métodos encontra-se na Figura 24. O documento à esquerda corresponde à textura base utilizada na síntese. A geração

randômica apresenta uma baixa complexidade computacional, enquanto que o *Image Quilting* preserva dados estatísticos da textura original e reproduz os padrões estruturais da textura do documento. É especialmente eficaz no caso de padrões explícitos, como papel com linhas de pauta, murchos ou com dobras. Texturas com ruídos locais, se estiverem presentes na textura de entrada, levam o ruído local a se espalhar por todo o papel, logo foi preciso atentar a isso durante a captura da textura base.

Figura 24 – Exemplos de Geração de Textura



(a) Documento Original

(b) Geração Randômica

(c) Geração com *Quilting*

Fonte – Projeto Nabuco [8] (direita) e próprio autor (esquerda)

Utilizando ambas as técnicas de síntese de textura nas amostras, gerou-se 200 imagens de folhas de papel sintéticas. Durante a síntese, um recorte é realizado na imagem de papel gerada para corresponder ao tamanho do documento sintético que se deseja compor. Após gerar a imagem com os *pixels* da frente, a imagem de interferência frente-verso e gerar a textura no tamanho adequado ao documento que se deseja sintetizar, essas três imagens são mescladas de forma a prover todas as características já mencionadas de um documento sintético.

2.4 Composição do Documento Sintético Frente e Verso

Após a síntese da folha de papel, as duas imagens, frente e verso, precisam compor o documento final. Nesse momento, a imagem do fundo já contém a interferência frente-verso esmaecida, borrada e deslocada. Na imagem da frente, todos os *pixels* que não são 100% brancos

(valor 255 em todos os canais RGB), correspondem à informação textual. Sendo assim, para mesclar frente com fundo, todo *pixel* que não for branco na frente é copiado para uma cópia da folha de papel sintética aplicando-se uma operação especial chamada *darker*, inicialmente proposta em [39].

A operação *darker* é executada *pixel* a *pixel* da seguinte maneira: se um *pixel* da frente tem luminância menor ou igual ao *pixel* da imagem mesclada, ele é copiado para a imagem mesclada. Se não for, o *pixel* da imagem mesclada permanece inalterado. Esse processo é necessário porque as bordas dos *pixels* do texto tendem a ser mais claras que o núcleo das linhas e, em alguns casos, podem se tornar mais claros que os *pixels* da textura onde se vai colar o texto, gerando, nesse caso, imagens de documentos não realistas.

Ao final desta etapa, o processo de síntese da imagem de documento estará concluído. Para uma avaliação apropriada dos algoritmos de binarização, uma ampla diversidade de documentos precisa ser considerada. Portanto, variaram-se os diversos parâmetros mencionados em cada etapa para gerar um banco de imagens sintéticas suficientemente diverso, representativo do universo de possíveis documentos textuais.

2.5 Parâmetros da Síntese

O processo de síntese de documentos recebe como entrada uma imagem de onde serão extraídos os *pixels* do texto, o *ground truth* dessa imagem, uma imagem de folha de papel e valores de embaçamento (*blur*), deslocamento (*shift*) e transparência (*alpha*) da interferência frente-verso. Para cada parâmetro, determinou-se um conjunto de valores que gerassem documentos representativos do universo de documentos textuais. As possíveis configurações para cada documento são:

***Pixels* do Texto** 217 documentos modernos e históricos com seus GT (seção 2.1);

Imagem de Folha de Papel 200 imagens de textura de papel sintéticas (seção 2.3);

Blur 2 níveis: núcleo de 3×3 e 5×5 *pixels*. Usados como entrada para o filtro gaussiano que irá gerar o borramento da interferência (seção 2.2);

Shift 3 valores: 10, 20 e 30 *pixels*. Irão determinar o deslocamento horizontal da interferência;

Alpha 10 valores de *alpha*: de 0,1 a 1,0 em passos de 0,1. Irá determinar o nível de transparência da interferência;

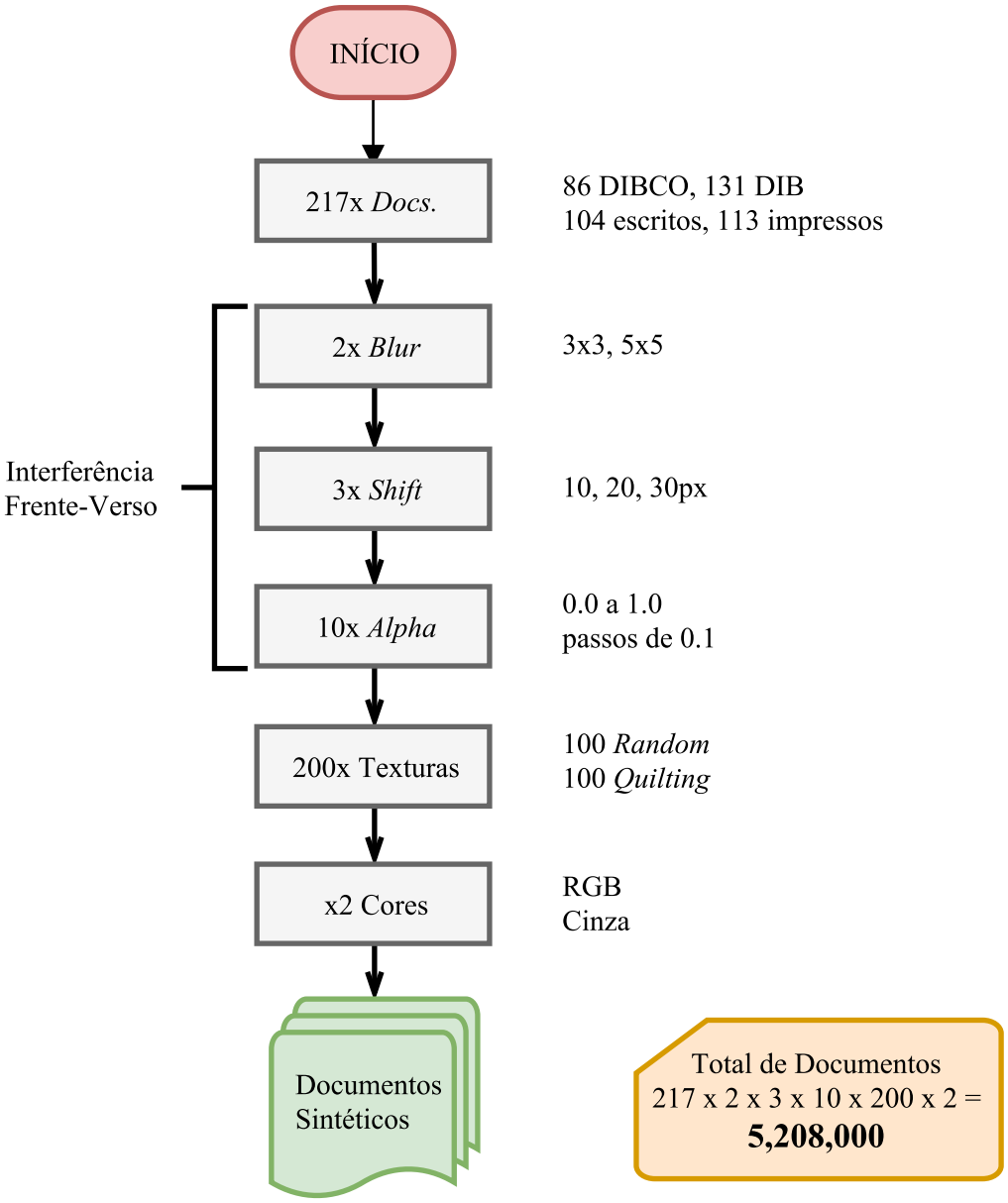
Cor 2 representações de cor: RGB e escala de cinza (luminância).

Sendo assim, o número total de documentos sintéticos que podem ser gerados com os parâmetros mencionados são: $[217 \times texto] \times [200 \times textura] \times [2 \times blur] \times [3 \times shift] \times$

$[10 \times alphas] \times [2 \times cor] = 5.208.000$. Uma representação gráfica desse cálculo encontra-se na Figura 25 e alguns exemplos de imagens geradas, com seus respectivos parâmetros, são apresentados na Figura 26. Devido ao elevado número de possibilidades, seriam necessários vários meses de processamento para obter os resultados para todas as imagens sintéticas.

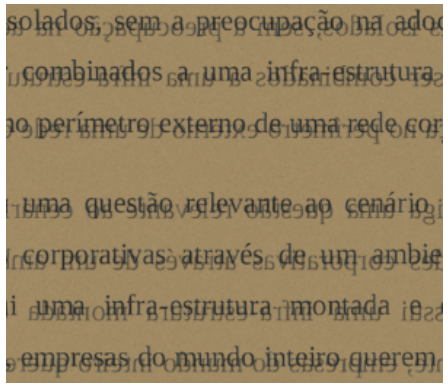
Para que fosse possível processar e compilar todas as imagens em tempo hábil, reduziu-se a quantidade de valores do parâmetro *alpha* da síntese (nível de transparência da interferência frente-verso). Sendo assim, priorizou-se um conjunto de valores de *alpha*, chamados "*alpha* prioritários", que são os níveis de interferências mais prováveis de ocorrerem em imagens de documentos: $0{,}4$; $0{,}6$; $0{,}8$ e $1{,}0$. Dessa maneira, o número total de imagens analisadas foi reduzido em, aproximadamente, 62%, caindo para o total de $2.083.200$ imagens sintéticas. No próximo capítulo, a análise dos algoritmos de binarização é discutida.

Figura 25 – Processamento das Imagens Sintéticas



Fonte – Próprio Autor

Figura 26 – Exemplos de Imagens Sintéticas



Texto Doc. Impresso 87

Textura *Random* no. 1

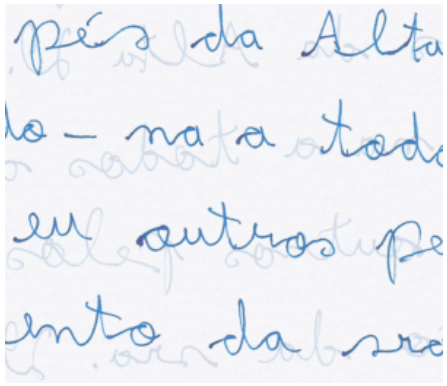
Blur 3×3

Shift 10 *pixels*

Alpha 0.4

Cor Colorido

(a) Imagem Sintética 1



Texto Doc. Escrito 29

Textura *Quilting* no. 87

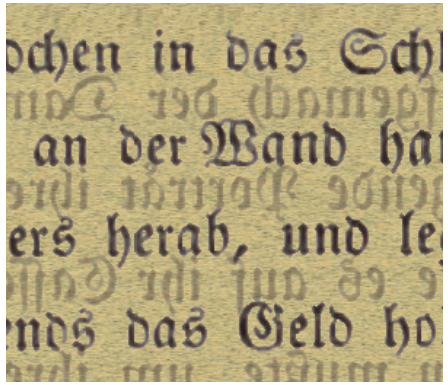
Blur 3×3

Shift 20 *pixels*

Alpha 0.8

Cor Colorido

(b) Imagem Sintética 2



Texto DIBCO'09 Impresso 1

Textura *Quilting* no. 5

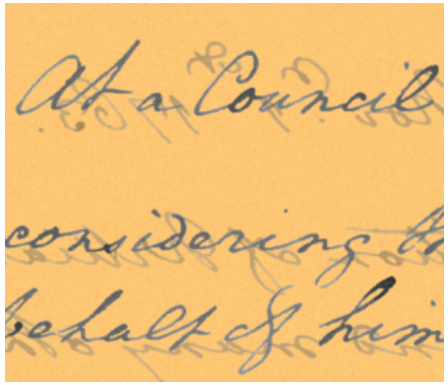
Blur 3×3

Shift 30 *pixels*

Alpha 0.6

Cor Colorido

(c) Imagem Sintética 3



Texto DIBCO'09 Escrito 01

Textura *Random* no. 56

Blur 3×3

Shift 10 *pixels*

Alpha 0.6

Cor Colorido

(d) Imagem Sintética 4

Fonte – Próprio Autor

3 ANÁLISE DE ALGORITMOS DE BINARIZAÇÃO

Desde o surgimento dos primeiros algoritmos de binarização, dezenas de propostas foram e continuam sendo publicadas todos os anos. Vários desses métodos foram concebidos para serem utilizados em um contexto específico, como imagens obtidas de celulares [40] ou imagens médicas [17]. Os estudos de algoritmos de binarização encontrados na literatura, em grande parte dos casos, limitam-se a analisar um pequeno conjunto de imagens de teste. No concurso anual de binarização DIBCO, apenas dez imagens, em média, são utilizadas para a avaliação. Além disso, o contexto em que o algoritmo foi concebido não é considerado. Procura-se determinar um único algoritmo que apresente bons resultados para todo tipo de imagem de documentos. O presente estudo, porém, mostrou que a classificação dos algoritmos, quanto à qualidade, varia de acordo com as características do documento, não sendo possível eleger um método que funcione bem para todos os tipos de imagem.

A abordagem deste trabalho segue o caminho inverso dos estudos encontrados na literatura: ao invés de buscar o melhor algoritmo para todo tipo de imagem, procura-se determinar qual método possui melhor desempenho para uma imagem específica que se deseje binarizar. Para tanto, imagens sintéticas de documentos foram geradas para testar diversos algoritmos de binarização e compor uma base de dados contendo uma ampla variedade de imagens com seus respectivos resultados para cada algoritmo considerado. Essas imagens são tão variadas quanto possível, com o intuito de representar o universo de documentos possíveis, com diferentes tipos de ruídos.

A enorme variedade de tipos de documentos textuais torna extremamente improvável que um único algorítimo seja capaz de binarizar satisfatoriamente todos os tipos de documentos. É mais provável que, dependendo da natureza (ou grau de complexidade) da imagem, vários ou nenhum algoritmo seja capaz de fornecer bons resultados [21]. Algoritmos de binarização são usados em diversas aplicações, portanto é importante conhecer o desempenho de cada um para determinar qual utilizar em cada caso. Porém, eleger um único algoritmo para todos os casos (como é feito na literatura), pode não ser uma boa estratégia.

A pergunta a ser respondida, no âmbito da análise, é “Quais são os melhores algoritmos e seus parâmetros para binarizar a imagem X?”, em oposição ao tradicional “Qual é o melhor algoritmo?”. Por meio dessa abordagem nova, não se obtém uma única resposta, mas um conjunto de respostas. Para tanto, gerou-se um conjunto de imagens de documentos sintéticos que representem o universo de imagens de documentos textuais e os algoritmos de binarização citados neste capítulo foram aplicados. O desempenho, em termos de qualidade e tempo de processamento, foram registrados para cada resultado.

A síntese de um documento requer, primeiramente, um documento real e seu GT para

gerar os *pixels* do texto, amostras de texturas para gerar a imagem do papel e parâmetros específicos que determinarão a aparência final do documento. Devido à elevada quantidade de combinações, desenvolveu-se um *software* para gerenciar a execução dos experimentos.

3.1 Medidas de Qualidade

A principal abordagem para avaliar algoritmos de binarização é comparar a imagem binária gerada pelo algoritmo com o respectivo *ground truth* (GT), a fim de determinar o nível de similaridade entre as imagens. O GT é uma imagem binária gerada em um processo semi-automático de forma a obter como resultado uma imagem binarizada "quase perfeita". Para realizar a comparação do resultado de um algoritmo com o respectivo GT, as medidas mais amplamente utilizadas são o *F-Measure*, *pseudo F-Measure*, PSNR e *Distance Reciprocal Distortion Metric* (DRD), que correspondem às utilizadas no concurso de binarização DIBCO [11].

O *F-Measure* é definido como:

$$FM = \frac{2 \times Recall \times Precision}{Recall + Precision}, \tag{3.1}$$

onde *Recall* e *Precision* são definidos como:

$$Recall = \frac{TP}{TP + FN}, \quad Precision = \frac{TP}{TP + FP}, \tag{3.2}$$

onde TP, FP e FN são definidos como:

TP *True Positive* ou Positivo Verdadeiro – número de *pixels* que são iguais na imagem resultante do algoritmo e no GT, ou seja, corretamente mapeados como texto ou fundo;

FP *False Positive* ou Positivo Falso – número de *pixels* que são pretos na imagem binária do algoritmo, mas brancos no GT;

FN *False Negative* ou Negativo Falso – número de *pixels* que são brancos na imagem binária, mas pretos no GT.

O *pseudo F-Measure* é uma medida baseada no fato de cada caractere possui uma silhueta única, que pode ser representada pela sua esqueletização. O *pseudo F-Measure* é obtido aplicando-se a mesma Equação (3.1), substituindo o *Recall* pelo *pseudo Recall*, que pode ser entendido como a porcentagem do *ground truth* esqueletizado que é detectado na imagem resultante do algoritmo, calculado como segue:

$$p\_Recall = \sum_{x=1,y=1}^{x=M,y=N} \frac{GT(x,y)_{esq} * B(x,y)}{GT(x,y)_{esq}} 100\%, \tag{3.3}$$

onde GT indica a imagem de *ground truth*, B a imagem binarizada por algum algoritmo, (x, y) as coordenadas de uma imagem e GT_{esq}, a versão esqueletizada do *ground truth*.

O PSNR – *Peak Signal to Noise Ratio* é a medida que busca definir o nível de similaridade entre duas imagens. Quanto maior o seu valor, mais semelhantes as imagens serão. Pode ser calculada da seguinte maneira:

$$PSNR = 10log(\frac{C^2}{MSE}), \tag{3.4}$$

onde

$$MSE = \sum_{x=1}^{M}\sum_{y=1}^{N}\frac{(B(x,y) - GT(x,y))^2}{MN}. \tag{3.5}$$

Já o DRD – *Distance Reciprocal Distortion*, procura avaliar a distorção visual na imagem binarizada. A distorção é medida para todos os *S pixels* invertidos como segue:

$$DRD = \frac{\sum_{k=1}^{S} DRD_k}{NUBN}, \tag{3.6}$$

onde DRD_k é a distorção do k-ésimo *pixel* invertido e é calculada usando uma matriz 5×5 de pesos normalizados W_{Nm}, definida em

$$DRD_k = \sum_{i=-2}^{2}\sum_{j=-2}^{2} |GT_k(i,j) - B_k(x,y)| \times W_{Nm}(i,j). \tag{3.7}$$

Nesta dissertação, não são aplicadas medidas de qualidade para classificar os algoritmos. Os resultados para cada imagem sintética são armazenados para que possam ser utilizados em estudos futuros. Para cada imagem, são registrados as proporções de *pixels* pretos e brancos que foram corretamente e erroneamente mapeados, formando a matriz de confusão (Equação (3.8)).

$$\begin{bmatrix} P_{f|f} & P_{f|b} \\ P_{b|f} & P_{b|b} \end{bmatrix} \tag{3.8}$$

A nomenclatura utilizada é como segue:

$P_{f|f}$ Porcentagem de *pixels* pretos corretamente mapeados, ou seja, que são pretos na imagem binarizada pelo algoritmo e no GT;

$P_{f|b}$ Porcentagem de *pixels* erroneamente mapeados como branco, ou seja, são brancos no GT, mas pretos na imagem binarizada pelo algoritmo;

$P_{b|b}$ Porcentagem de *pixels* brancos corretamente mapeados, ou seja, que seão pretos na imagem binarizada pelo algoritmo e no GT;

$P_{b|f}$ Porcentagem de *pixels* erroneamente mapeados como preto, ou seja, são pretos no GT, mas brancos na imagem binarizada pelo algoritmo.

3.1.1 Avaliação com Documentos Sintéticos

Em [41], foi proposta uma nova estratégia para avaliar algoritmos de binarização, especificamente na remoção da interferência frente-verso. Nessa abordagem, a textura do papel é extraída de uma imagem do documento e usada como fundo para geração de documentos sintéticos. As letras da frente são mantidas, mas, à textura, é acrescido texto com transparência reduzida, simulando a interferência desejada. A posição e intensidade das letras do verso é controlada e o impacto em cada algoritmo é medido.

Similarmente, em [12], documentos sintéticos também são usados na análise. Dez documentos em formato PDF, com variadas formatações, são usados como base para a síntese. Ruídos de diferentes tipos, extraídos de documentos do século 18, são adicionados aos PDFs. Os ruídos são: iluminação não-uniforme, interferência frente-verso, manchas e amassados, etc. Os dois tipos de imagens foram combinadas utilizando uma técnica de "superimposição de mosaicos" para mesclagem [12].

O método de análise proposto nesta dissertação segue, em suma, os seguintes passos:

(i) Geração de um banco de imagens com *ground truth* conhecido;

(ii) Geração de um conjunto de texturas de papel extraídos de documentos reais;

(iii) Síntese de folhas de papel utilizando as texturas sintéticas do passo (ii);

(iv) Síntese dos documentos de referência;

(v) Binarização dos documentos gerados em (iv) com o maior número de técnicas de binarização que se consiga;

Em (i), o banco de imagens deve ser composto de documentos impressos, datilografados e manuscritos, históricos e modernos. Em (ii), as texturas devem ser tão diferentes entre si quanto possível. As imagens de folha de papel, geradas em (iii), devem ser sintetizadas com diferentes técnicas a fim de aumentar a diversidade das texturas de fundo dos documentos. Na síntese de documentos de referência (passo (iv)), os *pixels* do texto obtidos dos documentos reais em (i) serão combinados com as folhas de papel sintéticas geradas em (iii) para, utilizando métodos de síntese de ruídos, gerar documentos sintéticos ruidosos semelhantes aos encontrados no dia a dia e em banco de imagens de documentos históricos. Por fim, em (v), as imagens de documentos sintéticos são binarizadas e os resultados para cada algoritmo são registrados.

Uma vez que os *pixels* do texto possuem *ground truth* conhecido (passo (i)), é possível comparar o resultado da binarização aplicada ao documento sintético com a imagem de *ground*

truth correspondente. Sendo assim, uma vasta variedade de documentos pode ser gerada com *ground truth* e a eficácia de cada algoritmo de binarização pode ser melhor estudada.

3.2 Execução dos Experimentos

Após determinar quais as imagens base, os parâmetros de síntese e identificar os vetores de parâmetros com números únicos, partiu-se para a execução dos experimentos. Para tal, um *software* foi desenvolvido especificamente para esse propósito. Este programa recebe como entrada um intervalo de *ids* e, para cada *id* no intervalo, sintetiza a imagem correspondente, binariza com os algoritmos considerados e avalia cada resultado.

Devido ao elevado número de imagens sintéticas, como explanado na seção anterior, as imagens sintéticas a serem geradas foram divididas em grupos, ou *datasets*. Isso facilitou o controle do processamento e compilação dos resultados. Cada uma das possíveis combinações de parâmetros resulta em uma imagem sintética com características únicas, logo, cada imagem (ou vetor de parâmetros) foi identificada com um número único. Com essa abordagem, é possível, com apenas um número, determinar todos os parâmetros da imagem a ser gerada. Essa forma de identificar as imagens sintéticas permitiu a distribuição da execução em um *cluster* de computadores. Além disso, facilitou a validação das imagens geradas e o gerenciamento e atualização do banco de dados da plataforma web.

A plataforma de binarização foi batizada de DIB – *Digital Image Binarization*. Ela consiste de um sistema que coordena a síntese e binarização automatizada de imagens sintéticas. Dado o grande volume de resultados, foi preciso desenvolver um sistema Web para visualizar e navegar pelos dados gerados. Ele pode ser acessado pela URL <https://dib.cin.ufpe.br/>. O DIB foi desenvolvido na linguagem Java, já o módulo que implementa a síntese de documentos propriamente dita foi implementado em `C++`. Além disso, outros três módulos foram desenvolvidos para as implementações dos algoritmos nas suas respectivas linguagens: `C++`, `Java` e `Matlab`.

O DIB recebe como entrada um número de *dataset* ou um intervalo de *ids*. No primeiro caso, todas as imagens do *dataset* informado são processadas. No segundo, apenas as imagens correspondentes aos *ids* informados são processadas. É possível configurar a execução com ou sem *alphas* prioritários. Além disso, é preciso configurar a localização no computador das 213 imagens de documentos de onde os *pixels* da frente serão extraídos.

O ciclo de execução, para cada imagem, consiste de cinco passos: (i) O *id* da imagem é convertido no vetor de parâmetros para a síntese, incluindo a identificação de quais imagens base serão utilizadas. (ii) O documento é sintetizado. (iii) A imagem gerada é binarizada com cada um dos algoritmos considerados, registrando os tempos de processamento. (iv) A imagem resultante de cada algoritmos é comparada com o GT do documento base, usando a medida de matriz de confusão. (v) Os resultados da imagem sintética são compilados em um único arquivo com as medidas de qualidade e tempos de processamento. Além disso, é feito um recorte na

imagem sintética com $1/3$ do tamanho desta, esses recortes são salvos, enquanto que a imagem original, completa, é descartada.

O processamento foi executado utilizando um *cluster* de 15 computadores nas seguintes configurações:

Sistema Operacional Linux Mint

Versão 18.1 Serena

Arquitetura `x86_64`

Processador Intel(R) Core(TM) i7-3610QM CPU @ 2.30GHz

Memória RAM 8 GB

3.3 Algoritmos de Binarização

Os algoritmos de binarização separam os *pixels* em dois grupos: região de interesse, ou frente, e fundo. Recebem como entrada uma imagem em escala de cinza ou colorida e fornecem como saída uma versão em preto e branco (binária) da mesma. Geralmente, os *pixels* pretos representam a informação de interesse. Esses algoritmos aplicam um processo de limiarização, onde cada *pixel* é comparado com um valor pré-definido (limiar, ou *threshold*), para decidir se será convertido em preto ou branco. Estes algoritmos podem ser divididos em duas categorias principais: *globais* e *locais*.

Algoritmos globais baseados em *clustering* [9], minimização de entropia [15] e busca por um vale no histograma de intensidades [42], bem como baseado em características [43] e baseado em modelos [44], têm sido apresentados na literatura. Métodos dessa natureza são eficientes para imagens onde os níveis de cinza dos *pixels* do texto e do fundo são separáveis. Se o histograma do texto se sobrepuser com o do fundo, resulta em uma baixa qualidade de imagens binarizadas [45].

Sezgin e Sankur [23], fizeram uma análise abrangente comparando os algoritmos de binarização, agrupando-os de acordo com a sua natureza. Dos quase quarenta algoritmos apresentados, seis métodos foram tidos como apropriados para trabalhar com documentos com interferência frente-verso: referências [46, 15, 47, 16, 17, 9]. Além desses algoritmos, dois algoritmos baseados na entropia de Shannon foram considerados. Eles foram criados no escopo do Projeto Nabuco para filtragem da interferência frente-verso, propostos em: [48] e [41].

Na Tabela 3, os 22 algoritmos considerados são apresentados, com os respectivos anos de publicação e referências. Em seguida, é apresentada uma breve descrição de cada um desses algoritmos.

Tabela 3 – Algoritmos de Binarização Considerados

No	**Algoritmo**	**Ano**
1	Bilateral [49]	2018
2	Silva-Lins-Rocha [20]	2006
3	Huang [50]	1995
4	Intermodes [51]	2006
5	IsoData [52]	2007
6	Johannsen-Bille [47]	1982
7	Kapur-Sahoo-Wong [15]	1985
8	Li-Tam [53]	1998
9	Mean [54]	1993
10	Mello-Lins [55]	2000
11	Minimum [51]	2006
12	Minimum error [56]	1986
13	Mixture-Modeling [57]	2003
14	Moments [58]	1985
15	Otsu [9]	1979
16	Percentile [59]	1962
17	Pun [46]	1981
18	RenyEntropy [60]	1997
19	Shanbhag [61]	1994
20	Triangle [62]	1977
21	Wu-Lu [17]	1998
22	Yen-Chang-Chang [16]	1995

Fonte – Elaboração do autor.

Bilateral

O algoritmo proposto recentemente por Almeida et al. [49] é executado em quatro passos: filtragem da imagem utilizando o filtro bilateral [63], divisão da imagem em suas componentes RGB, binarização de cada canal RGB utilizando uma versão modificada do método de Otsu e classificação das imagens binarizadas de cada canal para decidir quais das componentes RGB melhor preservaram as informações do texto. O custo computacional é muito mais elevado que seus predecessores e envolve um treinamento para realizar a binarização dos canais.

Silva-Lins-Rocha

O algoritmo de Silva, Lins e Rocha [20] foi desenvolvido como uma melhoria do algoritmo Mello-Lins. Nele, o histograma é interpretado como uma distribuição de uma fonte de 256 símbolos (fonte a priori). Assume-se a hipótese de que todos os símbolos são estatisticamente independentes. No caso de documentos reais, sabe-se que este não é o caso, porém, como descrito na referência [20], isto simplifica o algoritmo e demonstrou ter resultados melhores que seus predecessores.

Huang

No método de Huang e Wang [50], uma técnica de análise de conjuntos *fuzzy* [64] é utilizada para a escolha de um limiar global. A função *membership*, no contexto de análises *fuzzy*, é usada para denotar as relações de características entre um *pixel* e a região à qual pertence (objeto/texto ou fundo/papel). Baseado na medida de *fuzziness*, um intervalo *fuzzy* é definido para encontrar o limiar mais adequado dentro desse intervalo.

Intermodes

O algoritmo proposto por Prewitt e Mendelsohn [51] é um dos métodos globais mais comumente usados [12]. Baseia-se na análise de histograma, onde este é considerado como um "histograma bimodal", com dois picos principais. O histograma é atenuado iterativamente utilizando um filtro de média (*running average*) de tamanho 3 até que restem apenas dois máximos locais, j e k. O limiar t é definido como $t = (j + k)/2$. Foi desenvolvido com o propósito de identificar células em imagens de microscópio, porém apresenta bom resultado para imagens de documentos. Não é apropriado para histogramas que possuam picos extremamente desiguais ou vales muito largos e chatos.

ISODATA

Clustering é um processo de classificação não supervisionado, uma vez que se assume não haver informação a priori disponível. O algoritmo ISODATA – *Iterative Self-Organazining Data Analysis Technique Algorithm*, também pode ser utilizado para binarização de imagens, como na referência [52]. Este método permite que o número de *clusters* seja ajustado automaticamente durante as iterações, mesclando *clusters* semelhantes e separando *clusters* com desvios padrões elevados. O algoritmos é altamente heurístico. No caso do uso do ISODATA para binarizar imagens de documentos, os *pixels* da imagem são colocados em dois *clusters*, correspondentes aos *pixels* pretos e aos brancos.

Johannsen-Bille

Este método [47] utiliza a entropia do histograma da representação da imagem em escala de cinza. Essencialmente, ele divide o conjunto de níveis de cinza em duas partes com o intuito de minimizar a interdependência entre eles.

Kapur-Sahoo-Wong

No contexto de teoria da informação, o algoritmo proposto por Kapur, Sahoo e Wong considera as imagens da frente (texto) e do fundo (papel) como duas fontes distintas, de tal modo que, quando a adição das duas entropias atingir um máximo, o seu argumento t atinge seu valor ótimo. [15]

Li-Tam

O método de Li e Lee [65] resolve o problema da binarização minimizando a entropia cruzada entre a imagem de entrada e a versão segmentada dela. A entropia cruzada é calculada *pixel* a *pixel* entre as duas imagens. Sem fazer inferências a priori sobre a população da distribuição, este método fornece uma estimativa não enviesada de uma versão binarizada da imagem, no sentido de teoria da informação. Alguns anos depois, Li e Tam [53] propuseram uma versão iterativa rápida do mesmo algoritmo e esta é a versão utilizada nas análises.

Mean

Este método simplesmente calcula a média dos níveis de cinza na representação em escala de cinza e usa o valor como limiar para a binarização. Esta abordagem foi estudada por Glasbey em [54].

Mello-Lins

O método proposto por Mello e Lins [55] baseia-se na entropia de Shannon para calcular um limiar global. Foi desenvolvido com o propósito de eliminar a interferência frente-verso.

Minimum

Este método é uma variação do Intermodes [51]. O histograma é considerado como sendo bimodal e o limiar é obtido por meio de atenuações no histograma. Exatamente como no Intermodes, um filtro de média de tamanho 3 é aplicado até que restem apenas dois máximos locais, j e k, porém, ao invés de calcular a média entre os dois (como no método Intermodes), o menor pico é escolhido como limiar.

Mixture-Modeling

O algoritmo *Mixture-Modeling* foi publicado na ferramenta ImageJ [57] no formato de um *plugin*, ou extensão. Ele separa o histograma de uma imagem em duas classes utilizando um modelo gaussiano. O limiar é calculado como sendo a interseção desses dois gaussianos.

Moments

No método *Moments* [58], o limiar é selecionado de forma a preservar o "momento" da versão em tons de cinza da imagem de entrada. Em processamento de imagens, o momento é o resultado da aplicação de um filtro de média (média móvel).

Otsu

Otsu [9] é o algoritmo de binarização global mais bem sucedido da literatura [22]. Considera a versão em escala de cinza da imagem e aplica um limiar para realizar a binarização.

O valor do limiar é escolhido de forma a separar o histograma em duas classes e minimizar a variância entre classes, por meio do uso da análise de discriminantes de Sahoo. Para imagens "limpas", com poucos ruídos ou fundo uniforme, ainda é uma das melhores opções, porém quando ocorrem variações, como manchas, o desempenho cai drasticamente.

Pun

O método proposto por Pun [46] recebe como entrada uma imagem em escala de cinza e a considera como uma fonte com um alfabeto consistindo de 256 símbolos estatisticamente independentes. O limiar é definido como a proporção entre a entropia a posteriori e a entropia total.

Wu-Lu

O algoritmo de binarização de Wu-Lu [17] foi originalmente desenvolvido para binarizar imagens de exames de ultrassonografia, mas apresenta resultados satisfatórios especialmente em imagens de documentos com contraste baixo. Baseia-se na entropia de Shannon e usa a menor diferença entre a entropia dos objetos (texto) e a entropia do fundo como valor de limiar.

Yen-Chang-Chang

O algoritmo de Binarização Yen-Chang-Chang [16] segue a mesma ideia que o algoritmo de Kapur, Sahoo e Wong [15] com relação à distribuição de entropia, porém adotando uma nova definição de correlação entrópica.

3.4 Amostras dos Resultados

Nesta seção são apresentados alguns resultados dos algoritmos de binarização utilizando a metodologia de análise proposta, centrada nas imagens. Nas Tabelas 4, 5, 6 e 7, as imagens mencionadas correspondem às imagens sintéticas da Figura 26, página 46. Nesse estudo, não foram aplicadas medidas de qualidade, mas apenas registrou-se a matriz de confusão, descrita na seção 3.1. Além disso, os tempos de processamento (em segundos) de cada algoritmo também são apresentados. O critério para ordenação dos algoritmos é discutido na próxima sessão.

Para facilitar a visualização dos resultados no formato de tabela, optou-se por abreviar alguns dos nomes de algoritmos. YCC – Yen-Chang-Chan [16]; KSW – Kapur-Sahoo-Wong [15]; RenyE – RenyEntropy [60]; SLR – Silva-Lins-Rocha [20].

3.4.1 Critério de Ordenação

A matriz de confusão fornece quatro medidas, complementares duas a duas – $P_{f|f}$ com $P_{f|b}$ e $P_{b|b}$ com $P_{b|f}$. Sendo assim, a classificação pode ser feita utilizando uma das medidas (acerto ou erro) do número de *pixels* pretos e outra de *pixels* brancos. Na análise apresentada nesta

Tabela 4 – Resultados para Imagem 1

#	**Algoritmo**	$P_{f\|f}$	$P_{b\|b}$	Tempo*	#	**Algoritmo**	$P_{f\|f}$	$P_{b\|b}$	Tempo*
1	Bilateral	91,90	96,50	7,24	**12**	Li-Tam	82,69	97,31	0,32
2	Triangle	90,45	95,61	0,30	**13**	Minimum	82,36	97,37	0,34
3	YCC	89,92	96,07	0,30	**14**	SLR	81,36	97,59	0,36
4	RenyE	89,68	96,23	0,31	**15**	Mean	93,63	81,28	0,34
5	KSW	89,68	96,23	0,34	**16**	Wu-Lu	55,24	100,00	0,31
6	Huang	89,68	96,23	0,34	**17**	Pun	95,25	57,91	0,34
7	JohanB	88,64	96,51	0,34	**18**	Percentile	95,62	51,80	0,35
8	Intermodes	87,03	96,68	0,32	**19**	Shanbhag	47,15	100,00	0,30
9	Moments	85,64	96,83	0,32	**20**	MinError	0,01	100,00	0,41
10	Otsu	84,79	96,94	0,31	**21**	MixtureM	0,00	100,00	0,35
11	IsoData	84,79	96,94	0,34	**22**	Mello-Lins	100,00	0,00	0,33

* Tempo em segundos

Fonte – Dados da Pesquisa

Tabela 5 – Resultados para Imagem 2

#	**Algoritmo**	$P_{f\|f}$	$P_{b\|b}$	Tempo*	#	**Algoritmo**	$P_{f\|f}$	$P_{b\|b}$	Tempo*
1	JohanB	100,00	100,00	0,36	**12**	SLR	100,00	98,09	0,44
2	Minimum	99,99	100,00	0,37	**13**	Triangle	100,00	96,64	0,37
3	Moments	100,00	99,96	0,37	**14**	Li-Tam	96,48	100,00	0,36
4	RenyE	100,00	99,85	0,39	**15**	Mean	100,00	91,95	0,36
5	YCC	100,00	99,78	0,37	**16**	Shanbhag	100,00	87,39	0,37
6	KSW	100,00	99,78	0,37	**17**	Pun	100,00	56,06	0,40
7	Bilateral	99,54	100,00	7,42	**18**	Percentile	100,00	56,06	0,42
8	Otsu	98,95	100,00	0,36	**19**	Mello-Lins	41,56	100,00	0,37
9	IsoData	98,95	100,00	0,37	**20**	Wu-Lu	0,10	100,00	0,37
10	Intermodes	98,95	100,00	0,37	**21**	MixtureM	0,00	100,00	0,39
11	Huang	98,95	100,00	0,39	**22**	MinError	0,00	100,00	0,40

* Tempo em segundos

Fonte – Dados da Pesquisa

Tabela 6 – Resultados para Imagem 3

#	**Algoritmo**	$P_{f\|f}$	$P_{b\|b}$	Tempo*	#	**Algoritmo**	$P_{f\|f}$	$P_{b\|b}$	Tempo*
1	Bilateral	99,60	91,37	0,46	**12**	MixtureM	99,74	81,87	0,03
2	Moments	95,56	94,79	0,01	**13**	Mean	99,82	79,64	0,02
3	Otsu	94,58	95,60	0,01	**14**	SLR	77,98	99,84	0,02
4	IsoData	93,99	95,99	0,01	**15**	JohanB	76,93	99,88	0,01
5	Triangle	98,76	90,16	0,01	**16**	MinError	99,96	64,85	0,03
6	Li-Tam	89,24	98,37	0,01	**17**	Minimum	64,38	100,00	0,02
7	Intermodes	87,65	98,83	0,01	**18**	Wu-Lu	62,11	100,00	0,02
8	YCC	86,77	99,02	0,01	**19**	Shanbhag	57,37	100,00	0,03
9	KSW	86,77	99,02	0,01	**20**	Pun	99,98	56,52	0,02
10	RenyE	85,85	99,19	0,01	**21**	Percentile	99,98	56,52	0,02
11	Huang	99,66	83,99	0,02	**22**	Mello-Lins	52,52	100,00	0,02

* Tempo em segundos

Fonte – Dados da Pesquisa

Tabela 7 – Resultados para Imagem 4

#	**Algoritmo**	$P_{f\|f}$	$P_{b\|b}$	Tempo*	#	**Algoritmo**	$P_{f\|f}$	$P_{b\|b}$	Tempo*
1	Bilateral	100,00	99,75	1,05	**12**	Li-Tam	93,75	98,41	0,01
2	Huang	99,30	95,36	0,02	**13**	MixtureM	99,98	92,09	0,04
3	Mello-Lins	99,46	95,13	0,01	**14**	KSW	90,29	99,15	0,01
4	RenyE	99,86	94,13	0,02	**15**	MinError	100,00	86,10	0,02
5	Otsu	95,76	97,76	0,01	**16**	Mean	100,00	86,10	0,01
6	Intermodes	95,76	97,76	0,02	**17**	SLR	84,68	99,69	0,02
7	Triangle	99,96	93,25	0,01	**18**	Minimum	69,03	99,96	0,02
8	IsoData	95,11	97,99	0,01	**19**	Pun	100,00	58,00	0,02
9	YCC	99,97	93,00	0,02	**20**	Percentile	100,00	58,00	0,02
10	JohanB	99,97	93,00	0,01	**21**	Shanbhag	6,61	100,00	0,02
11	Moments	93,75	98,41	0,01	**22**	Wu-Lu	1,03	100,00	0,02

* Tempo em segundos

Fonte – Dados da Pesquisa

seção, utilizaram-se as medidas de $P_{f|f}$ e $P_{b|b}$. Ainda são necessários estudos para determinar qual a forma de combinação que melhor avalia a qualidade da imagem. As medidas da matriz de confusão podem, porém, serem usadas para o cálculo de medidas de qualidade clássicas, que também são descritas na seção 3.1.

Para a classificação apenas pela matriz de confusão, suponhamos um contexto específico, em que o processo de binarização necessite de uma taxa de **acerto** de *pixels* do **texto** superior a 99%, enquanto que tolere taxas de **erro** de *pixels* do **fundo** superior a 5%. Nessa situação, uma forma adequada para ordenar os algoritmos seria, inicialmente, pelo $P_{f|f}$ e, para aqueles com $P_{f|f} > 99\%$, ordenar pelo $P_{b|b}$. Porém, se a aplicação for mais exigente quanto à acurácia do $P_{b|b}$, a ordenação poderia seguir o caminho inverso.

Por outro lado, a medida de $P_{b|f}$, ou seja, erros no mapeamento dos *pixels* do fundo, levam ao acréscimo de *pixels* pretos no documento binarizado. De modo geral, deseja-se eliminar o máximo desse tipo de erro, pois, no contexto de um sistema de análise de documentos, eles podem confundir, por exemplo, a etapa de segmentação, interpretando linhas de ruído como fazendo parte do texto. Já erros de $P_{f|b}$ implicam na perda de *pixels* do texto que, se não for muito elevada, pode ser compensada em uma etapa de interpolação de *pixels* pretos. Sendo assim, para a maioria das aplicações, deseja-se priorizar, ou maximizar, o $P_{b|b}$, enquanto se mantém o $P_{f|f}$ em uma taxa aceitável.

Nessa dissertação, a ordenação é meramente ilustrativa e não implica numa avaliação precisa da qualidade das imagens binárias. Optou-se por ordenar os algoritmos pela proporção de acerto (P_{acerto}), ou seja, a soma das proporções de *pixels* pretos e brancos corretamente mapeados, como na Equação (3.9):

$$P_{acerto} = P_{f|f} + P_{b|b}. \tag{3.9}$$

3.4.2 Imagens Binarizadas e Discussão dos Resultados

Nesta seção, a partir da página 62, são apresentadas as três primeiras imagens binárias para cada uma das 4 imagens sintéticas consideradas como amostra dos resultados: Imagem 1 (Figura 27 e Tabela 4), Imagem 2 (Figura 30 e Tabela 5), Imagem 3 (Figura 33 e Tabela 6) e Imagem 4 (Figura 34 e Tabela 7). O objetivo da análise é determinar, dentro do contexto de uma aplicação específica, qual o melhor algoritmo para binarizar determinada imagem. As imagens escolhidas para exemplificar o processo de análise e o critério para seleção foi obter um conjunto de 4 imagens com características totalmente diferentes entre si.

A Imagem 1 foi a que manteve mais ruídos, especialmente da interferência frente-verso, uma vez que o nível de interferência foi bastante elevado (transparência, ou coeficiente *alpha*, de 0.4). O primeiro algoritmo, Bilateral, conseguiu eliminar quase que totalmente a textura do papel e manter grande parte do texto. Porém, o Bilateral requer muito tempo de processamento

quando comparado com os outros resultados. Os dois algoritmos seguintes mantiveram um leve ruído da textura do papel e a diferença, visualmente, para o primeiro, foi pequena. Sendo assim, caso haja uma limitação quanto ao tempo de processamento, o mais indicado para a aplicação será o 2º, Triangle.

No caso da Imagem 2, por se tratar de um documento com textura de papel moderno, de boa qualidade, tinta para escrita também moderna, com uma boa qualidade e uma interferência do verso leve, a maioria dos algoritmos obtiveram excelentes resultados, com pequenas diferenças na qualidade. Uma vez que a diferença, tanto na qualidade quanto no tempo de processamento, para os três melhores algoritmos é pequena, o mais indicado para a aplicação seria o 1º, Johannsen-Bille, que apresentou o menor tempo de processamento dos três primeiros e qualidade superior.

Já a Imagem 3 apresentou significativas diferenças entre os resultados. A interferência do verso é moderada e as texturas de papel e imagem de texto foram extraídas de documentos históricos. O algoritmo Bilateral foi aparece em primeiro, o que indica uma qualidade superior pelo critério de P_{acerto}. A Interferência frente-verso foi praticamente eliminada, juntamente com a textura do papel. Ao mesmo tempo, grande parte do texto foi mantido. A diferença entre o 2º e o 3º melhor é mínima, porém o Bilateral requer muito mais tempo de processamento. Sendo assim, caso a aplicação não tolere um tempo da ordem do requerido pelo Bilateral, o 2º, Moments, poderia ser escolhido sem grandes perdas na qualidade da imagem binária gerada.

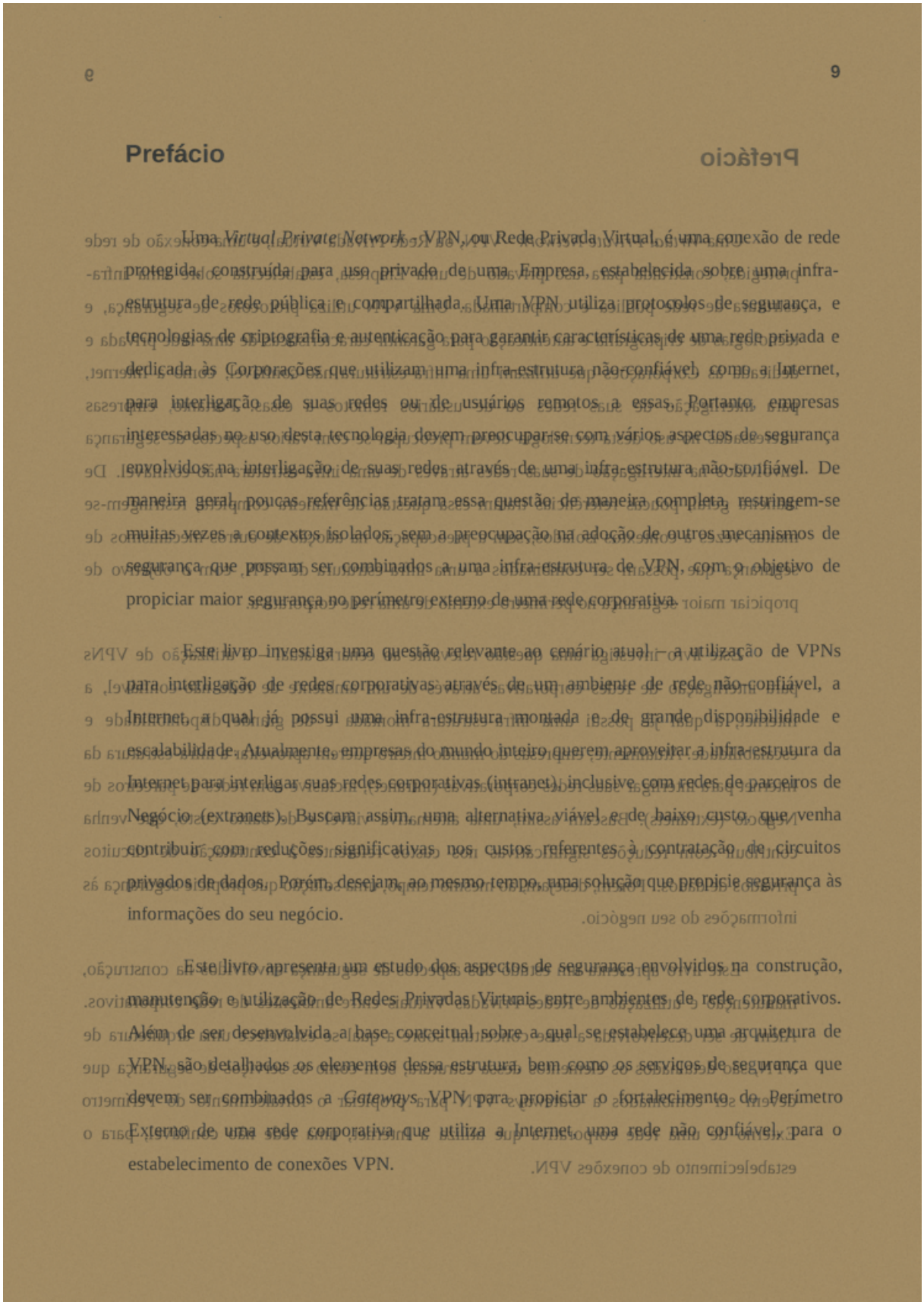
Para a Imagem 4, o primeiro algoritmo foi o Bilateral. Ao realizar uma inspeção visual, de fato este algoritmo gerou uma imagem binária de alta qualidade, porém requer um tempo de processamento dezenas de vezes maior que o segundo e terceiro melhor. Comparando o resultado do 2º com o 3º, nota-se que a qualidade é praticamente a mesma, com uma diferença significativa, porém, no tempo de processamento requerido, pois o Huang requer quase o dobro do tempo necessário para processar com o Mello-Lins. Sendo assim, se houver uma exigência grande quanto ao tempo de processamento, o mais indicado será o Mello-Lins, 3º colocado, mas caso a qualidade do 2º e 3º não sejam suficientes e o tempo de processamento seja aceitável, o Bilateral, 1º colocado, será a melhor escolha.

Como pôde ser observado, a escolha do melhor algoritmo não pode ser realizada de forma unilateral, ou seja, um único algoritmo para todo tipo de imagem. Dependendo das características de cada documento, como ruídos presentes, textura do papel, tipo de tinta e escrita, além da idade, a classificação dos algoritmos varia significativamente. Além disso, para cada classificação, somente após considerar as exigências da aplicação onde a binarização será usada é que o algoritmo mais indicado, de fato, poderá ser escolhido.

Um detalhe importante a se notar é o critério utilizado na classificação. Como mencionado em seção anterior, (3.4.1), o critério usado ($P_{f|f} + P_{b|b}$) é meramente ilustrativo, ainda se faz necessário realizar estudos para determinar como melhor combinar os valores da matriz de confusão e aferir a qualidade da imagem binária. De qualquer maneira, Percebe-se como a variação nas características da imagem de documento altera significativamente o desempenho

dos algoritmos e, portanto, a classificação dos algoritmos no contexto de cada imagem.

Figura 27 – Imagem 1 Completa



Prefácio

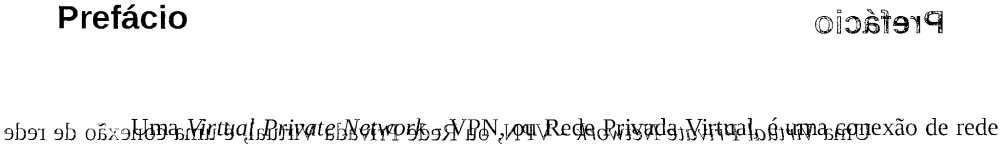
Uma *Virtual Private Network* – VPN, ou Rede Privada Virtual, é uma conexão de rede protegida, construída para uso privado de uma Empresa, estabelecida sobre uma infra-estrutura de rede pública e compartilhada. Uma VPN utiliza protocolos de segurança, e tecnologias de criptografia e autenticação para garantir características de uma rede privada e dedicada às Corporações que utilizam uma infra-estrutura não-confiável, como a Internet, para interligação de suas redes ou de usuários remotos a essas. Portanto, empresas interessadas no uso desta tecnologia devem preocupar-se com vários aspectos de segurança envolvidos na interligação de suas redes através de uma infra-estrutura não-confiável. De maneira geral, poucas referências tratam essa questão de maneira completa, restringem-se muitas vezes a contextos isolados, sem a preocupação na adoção de outros mecanismos de segurança que possam ser combinados a uma infra-estrutura de VPN, com o objetivo de propiciar maior segurança no perímetro externo de uma rede corporativa.

Este livro investiga uma questão relevante ao cenário atual – a utilização de VPNs para interligação de redes corporativas através de um ambiente de rede não-confiável, a Internet, a qual já possui uma infra-estrutura montada e de grande disponibilidade e escalabilidade. Atualmente, empresas do mundo inteiro querem aproveitar a infra-estrutura da Internet para interligar suas redes corporativas (intranet), inclusive com redes de parceiros de Negócio (extranets). Buscam, assim, uma alternativa viável e de baixo custo, que venha contribuir com reduções significativas nos custos referentes à contratação de circuitos privados de dados. Porém, desejam, ao mesmo tempo, uma solução que propicie segurança às informações do seu negócio.

Este livro apresenta um estudo dos aspectos de segurança envolvidos na construção, manutenção e utilização de Redes Privadas Virtuais entre ambientes de rede corporativos. Além de ser desenvolvida a base conceitual sobre a qual se estabelece uma arquitetura de VPN, são detalhados os elementos dessa estrutura, bem como os serviços de segurança que devem ser combinados a *Gateways* VPN para propiciar o fortalecimento do Perímetro Externo de uma rede corporativa que utiliza a Internet, uma rede não confiável, para o estabelecimento de conexões VPN.

Fonte – Resultado da Pesquisa

Figura 28 – Primeiro Resultado para Imagem 1: Algoritmo Bilateral



Prefácio

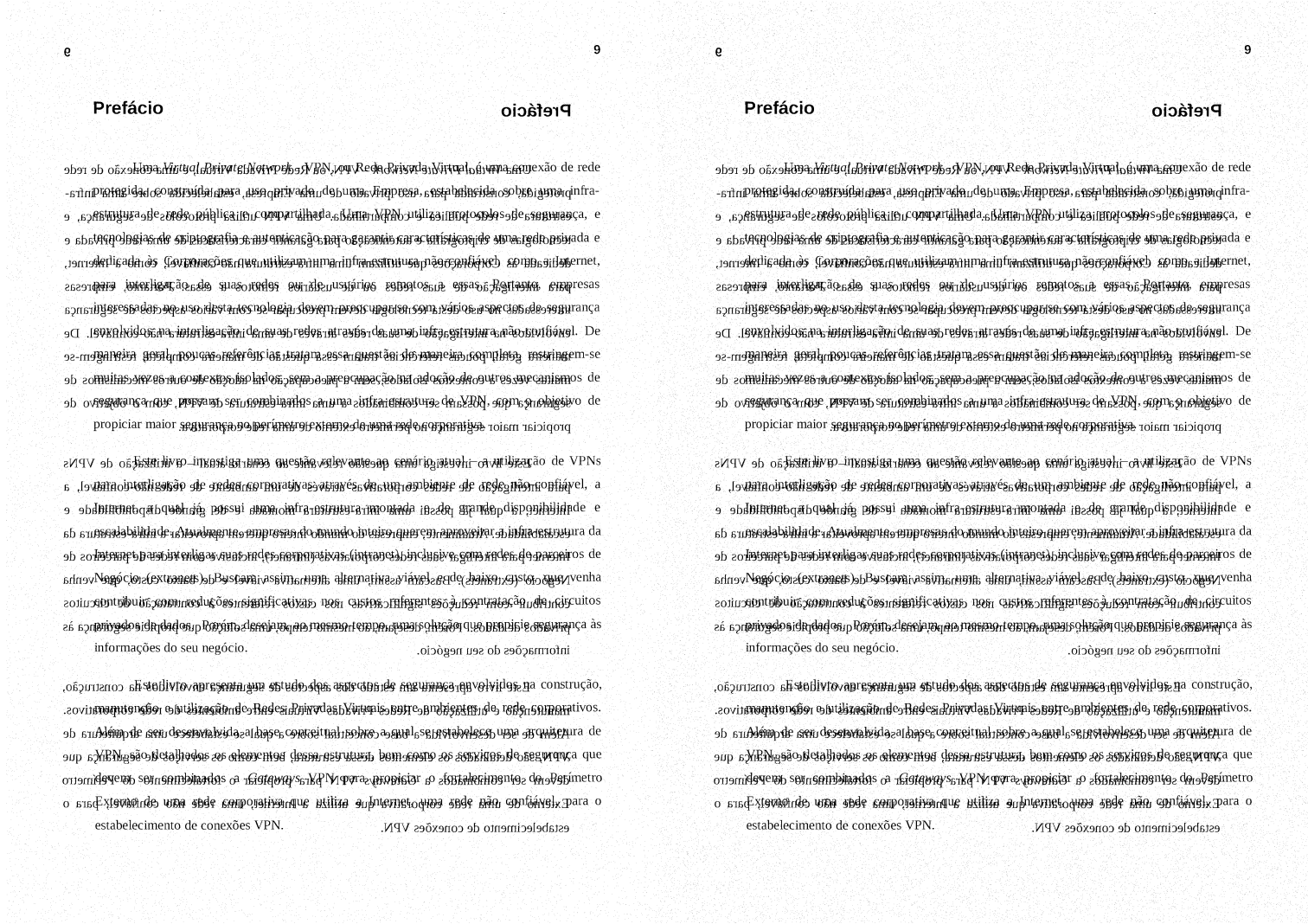
Uma *Virtual Private Network* – VPN, ou Rede Privada Virtual, é uma conexão de rede protegida, construída para uso privado de uma Empresa, estabelecida sobre uma infra-estrutura de rede pública e compartilhada. Uma VPN utiliza protocolos de segurança, e tecnologias de criptografia e autenticação para garantir características de uma rede privada e dedicada às Corporações que utilizam uma infra-estrutura não-confiável, como a Internet, para interligação de suas redes ou de usuários remotos a essas. Portanto, empresas interessadas no uso desta tecnologia devem preocupar-se com vários aspectos de segurança envolvidos na interligação de suas redes através de uma infra-estrutura não confiável. De maneira geral, poucas referências tratam essa questão de maneira completa, restringem-se muitas vezes a contextos isolados, sem a preocupação na adoção de outros mecanismos de segurança que possam ser combinados a uma infra-estrutura de VPN, com o objetivo de propiciar maior segurança no perímetro externo de uma rede corporativa.

Este livro investiga uma questão relevante ao cenário atual – a utilização de VPNs para interligação de redes corporativas através de um ambiente de rede não confiável, a Internet, a qual já possui uma infra-estrutura montada e de grande disponibilidade e escalabilidade. Atualmente empresas do mundo inteiro querem aproveitar a infra-estrutura da Internet para interligar suas redes corporativas (intranet), inclusive com redes de parceiros de Negócio (extranets). Buscam, assim, uma alternativa viável e de baixo custo, que venha contribuir com reduções significativas nos custos referentes à contratação de circuitos privados de dados. Porém, desejam, ao mesmo tempo, uma solução que propicie segurança às informações do seu negócio.

Este livro apresenta um estudo dos aspectos de segurança envolvidos na construção, manutenção e utilização de Redes Privadas Virtuais entre ambientes de rede corporativos. Além de ser desenvolvida a base conceitual sobre a qual se estabelece uma arquitetura de VPN, são detalhados os elementos dessa estrutura, bem como os serviços de segurança que devem ser combinados a *Gateways* VPN para propiciar o fortalecimento do Perímetro Externo de uma rede corporativa que utiliza a Internet, uma rede não confiável, para o estabelecimento de conexões VPN.

Fonte – Resultado da Pesquisa

Figura 29 – 2º e 3º Resultados para a Imagem 1

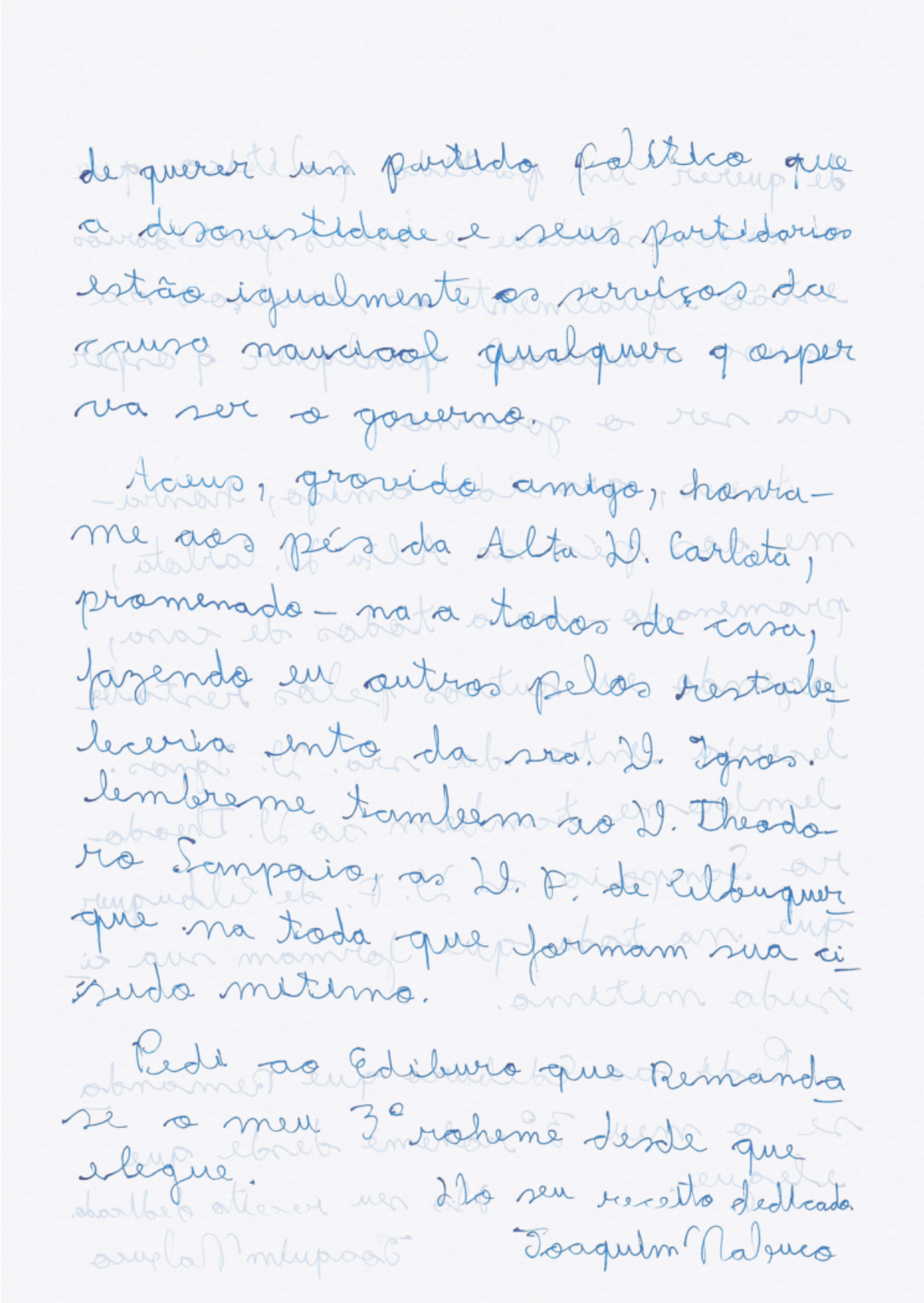


(a) Triangle

(b) Yean-Chang-Chang

Fonte – Dados da Pesquisa

Figura 30 – Imagem 2 Completa



de querer um partido político que a desonestidade e seus partidários estão igualmente ao serviços da causa nacional qualquer q esperança ser o governo.

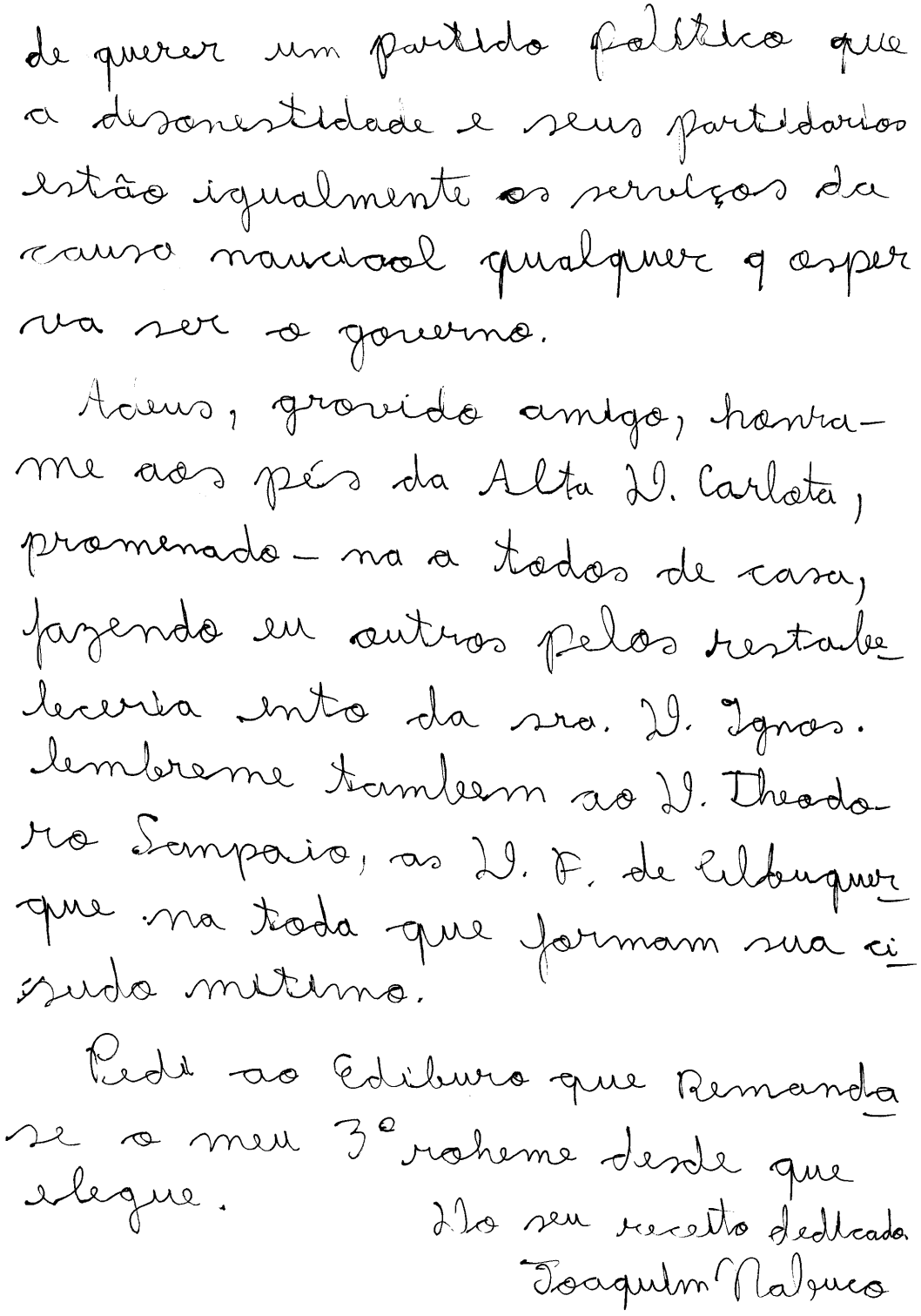
Adeus, querido amigo, honra-me aos pés da Alta D. Carlota, prometendo-na a todos de casa, fazendo eu outros pelos restabelecimento da sra. D. Ignes. lembre-me também ao D. Theodoro Sampaio, as D. P. de Albuquerque que na toda que formam sua círculo mesmo.

Pede ao Ediburo que Remanda-se o meu 3º romance desde que chegue.

Do seu coração dedicado
Joaquim Nabuco

Fonte – Resultado da Pesquisa

Figura 31 – 1º Resultado para Imagem 2: Algoritmo Johannsen-Bille



de querer um partido politico que a desonestidade e seus partidarios estão igualmente os serviços da causa nacional qualquer q esperança ser o governo.

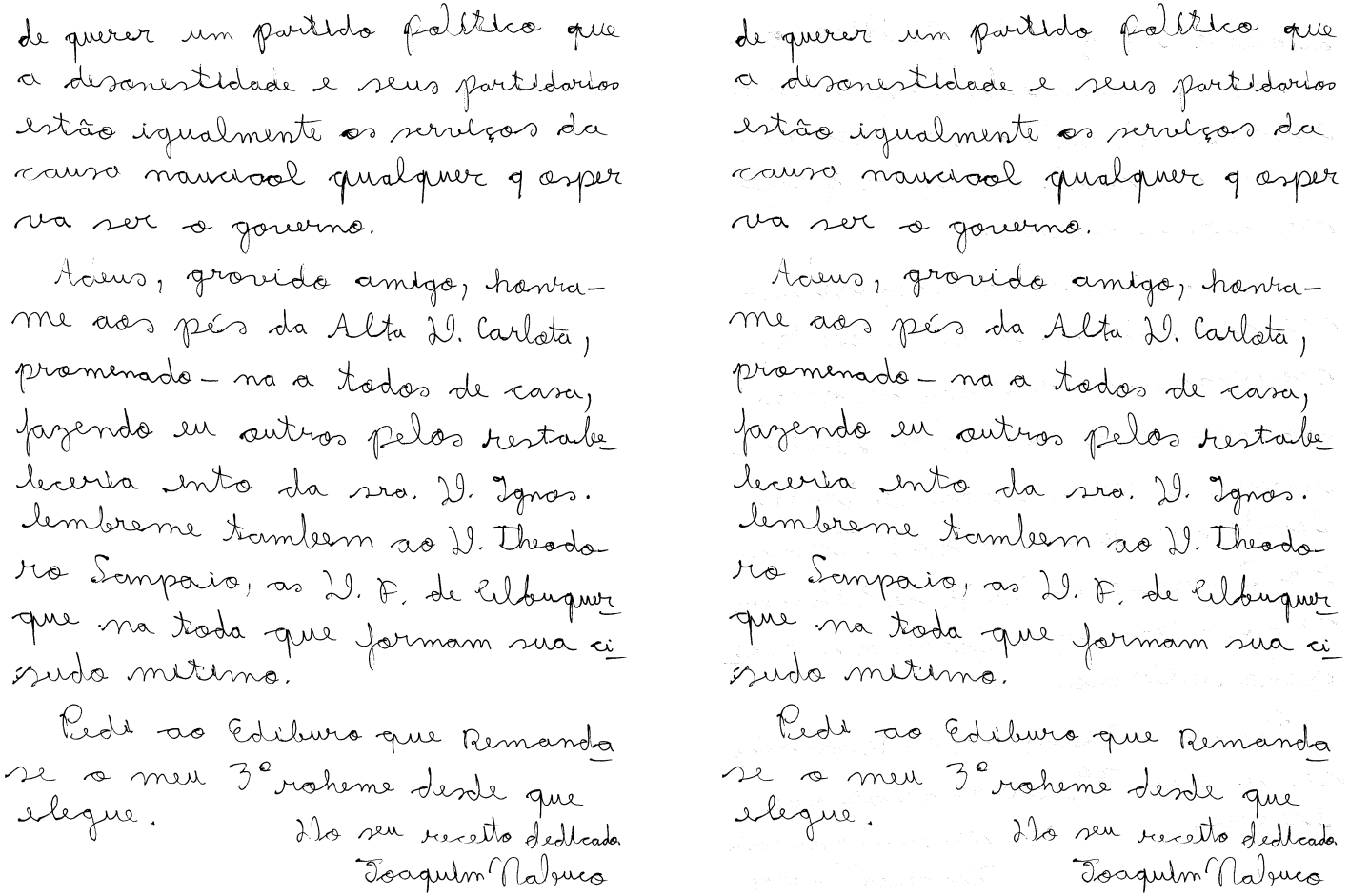
Adeus, grovido amigo, honra-me aos pés da Alta D. Carlota, promenado-na a todos de casa, fazendo eu outros pelos restabelecerão into da sra. D. Ignes. lembreme tambem ao D. Theodoro Sampaio, as D. F. de Albuquer que na toda que formam sua cizudo materno.

Pede ao Ediburo que Remanda se o meu 3º romeme desde que chegue.

Do seu recetto dedicado
Joaquim Nabuco

Fonte – Resultado da Pesquisa

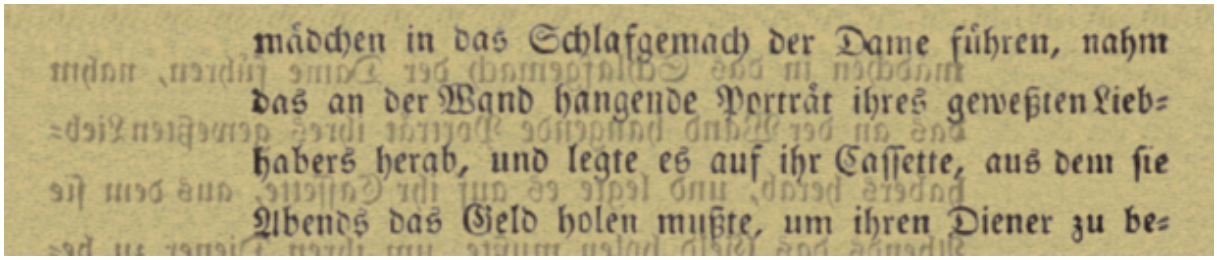
Figura 32 – 2º e 3º Melhores Resultados para Imagem 2



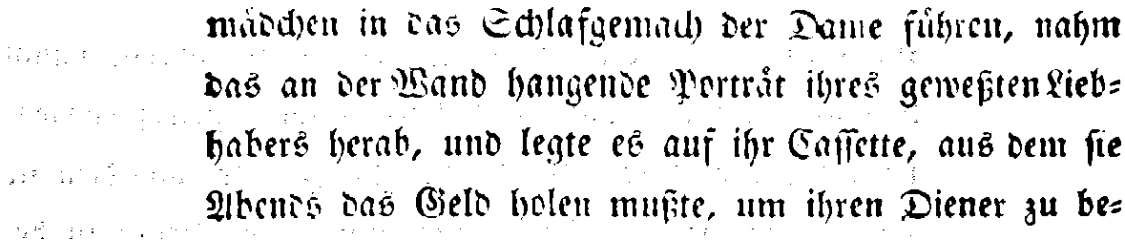
(a) Minimum **(b)** Moments

Fonte – Dados da Pesquisa

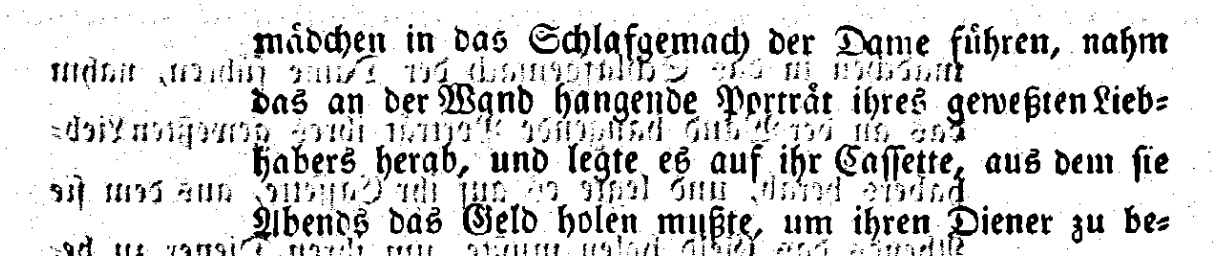
Figura 33 – Resultados para a Imagem 3



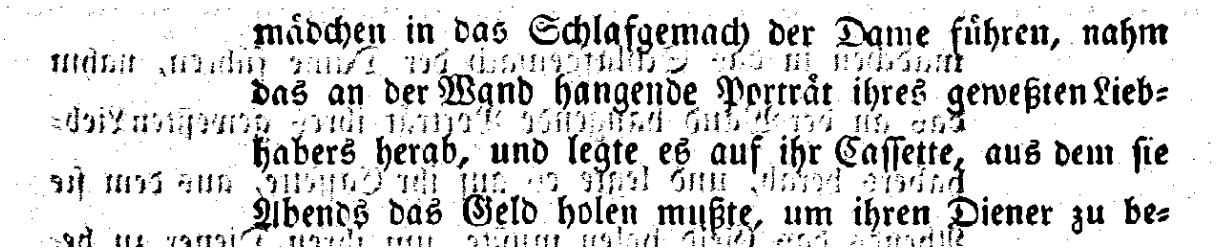
(a) Imagem 3



(b) 1º Resultado: Algoritmo Bilateral



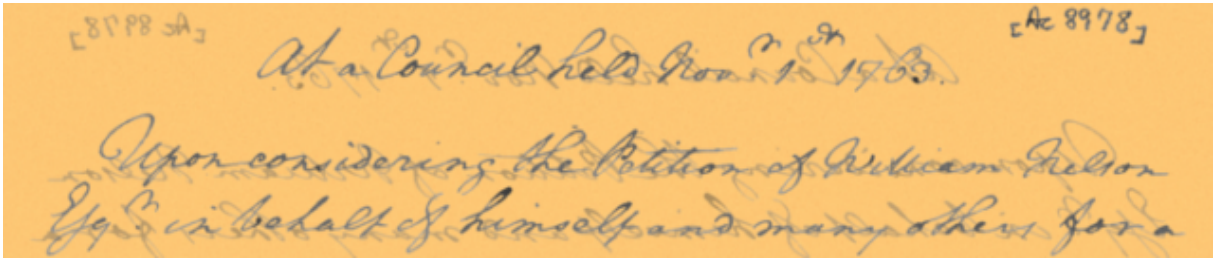
(c) 2º Resultado: Algoritmo Moments



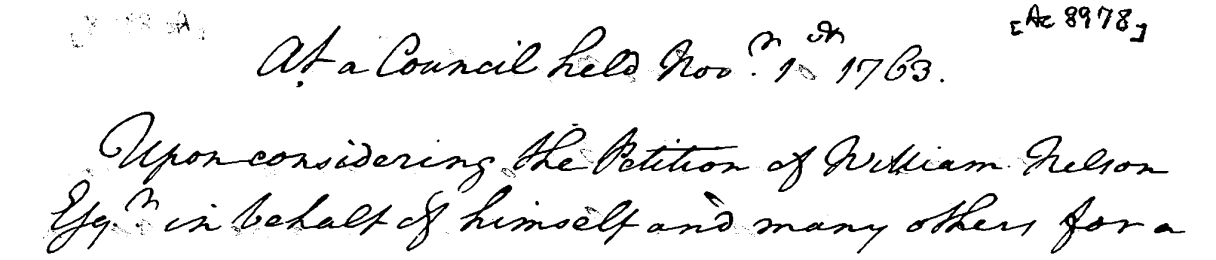
(d) 3º Resultado: Algoritmo Otsu

Fonte – Resultados da Pesquisa

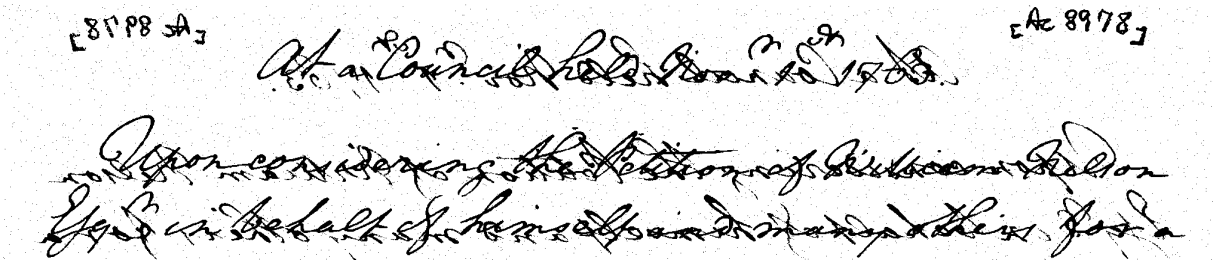
Figura 34 – Resultados para a Imagem 4



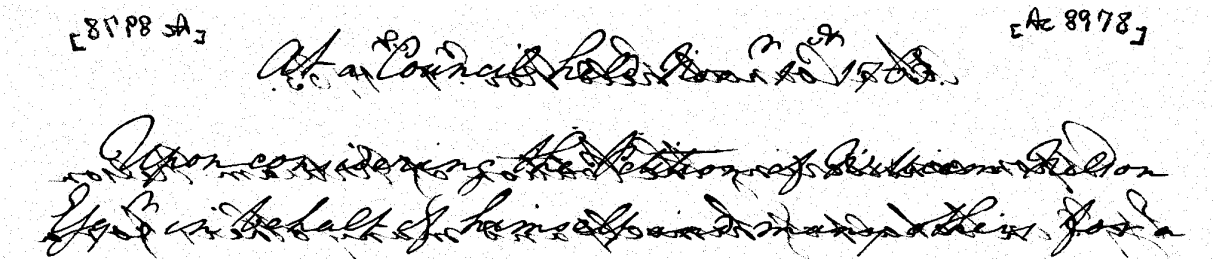
(a) Imagem 4



(b) Melhor Resultado: Algoritmo Bilateral



(c) 2º Melhor Resultado: Algoritmo Huang



(d) 3º Melhor Resultado: Algoritmo Mello-Lins

Fonte – Resultados da Pesquisa

4 CONCLUSÕES

Esta dissertação propõe e implementa uma metodologia para análise de qualidade e tempo de processamento de algoritmos de binarização. A partir de um conjunto de imagens de referência (*ground truth* – GT) conhecido, é possível, de forma automatizada, gerar milhões de imagens sintéticas visualmente semelhantes a documentos reais. Essas, por sua vez, são usadas para avalizar o desempenho dos algoritmos de binarização em termos de qualidade da imagem resultante e tempo de processamento. A estratégia de análise mais comumente utilizada requer a existência de imagens reais com GT para avaliar os algoritmos. Porém, produzir o conjunto de GT que possua representatividade não é uma tarefa trivial. Apesar do crescente número de bases de imagens hoje disponíveis na comunidade, ainda não há uma base que seja ampla o suficiente que represente todo o universo de documentos textuais.

O *clustering* aqui realizado para agrupar as texturas dos documentos também é, na medida do conhecimento do autor, uma abordagem inovadora para realizar a síntese de documentos. As características escolhidas para representar as texturas se mostraram bastante satisfatórias para realizar o agrupamento. Além disso, a combinação de técnicas de *clustering* utilizadas se mostrou favorável à formação de *clusters* contendo texturas semelhantes entre si. Foi possível, então, identificar os principais grupos de texturas, agrupados pela cor mais comum (moda estatística), no conjunto de teste utilizado. Esse tipo de resultado pode ser muito útil em outros contextos de análise de imagens de documentos onde a textura do papel seja uma fator decisivo.

Um total de 22 algoritmos de binarização foi considerado e, ao que se pôde observar, apenas alguns poucos desses, de fato, se comportam bem com a binarização de documentos. Dentre aqueles que apresentam bons resultados para ao menos alguma imagem, é notável a variabilidade da eficácia de acordo com o tipo de documento. Isso mostra como é importante avaliar o desempenho dos algoritmos no processamento da imagem, como proposto nesta dissertação. Além disso, verificou-se a importância de considerar o tempo de processamento, uma vez que um algoritmo como o Bilateral, por exemplo, gerou imagens binárias de elevada qualidade, mas possui um custo computacional também muito elevado, que pode tornar o seu uso inviável para determinadas aplicações.

Para garantir que avaliar imagens sintéticas fornece resultados confiáveis, comparou-se o desempenho dos algoritmos com um pequeno conjunto de imagens sintéticas visualmente semelhantes a documentos que se conhecia o *ground truth*. Os resultados foram semelhantes para o conjunto considerado, o que mostra que essa metodologia é cabível para analisar algoritmos de binarização. É fato, porém, que, devido ao elevado número de possibilidades, não se pode afirmar que todas as imagens sintéticas sejam válidas.

Visto a escassez de trabalhos com a estratégia de síntese de imagens para geração de base

de imagens com *ground truth*, este trabalho traz uma contribuição que se supõe importante para a comunidade. A síntese de imagens é uma área emergente e que promete bastantes avanços nos próximos anos. Novos trabalhos estão surgindo e continuarão a surgir com o intuito de fornecer meios de sintetizar imagens tão reais quanto se consiga e, ao mesmo tempo, gerar imagens de documentos com GT conhecido em larga escala.

A principal contribuição deste trabalho foi, então, fornecer um meio para gerar imagens de documentos com *ground truth* automaticamente, em larga escala. Face à importância desse tipo de base de dados, um trabalho de análise de documentos que precise treinar um classificador pode se beneficiar largamente de tal mecanismo. Além disso, desenvolvedores de algoritmos poderão explorar documentos que não se encontram na sua base original de imagens e descobrir os tipos de imagens para os quais o algoritmo em questão apresenta melhores resultados.

4.1 Trabalhos Publicados

Os seguintes trabalhos estão associados a esta dissertação:

1. LINS, R. D., ALMEIDA, M. M., **BERNARDINO, R.**, JESUS, D., OLIVEIRA, J. M. Assessing Binarization Techniques for Document Images. *Proccedings of the 2017 ACM Symposium on Documento Engineering* – DocEng '17. 2017. p. 183-192. Referência [21]

2. LINS, R. D., **BERNARDINO, R.**, OLIVEIRA, J. M.. Binarizing Document Images Acquired with Portable Cameras. *14th IAPR International Conference on Document Analysis and Recognition*. IEEE, 2017. p. 45–50. Referência [66]

3. ALMEIDA, M. M.; LINS, R. D.; **BERNARDINO, R.**; JESUS, D. A New Binarization Algorithm for Historical Documents. *Journal of Imaging*, Special Issue, p. 1–12, 2018. Referência [49]

Todos os artigos listados estão em apêndice a esta dissertação.

4.2 Trabalhos Futuros

O principal desdobramento que se pode dar a este trabalho é utilizar a grande massa de resultados para treinar um classificador de tal modo que, dado uma imagem de entrada, seja possível determinar qual a imagem sintética que mais se assemelha à imagem de entrada. Os algoritmos mais indicados para a imagem sintética selecionada deverão ser também, em teoria, os mais indicados para a imagem de entrada.

Como parte do processo de desenvolvimento do classificador, é preciso realizar uma análise estatística das imagens sintéticas geradas. Ao se realizar a comparação entre uma imagem real e outra sintética, quais medidas podem ser realizadas para aferir sua similaridade? Além

disso, as técnicas de síntese atualmente presentes na plataforma são muito limitadas e poucos tipos de ruídos podem ser gerados. Logo, os tipos de documentos que seriam corretamente classificados é limitado aos que são semelhantes às texturas e tipos de ruídos considerados. Uma melhoria necessária é o desenvolvimento de novas técnicas de síntese de textura e *pixels* do texto, além da implementação de meios para gerar ruídos como manchas, rasgados, entre outros.

Outro ponto pouco explorado neste trabalho foram as medidas de eficácia, aqui considerada apenas a proporção de *pixels* corretamente associados a preto ou branco. Esta medida fornece um excelente meio de aferir a corretude da binarização, porém quando se deseja classificar entre diferentes resultados, não há meios para determinar quando se deve priorizar o mapeamento dos *pixels* pretos ou brancos. Um estudo nesse sentido, que considere outros fatores além das proporções, pode ser realizado para melhor comparar os resultados dos algoritmos.

Foram considerados apenas 22 algoritmos, quando, na literatura, pode-se encontrar dezenas de outras estratégias de binarização. O critério de escolha para os algoritmos estudados foi o de disponibilidade do código. Uma busca extensiva pode ser realizada para listar e acrescentar o maior número possível de algoritmos à plataforma. Além disso, dentre os algoritmos já considerados, alguns requerem a entrada de um ou mais parâmetros de configuração que foram otimizados pelos autores originais, mas que não necessariamente são os melhores para todo tipo de imagem. É importante considerar, além de um maior número de algoritmos, uma variedade de parâmetros para cada algoritmo.

Por fim, uma vez desenvolvido o classificador, é importante disponibilizá-lo para a comunidade, de forma acessível, para testes diversos. A Plataforma DIB possui uma interface web que dá acesso aos resultados da binarização das imagens sintéticas, logo ela pode ser estendida para acrescentar o módulo que receberá imagens de usuários e encontrará qual a imagem sintética que mais se assemelha à imagem de entrada, fornecendo os resultados da imagem sintética encontrada.

Ou seja, resumidamente, os possíveis passos a se seguir para complementar este trabalho são:

- Desenvolver um sistema de classificação que permita, de forma automatizada, encontrar qual a imagem sintética que mais se parece com uma imagem de um documento qualquer;
- Implementar um módulo na Plataforma DIB que permita que o usuário binarize uma imagem qualquer;
- Desenvolver novas técnicas de síntese de textura do papel e *pixels* do texto;
- Desenvolver métodos para gerar uma ampla variedade de tipos de ruídos comuns em documentos digitalizados;

- Desenvolver uma heurística para classificar os algoritmos baseados nas medidas de acerto P_{bb} e P_{ff};
- Implementar, ou acrescentar implementações, de uma variedade maior de algoritmos de binarização;
- Considerar diferentes parâmetros para um mesmo algoritmo, quando for o caso.

REFERÊNCIAS

1 VICENTINO, C. *História Geral*. São Paulo: Scipione, 2004. Citado na página 10.

2 MELLO, C. A. B. de. *Filtragem, Compressão e Síntese de Imagens de Documentos Históricos*. 119 p. Tese (Tese de Doutorado), 2002. Citado 2 vezes nas páginas 10 e 11.

3 GONZALEZ, R. C.; WOODS, R. E.; MASTERS, B. R. Digital Image Processing, Third Edition. *Journal of Biomedical Optics*, v. 14, n. 2, p. 029901, 2009. Citado 2 vezes nas páginas 11 e 12.

4 O'GORMAN, L.; KASTURI, R. *Document Image Analysis*. [S.l.: s.n.], 1997. v. 26. NP p. Citado na página 12.

5 Jailin Reshma, A. et al. An overview of character recognition focused on offline handwriting. *ARPN Journal of Engineering and Applied Sciences*, v. 11, n. 15, p. 9372–9378, 2016. Citado 2 vezes nas páginas 12 e 13.

6 KASTURI, R.; O'GORMAN, L.; GOVINDARAJU, V. Document image analysis: A primer. *Sadhana*, v. 27, n. February, p. 3–22, 2002. Citado na página 12.

7 MELLO, C. A. B. de; LINS, R. D. A comparative study on commercial OCR toolsNo Title. In: *Vision Interface'99*. Québec, Canadá: [s.n.], 1999. p. 224–232. Citado na página 13.

8 LINS, R. D. Two Decades of Document Processing in Latin America. *Journal of Universal Computer Science*, v. 17, n. 1, p. 151–161, 2011. Citado 5 vezes nas páginas 14, 15, 18, 26 e 42.

9 OTSU, N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 9, n. 1, p. 62–66, 1979. Citado 5 vezes nas páginas 13, 19, 52, 53 e 55.

10 GATOS, B.; NTIROGIANNIS, K.; PRATIKAKIS, I. ICDAR 2009 Document Image Binarization Contest (DIBCO 2009).pdf. n. Dibco, 2009. Citado 2 vezes nas páginas 16 e 21.

11 PRATIKAKIS, I. et al. ICFHR 2016 handwritten document image binarization contest (H-DIBCO 2016). *Proceedings of International Conference on Frontiers in Handwriting Recognition, ICFHR*, p. 619–623, 2017. Citado 4 vezes nas páginas 16, 18, 21 e 48.

12 STATHIS, P.; PAPAMARKOS, N. An Evaluation Technique for Binarization Algorithms. *Journal of Universal Computer Science*, v. 14, n. 18, p. 3011–3030, 2008. Citado 3 vezes nas páginas 16, 50 e 54.

13 NTIROGIANNIS, K.; GATOS, B.; PRATIKAKIS, I. Performance Evaluation Methodology for Historical Document Image Binarization. *IEEE Transactions on Image Processing*, v. 22, n. 2, p. 595–609, feb 2013. Citado na página 16.

14 H-DIBCO 2016 Handwritten Document Image Binarization Contest. 2016. Disponível em: <http://vc.ee.duth.gr/h-dibco2016/>. Acesso em: 21 dez. 2016. Citado 3 vezes nas páginas 17, 18 e 22.

15 KAPUR, J.; SAHOO, P.; WONG, A. A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing*, v. 29, n. 1, p. 140, jan 1985. Citado 5 vezes nas páginas 16, 52, 53, 54 e 56.

16 YEN, J. C.; CHANG, F. J.; CHANG, S. A New Criterion for Automatic Multilevel Thresholding. *IEEE Transactions on Image Processing*, v. 4, n. 3, p. 370–378, 1995. Citado 4 vezes nas páginas 16, 52, 53 e 56.

17 LU, W.; SONGDE, M.; LU, H. An effective entropic thresholding for ultrasonic images. *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, p. 1552–1554, vol. 2, 1998. Citado 5 vezes nas páginas 16, 47, 52, 53 e 56.

18 TIAN, D. Z.; WANG, C.; ZHANG, Z. M. Dynamic threshold algorithm for removal of Back-to-Front noises of visual document image. *Proceedings - International Conference on Machine Learning and Cybernetics*, v. 4, p. 1752–1755, 2011. Citado na página 16.

19 SILVA, J. M. M. da et al. A New and Efficient Algorithm to Binarize Document Images Removing Back-to-Front Interference. *Journal of Universal Computer Science*, v. 14, n. 2, p. 299–313, 2008. Citado na página 16.

20 SILVA, J. M. M. da; LINS, R. D.; ROCHA, V. C. da. Binarizing and filtering historical documents with back-to-front interference. In: *Proceedings of the 2006 ACM symposium on Applied computing - SAC '06*. [S.l.: s.n.], 2006. p. 853–858. Citado 3 vezes nas páginas 16, 53 e 56.

21 LINS, R. D. et al. Assessing Binarization Techniques for Document Images. In: *Proceedings of the 2017 ACM Symposium on Document Engineering - DocEng '17*. [S.l.: s.n.], 2017. p. 183–192. Citado 4 vezes nas páginas 16, 47, 71 e 79.

22 CHAKI, N.; SHAIKH, S. H.; SAEED, K. A Comprehensive Survey on Image Binarization Techniques. *Studies in Computational Intelligence*, v. 560, p. 5–16, 2014. Citado 2 vezes nas páginas 18 e 55.

23 SEZGIN, M.; SANKUR, B. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, v. 13, n. 1, p. 146, jan 2004. Citado 2 vezes nas páginas 18 e 52.

24 LINS, R. D. et al. Efficiently Generating Digital Libraries of Proceedings with The LiveMemory Platform. In: *International Telecommunications Symposium*. [S.l.: s.n.], 2010. Citado 2 vezes nas páginas 19 e 26.

25 PRATIKAKIS, I.; GATOS, B.; NTIROGIANNIS, K. ICDAR 2013 Document Image Binarization Contest (DIBCO 2013). In: *2013 12th International Conference on Document Analysis and Recognition*. [S.l.]: IEEE, 2013. p. 1471–1476. Citado na página 21.

26 PRATIKAKIS, I.; GATOS, B.; NTIROGIANNIS, K. ICDAR 2011 Document Image Binarization Contest (DIBCO 2011). *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, n. Dibco, p. 1506–1510, 2011. Citado na página 21.

27 PRATIKAKIS, I.; GATOS, B.; NTIROGIANNIS, K. H-DIBCO 2010 - Handwritten Document Image Binarization Competition. *2010 12th International Conference on Frontiers in Handwriting Recognition*, IEEE, p. 727–732, nov 2010. Citado na página 21.

28 JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: a review. *ACM Computing Surveys*, v. 31, n. 3, p. 264–323, sep 1999. Citado 4 vezes nas páginas 26, 28, 29 e 30.

29 RAJARAMAN, A.; ULLMAN, J. D. Clustering. In: *Mining of Massive Datasets*. Cambridge: Cambridge University Press, 2014. cap. 7, p. 213–251. Citado 5 vezes nas páginas 26, 27, 28, 32 e 36.

30 TAN, P.; STEINBACH, M.; KUMAR, V. Data mining cluster analysis: basic concepts and algorithms. In: *Introduction to Data Mining*. [S.l.]: Addison-Wesley, 2013. cap. 8, p. 487–568. Citado 2 vezes nas páginas 26 e 28.

31 LLOYD, S. P. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, v. 28, n. 2, p. 129–137, 1982. Citado 2 vezes nas páginas 29 e 32.

32 WU, G. et al. An Improved K-means Algorithm for Document Clustering. In: *2015 International Conference on Computer Science and Mechanical Automation (CSMA)*. [S.l.]: IEEE, 2015. p. 65–69. Citado na página 29.

33 DEBORAH, L. J.; BASKARAN, R.; KANNAN, A. A Survey on Internal Validity Measure for Cluster Validation. *International Journal of Computer Science & Engineering Survey*, v. 1, n. 2, p. 85–102, nov 2010. Citado na página 30.

34 FRANK, E.; HALL, M. A.; WITTEN, I. H. The WEKA Workbench. In: *Data Mining: Practical Machine Learning Tools and Techniques*. Fourth edi. [S.l.]: Morgan Kaufmann, 2016. cap. Apêndice O. Citado na página 30.

35 SOKAL, R. R.; MICHENER, C. D. A statistical method for evaluating systematic relationship. *University of Kansas science bulletin*, v. 28, p. 1409–1438, 1958. Citado na página 32.

36 WARD, J. H. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, v. 58, n. 301, p. 236–244, mar 1963. Citado 2 vezes nas páginas 32 e 34.

37 ARTHUR, D.; VASSILVITSKII, S. K-Means++: the Advantages of Careful Seeding. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, v. 8, p. 1027–1025, 2007. Citado na página 36.

38 EFROS, A. A.; FREEMAN, W. T. Image quilting for texture synthesis and transfer. *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01*, p. 341–346, 2001. Citado na página 41.

39 LINS, R. D. et al. Assessing Strategies to Remove Back-to-Front Interference in Color Documents Assessing Strategies to Remove Back-to-Front Interference in Color Documents. n. August, 2015. Citado na página 43.

40 PENG, X. et al. Markov random field based binarization for hand-held devices captured document images. *Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing*, n. i, p. 71–76, 2010. Citado na página 47.

41 LINS, R. D.; ÁVILA, B. T.; de Araújo Formiga, A. BigBatch – An Environment for Processing Monochromatic Documents. In: CAMPILHO, A.; KAMEL, M. (Ed.). *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*. Berlin, Heidelberg:

Springer Berlin Heidelberg, 2006, (Lecture Notes in Computer Science, October). p. 886–896. Citado 2 vezes nas páginas 50 e 52.

42 Weszka J. S.; Rosenfeld, A. Histogram Modification for Threshold Selection. *IEEE Transaction on Systems, Man and Cybernetics*, SMC-9, n. 1, p. 38–52, 1979. Citado na página 52.

43 DAWOUD, A.; KAMEL, M. S. Iterative multimodel subimage binarization for handwritten character segmentation. *IEEE Transactions on Image Processing*, v. 13, n. 9, p. 1223–1230, 2004. Citado na página 52.

44 LIU, Y.; SRIHARI, S. N. Document Image Binarization Based on Texture Features. *IEEE Transaction on Pattern Analysis and Machine Inteligence*, v. 19, n. 5, p. 540–544, 1997. Citado na página 52.

45 VALIZADEH, M.; KABIR, E. Binarization of degraded document image based on feature space partitioning and classification. *IJDAR*, v. 15, p. 57–69, 2012. Citado na página 52.

46 PUN, T. Entropic thresholding, a new approach. *Computer Graphics and Image Processing*, v. 16, n. 3, p. 210–239, 1981. Citado 3 vezes nas páginas 52, 53 e 56.

47 Johannsen, G and Bille, J. A threshold selection method using information measures. In: *Int'l Conf. Pattern Recognition*. [S.l.: s.n.], 1982. p. 140–143. Citado 3 vezes nas páginas 52, 53 e 54.

48 MELLO, C. A. B.; LINS, R. D. Generation of Images of Historical Documents by Composition. In: *Proceedings of the 2002 ACM symposium on Document engineering - DocEng '02*. [S.l.: s.n.], 2002. p. 127. Citado na página 52.

49 ALMEIDA, M. M. et al. A New Binarization Algorithm for Historical Documents. *Journal of Imaging*, n. Special Issue, p. 1–12, 2018. Citado 3 vezes nas páginas 53, 71 e 79.

50 HUANG, L. K.; WANG, M. J. J. Image thresholding by minimizing the measures of fuzziness. *Pattern Recognition*, v. 28, n. 1, p. 41–51, 1995. Citado 2 vezes nas páginas 53 e 54.

51 PREWITT, J. M. S.; MENDELSOHN, M. L. THE ANALYSIS OF CELL IMAGES. *Annals of the New York Academy of Sciences*, v. 128, n. 3, p. 1035–1053, dec 2006. Citado 3 vezes nas páginas 53, 54 e 55.

52 VELASCO, F. R. *Thresholding Using the Isodata Clustering Algorithm*. [S.l.], 1979. 14 p. Citado 2 vezes nas páginas 53 e 54.

53 LI, C.; TAM, P. An iterative algorithm for minimum cross entropy thresholding. *Pattern Recognition Letters*, v. 19, n. 8, p. 771–776, 1998. Citado 2 vezes nas páginas 53 e 55.

54 GLASBEY, C. An Analysis of Histogram-Based Thresholding Algorithms. *Graphical Models and Image Processing*, v. 55, n. 6, p. 532–537, nov 1993. Citado 2 vezes nas páginas 53 e 55.

55 MELLO, C. A. B.; LINS, R. D. Image segmentation of historical documents. *Visual 2000*, 2000. Citado 2 vezes nas páginas 53 e 55.

56 KITTLER, J.; ILLINGWORTH, J. Minimum error thresholding. *Pattern Recognition*, v. 19, n. 1, p. 41–47, jan 1986. Citado na página 53.

57 ImageJ. 2003. Disponível em: <https://imagej.nih.gov/ij/index.html>. Acesso em: 21 dez. 2017. Citado 2 vezes nas páginas 53 e 55.

58 TSAI, W.-H. Moment-preserving thresolding: A new approach. *Computer Vision, Graphics, and Image Processing*, v. 29, n. 3, p. 377–393, 1985. Citado 2 vezes nas páginas 53 e 55.

59 DOYLE, W. Operations Useful for Similarity-Invariant Pattern Recognition. *Journal of the ACM*, v. 9, n. 2, p. 259–267, apr 1962. Citado na página 53.

60 SAHOO, P.; WILKINS, C.; YEAGER, J. Threshold selection using Renyi's entropy. *Pattern Recognition*, v. 30, n. 1, p. 71–84, 1997. Citado 2 vezes nas páginas 53 e 56.

61 SHANBHAG, A. G. Utilization of Information Measure as a Means of Image Thresholding. *CVGIP: Graphical Models and Image Processing*, v. 56, n. 5, p. 414–419, 1994. Citado na página 53.

62 ZACK, G. W.; ROGERS, W. E.; LATT, S. A. Automatic measurement of sister chromatid exchange frequency. *The Journal of Histochemistry and Cytochemistry*, v. 25, n. 7, p. 741–753, 1977. Citado na página 53.

63 TOMASI, C.; MANDUCHI, R. Bilateral filtering for gray and color images. *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, p. 839–846, 1998. Citado na página 53.

64 VERKUILEN, J. Assigning membership in a fuzzy set analysis. *Sociological Methods and Research*, v. 33, n. 4, p. 462–496, 2005. Citado na página 54.

65 LI, C.; LEE, C. Minimum cross entropy thresholding. *Pattern Recognition*, v. 26, n. 4, p. 617–625, apr 1993. Citado na página 55.

66 LINS, R. D. et al. Binarizing Document Images Acquired with Portable Cameras. In: *14th IAPR International Conference on Document Analysis and Recognition*. [S.l.]: IEEE, 2017. p. 45–50. Citado 2 vezes nas páginas 71 e 79.

APÊNDICE A – TRABALHOS PUBLICADOS

Os seguintes trabalhos estão associados a esta dissertação:

1. LINS, R. D. et al. Assessing Binarization Techniques for Document Images. Em: *Proccedings of the 2017 ACM Symposium on Documento Engineering* – DocEng '17. 2017. p. 183-192. Referência [21]

2. LINS, R. D. et al. Binarizing Document Images Acquired with Portable Cameras. Em: 14th IAPR International Conference on Document Analysis and Recognition. IEEE, 2017. p. 45–50. Referência [66]

3. ALMEIDA, M. M. et al. A New Binarization Algorithm for Historical Documents. *Journal of Imaging*, Special Issue, p. 1–12, 2018. Referência [49]

Assessing Binarization Techniques for Document Images

Rafael Dueire Lins
UFPE/UFRPE, Recife, PE
Brazil
rdl.ufpe@gmail.com

Marcos Martins de Almeida
UFPE, Recife, PE
Brazil
mm.ufpe@gmail.com

Rodrigo Barros Bernardino
UFPE, Recife, PE
Brazil
rbbernardino@gmail.com

Darlisson Jesus
UFPE, Recife, PE
Brazil
dmj.ufpe@gmail.com

José Mário Oliveira
IFPE/UFPE, Recife, PE
Brazil
josealexandre@recife.ifpe.edu.br

DocEng '17, September 04-07, 2017, Valletta, Malta

ACM ISBN 978-1-4503-4689-4/17/09…$15.00
http://dx.doi.org/10.1145/3103010.3103021

ABSTRACT

Image binarization is a technique widely used for documents as monochromatic documents claim for far less space for storage and computer bandwidth for network transmission than their color or even grayscale equivalent. Paper color, texture, aging, translucidity, kind and color of ink used in handwritting, printing process, digitalization process, etc., are some of the factors that affect binarization. No algorithm is good enough to be a winner in the binarization of all kinds of documents. This paper presents a methodology to assess the performance of binarization algorithms for a wide variety of text documents, allowing a judicious quantitative choice of the best algorithms and their parameters.

CCS CONCEPTS

• **Applied computing** → **Computers in other domains** → **Publishing**

KEYWORDS

Documents, binarization, back-to-front interference, bleeding, show through, image filtering, big-data.

ACM Reference format:

R.D.Lins, M.M. de Almeida, R.B. Bernardino, D. Jesus, J.M. Oliveira. 2017. Assessing Binarization Techniques for Document Images. In *Proceedings of ACM Symposium on Document Engineering, Valetta, Malta, September 2017, (DocEng' 17)*, 10 pages.
DOI: 10.1145/3103010.3103021

1 INTRODUCTION

Document image binarization is an important step in the document image analysis and recognition pipeline. Monochromatic documents claim for far less storage space and computer bandwidth for network transmission than color or grayscale documents. It is imperative to have a benchmarking dataset along with an objective evaluation methodology to capture the efficiency of current document image binarization algorithms.

The international competitions on binarization algorithms are an evidence of the relevance of this area. The most traditional of such competitions is possibly DIBCO - Document Image Binarization Competition, which was first organized at the ICDAR-International Conference on Document Analysis and Recognition in 2009 and has been repeated yearly ever since. The methodology used by DIBCO is to offer a small set of "real-world" images and their "ground-truth" binary equivalent that were "hand-generated" or "hand-retouched". Figure 1 presents the complete test set of the ten images used at DIBCO 2016, which may be obtained at http://vc.ee.duth.gr/h-dibco2016/benchmark/. As one may observe in Figure 1, the DIBCO test set is formed only by handwritten documents both in grayscale and color. Some documents present stains (1, 3, 4, 10) and aging marks (4, 9, 10). DIBCO provides an evaluation tool that yields as output the F-Measure, pseudo F-Measure, PSNR, DRD,

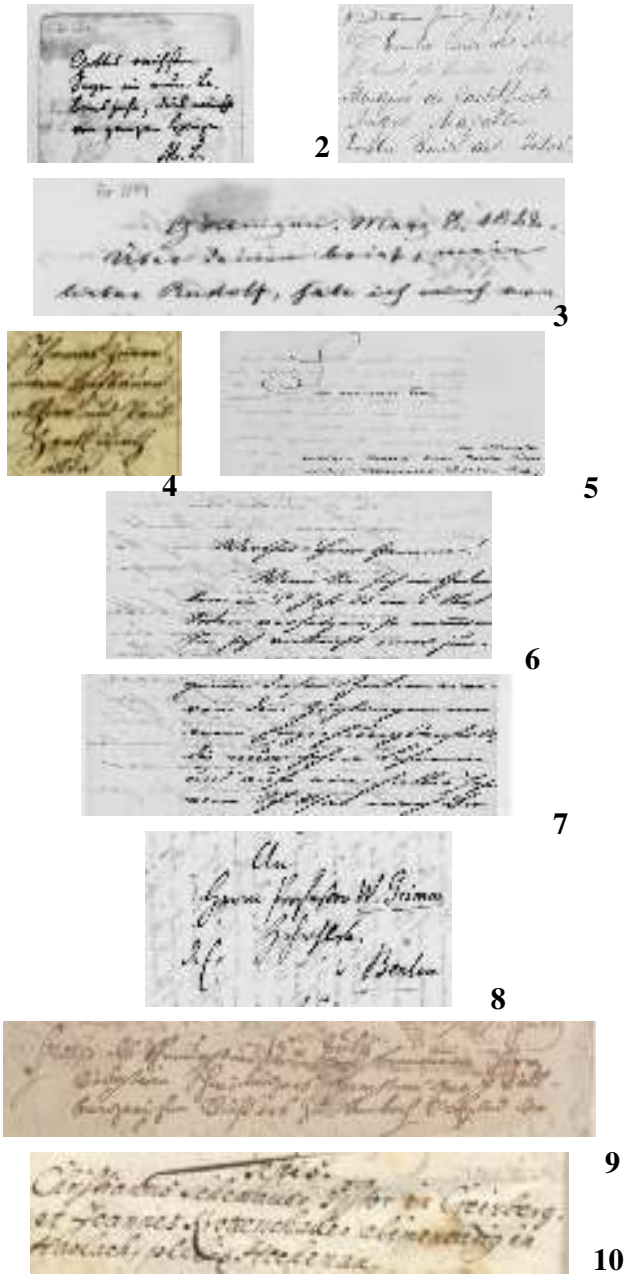


Figure 1: DIBCO 2016 Test images

Recall, Precision, pseudo-Recall and pseudo-Precision. Some of those measures are not usual and are explained in reference [1]. DIBCO 2017 intends to include images of typed or printed documents in its dataset, which has not been released so far (https://vc.ee.duth.gr/dibco2017/ last visited on 04th July, 2017).

As one may observe, all document images in DIBCO 2016 test set, but the first one, have the back-to-front interference, that is, whenever a document is typed or written on both sides of a sheet of paper and the opacity of the paper is such as to allow the back printing or writing to be visualized on the front side. Such image overlap phenomenon was first addressed in the literature by Lins in 1994 [2], who called it back-to-front interference. Much later, other researchers called it bleeding or show-through [3]. The human brain is able to filter out that sort of noise keeping document readability. This is not the case with automatic tools such as OCRs. The direct application of some binarization algorithms such as the one in Jasc Paint Shop Pro TM version 8 (Palette component: Gray values, Reduction component: nearest color, Palette weight: non-weighted), as many other commercial tools, yield a completely unreadable document, as the interfering ink of the backside of the paper overlaps with the binary one in the foreground. Several algorithms were developed specifically to binarize documents with back-to-front interference [7][10][11][13], but depending on the strength of the interference present, which accounts on the opacity of the paper, its permeability, the kind and degree of fluidity of the ink used, the degree of difficulty for obtaining a good segmentation capable of filtering-out such a noise increases enormously, as new set of hues of paper and printing colors appear.

Document image binarization is extremely challenging and there is no chance of a specific algorithm to be an all case winner as many parameters may interfere in the quality of the resulting image. Besides that, a small set of test images will never be able to provide a real quality assessment of binarization algorithms. It is important to be able to have a very large test set of synthetic images representative of the universe of text documents and to know for each of them which algorithms and with which parameters, minimum space and processing time one is able to get the best binarization result. Artificial intelligence and big-data strategies now provide the resources to given a "real-world" document image to be able to decide which kind of document it better matches in such a large database. Known the best-match between the "real-world" document and the synthetic one, the set of suitable binarization algorithms and their parameters becomes known.

This paper explains the methodology used in the generation of such a large controlled database for synthetic images. A quantitative measure of quality is introduced. Some evidence of the effectiveness of the method proposed is also provided.

2 GENERATING SYNTHETIC IMAGES

Historical documents with back-to-front interference are certainly the most difficult kind of document to binarize, as paper aging introduce non-uniform textures whose color distribution may overlap with the distribution of the colors from the writing in the back of the paper. Figure 2 presents the block diagram for the generation of synthetic images.

Two images of documents of different nature (typed, handwritten with different pens, printed, etc.) are taken: F – front and V – verso (back). The verso image is offset by 10, 20 and 30 pixels to make the back image not to coincide with the front one. Then, the offset verso image is "blurred" by passing though Gaussian filters that simulate the low-pass effect of the

translucidity of the verso as seen in the front part of the paper. The "blurred" verso image is now faded with a coefficient α varying between 0 and 1 in steps of 0.1. The two images are overlapped by performing a "darker" operation [20] pixel-by-pixel in the images. Paper texture is added to the image to simulate the effect of document aging. The steps in the generation of the synthetic images are explained next. It is important to remark that the two major concerns here: the first one is to have ground-truth images to be able to assess the performance of the several different binarization algorithms, the second one is to be able to have a very large set of synthetic images that will be used to train a classifier that will be able to automatically match a "real-world" image with the synthetic one.

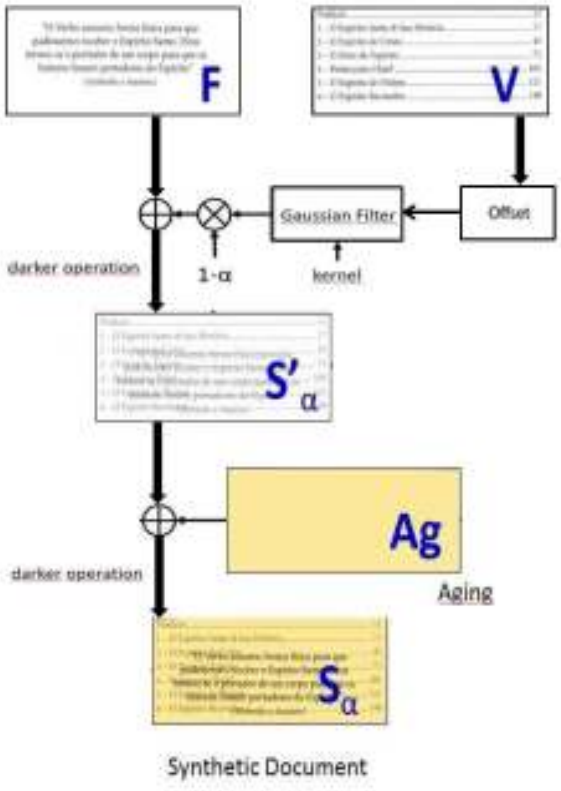


Figure 2: Block diagram of the scheme for the generation of synthetic images

2.1 The Ground-truth images

The first step of the generation of synthetic images was to produce a set of images that covers all the universe of text documents: typed in mechanical typewriters, printed in inkjet, laser, offset in most usual colors (black, blue, red), handwritten with different kinds of pen (fountain, ballpen, felt pen) from different manufacturers, using black and blue ink. Such documents were typed/printed/written in good quality A4 white papers. Such images were scanned using a flatbed scanner set to a resolution of 300 dpi in true-color (24 bits RGB) yielding raster images standardized in 2,480 × 3,508 pixels. The images obtained were binarized using the standard binarization algorithm in Jasc Paint Shop Pro version 8 and are used as ground truth images and also in the generation of the synthetic images. Salt and pepper noise is removed. Such images correspond to 43 handwritten and 88 printed documents.

The set of ground truth documents of the whole DIBCO series, 61 handwritten and 25 typewritten documents, were also used here. Besides those, 14 documents electronically generated pdf documents are also used as ground-truth. Thus, currently, 231 document images compose the set of ground-truth images in total.

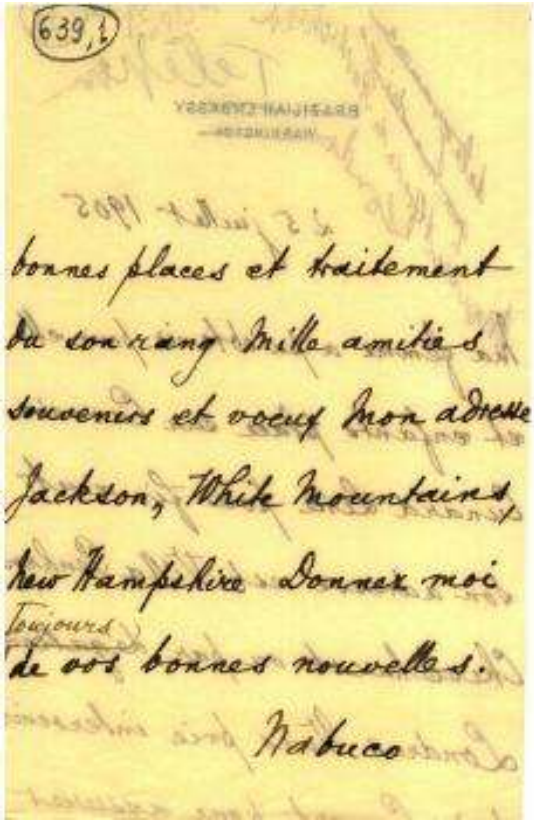


Figure 3 – Letter from Joaquim Nabuco

2.2 The Back-to-front blur

As already mentioned, documents with the back-to-front interference are much harder to binarize. Depending on the thickness of the paper, its texture,

permeability, age, storage conditions (temperature, humidity, direct exposure to sun light, etc.), kind of ink, printing process or pen in case of handwritten documents, etc., the back ink is seen more or less blurred in the front side of the paper. Such effect has been modeled now as being performed by a Gaussian filter.

Two "light" Gaussian filters 3x3 and 5x5 pixel-

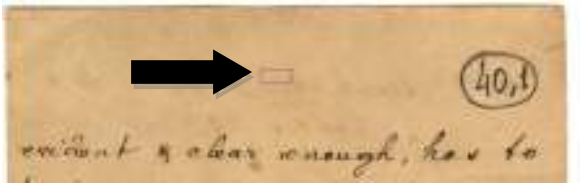


Figure 4 – Sample of the texture of paper background.

kernels were used at the current stage of the generation of the database of synthetic images, presenting "similar" effect as the one in real documents under visual inspection. Current work is being developed to better model this effect in the different kinds of documents. For that, several samples of small windows are being used collecting parts from the foreground and back-to-front interference. The foreground window will be blurred using Gaussian filters having their parameters modified to match the one of the interference. Performing such approximation in several different kinds of documents one will be able to obtain the parameters of the different low-pass filters that better model the bleeding effect, or the back-to-front blur.

2.3 Image Fading

The origin of this project dates back to the early 1990's when the first author of this paper [4] undertook the mission of digitalizing the bequest of historic documents of Joaquim Nabuco, a Brazilian statesman, writer and diplomat, leader in the freedom of black slaves in Brazil. His active correspondence is of paramount importance for understanding the history of the Americas in the late 19th century. That bequest of about 6,500 documents encompassed over 18,000 pages. Those documents were risking of degradation due to problems in the extreme acidity of the paper. A careful analysis of the preservation staff of the Joaquim Nabuco Foundation, the social science research institute in Recife, Brazil, that keeps most of Nabuco's documents, selected about 300 documents as representative of the universe of documents. At that time, for storage restrictions and transmission of documents via FAX-simile devices, binarization was mandatory. That was exactly the first time that the back-to-front interference was reported in the technical literature [2], because about 200 of those documents were written on both sides of translucent paper, with a great variability of strength. Figure 3 presents an example of one of those letters from Nabuco bequest.

The "strength" of the back-to-front interference is modeled by the fading coefficient α. One hundred different levels of fading coefficients were chosen, thus $0<\alpha<1$ in steps of 0.01.

2.4 Adding paper texture

The texture of the paper has a strong influence the performance of binarization algorithms. Thus, it is of paramount importance to get a set of paper textures that are representative of the universe of documents intended to be modeled, from late 19th century to today, which will be used in the assessment of binarization algorithms. To do so 3,351 document images were used, of which 1,048 were from Nabuco bequest and the other 2,303 were obtained from five years of the LiveMemory Project, which generated a digital library of all the proceedings the SBrT - Brazilian Telecommunications Symposium. The images were automatically scanned looking for a window of 20x50 pixels such as the purple one shown in Figure 4.

The automatic window selection was human checked to guarantee that the area has no ink or other sort of noises. For each texture sample a vector of features was built taking into account each RGB-channel of the sample, the image average filtered (R+G+B)/3, and its grayscale equivalent. For each of those 5 images the following 7 statistic measures were taken and placed in a vector: mean, standard deviation, mode, minimum value, maximum value, median, and kurtosis.

The 3,351 vectors were statistically analyzed using the hierarchical clustering method implemented in the scikit-learn library [22]. It uses a bottom up approach, where each observation starts in its own cluster, and clusters are successively merged together, providing 84 cluster distributions of paper texture as shown in Figure 5. The texture in the centroid of each of such clusters was taken as being representative of the whole cluster. The visual inspection made in the 84 clusters showed acceptable texture variation within each cluster.

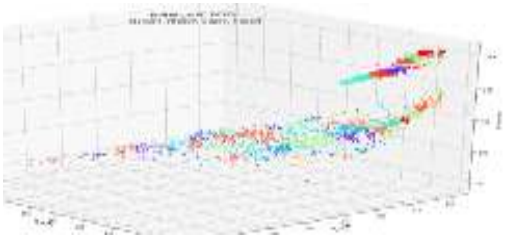


Figure 5 – Distribution of 3,351 paper textures in 84 representative clusters.

Besides those 84 centroid cluster-representative textures, 16 "isolated" textures that were left-out of the clusters were added to the texture set, totaling 100 different textures. Each of those textures is used for generating a "blank" sheet of paper to be used to colorize the synthetic image providing the "aging" effect in the scheme presented in Figure 2. For that, a RGB-image with 2,480

× 3,508 pixels (equivalent to an A4 blank sheet of paper with 300 dpi resolution) is generated. A similar technique is used to generate a 300 dpi texture for the smaller DIBCO ground-truth images. Two different texture generation strategies were adopted. In the first one, the color of each pixel is randomly chosen from the 10,000 pixels in another 100x100 pixels sample of the texture at the center of the texture cluster, providing a 300 dpi image with the same distribution as the original sample. The second technique employs image quilting [17]. Figure 6 presents an example of a texture generated using both techniques, in which the latter more closely resemble the texture of the sample document.

Each image is than added with a "darker" operation [20] generating the set of Sα synthetic images, which will be used to assess the binarization algorithms. Reference [5] proposes a parametric scheme for image compression and generation in which the paper texture is generated through a Gaussian distribution centered on the mean value of the color of the pixels. Both schemes presented here allows more "natural looking" textures that can be efficiently indexed.

The current version of the test set of synthetic images encompasses a total 2,777,000 color images (231 groud-truth x 2 blur x 100 α-fading-coefficients x 3 offsets x 100 textures-patterns x 2 texture generation schemes) and the same number of grayscale equivalent. It is probable that the analysis of the binarization of this set of 5,554,000 images will provide a better assessment of the binarization capability of algorithms than the set of only 10 images in DIBCO 2016.

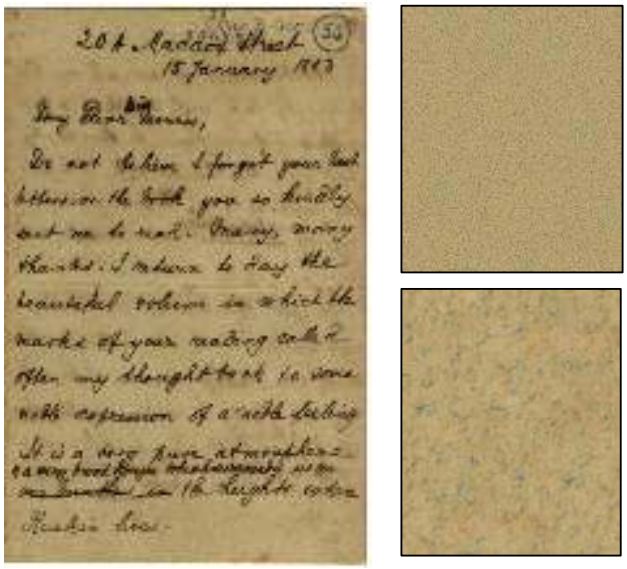


Figure 6 – (Left) Historic document. (Top-right) Texture: random distribution. (Bottom-right) Texture: image quilting.

3 ASSESSING ALGORITHMS

The enormous variety of kinds of text documents makes extremely improbable that one single algorithm is able to satisfactorily binarize all kinds of documents. Most probably, depending on the nature (or degree of complexity) of the image several or no algorithm will be able to provide good results. If binarization is part of an OCR transcription platform, the higher the correct transcription rate the better the algorithm is. It is important to remark that, according to the experiments made by the authors of this paper, OCR transcription and "visual inspection" assessment methods do not provide similar results, even in printed or typed documents. The assessment method proposed here is to provide accurate information about the binarized documents generated by the different algorithms, and the user will choose the most suitable one depending on the target application.

The assessment methodology proposed here is "image centered" instead of the traditional "algorithm centered" approach. This means that the question to be answered here is "Which are the best algorithms and their parameters to binarize image X?" instead of the traditional one "Which is the best algorithm?". Such a new approach does not provide an answer, but a set of answers. Obviously, humans are not able to handle and analyze such a large set of data, which has to be made "user-friendly" in an automated platform, currently under development by the authors.

Binarization algorithms, in general, make use of different criteria to find a threshold that splits the mapping of pixels onto white or black. Thresholding algorithms can be classified into global or local algorithms. Global algorithms define a unique threshold value for the complete image. Local algorithms first split the image into regions according to some criterion and then define threshold values for each region. In general, global algorithms are faster than local algorithms. Although local algorithms potentially provide better results as their parameters are better tuned for each small window, the kind of "tiling" effect of the small blocks tend not to yield acceptable quality results. The assessment methodology presented here works equally well with global and local binarization algorithms.

Sezgin and Sankur [6] presented a comprehensive overview and comparison of the "classical" binarization algorithms, clustering them according to their nature. From the almost forty algorithms presented there, six schemes were chosen to illustrate: Kapur-Sahoo-Wang [7], Otsu [8], Johannsen-Bille [9], Yen-Chang-Chang [10], Wu-Lu [11], and Pun [19] algorithm.

The binarization using the IsoData - Iterative Self Organizing Data Analysis Technique [18] was also tested. It is a method of unsupervised classification, and the computer runs the algorithm through several iterations until the threshold is reached.

Four algorithms specifically developed to filter-out the back-to-front interference were also assessed: Mello-Lins [13], Silva-Lins-Rocha [11], Roe-Mello [7], and Almeida-Lins-Lima [15].

The basic criterion for the choice of the algorithms assessed here was code availability. To illustrate the assessment methodology proposed here, one synthetic document was chosen with 0.1≤α≤ in steps of 0.1. Samples of some of those documents are presented in Figure 7.

Figure 7 – Synthetic images with 0.6<α<1.

The tables below present: P(b|b) - the percentage of background pixels correctly mapped onto white pixels of the ground-truth image, P(f|f) – the percentage of foreground pixels correctly mapped onto black pixels of the ground-truth image, P(f|b) and P(b|f) are the percentage of mismatches. The column "Threshold" presents the value of the threshold automatically chosen by the algorithm.

The tables below present: P(b|b) - the percentage of background pixels correctly mapped onto white pixels of the ground-truth image, P(f|f) – the percentage of foreground pixels correctly mapped onto black pixels of the ground-truth image, P(f|b) and P(b|f) are the percentage of mismatches. The column "Threshold" presents the value of the threshold automatically chosen by the algorithm.

3.1 The Kapur-Sahoo-Wong Filter

The algorithm by Kapur et al. [7] considers the foreground and background images as two distinct sources, such that whenever the addition of the two entropies reach a maximum, its argument t reaches the optimal value.

Table 1: Kapur-Sahoo-Wong

α	Threshold	P(b|b)%	P(b|f)%	P(f|f)%	P(f|b)%
0.1	176	90.88	9.12	100.00	0.00
0.2	174	91.50	8.50	100.00	0.00
0.3	174	91.86	8.15	100.00	0.00
0.4	174	92.29	7.71	100.00	0.00
0.5	173	92.98	7.02	100.00	0.00
0.6	174	93.49	6.51	100.00	0.00
0.7	147	99.25	0.75	100.00	0.00
0.8	162	98.87	1.13	100.00	0.00
0.9	175	98.59	1.41	100.00	0.00
1.0	182	98.36	1.64	100.00	0.00

The analysis of the data in Table 1 reveals that there was the partial elimination of the back-to-front interference, for $0.7 \leq \alpha \leq 1.0$ as the value of background-background probability P(b|b) varied between 99.25% and 98.36%, an error less than 1.64%, considering that the foreground-foreground matching percentage P(b|b) was of 100.00%. Table 1 clearly shows that this algorithm reaches the best performance for the image with α=0.7, with a P(b|f) of 0.75%.

3.2 Otsu threshold method

Otsu [8] is the most widely used global thresholding algorithm. Otsu's algorithm is adaptive and requires no adjustment setting. It considers that there are two classes, separated by a threshold value. Otsu's algorithm makes use of Sahoo discriminator analysis for defining whether a gray level t is mapped onto foreground or background information. The result of this algorithm applied to the synthetic images with different alphas is shown in Table 2.

Although Otsu algorithm was originally developed for ultrasound images, the results above show that it performs well with document images. Table 2 shows that for $0.7 \leq \alpha \leq 1.0$, the value of background-background correct mapping percentage was $99.87\% \leq P(b|b) \leq 99.95\%$ yielding error less than 0.13%, while the foreground-foreground percentage $99.54\% \leq P(f|f) \leq 99.56\%$, an error less than 0.47%. Comparing the data presenting in Table 1 and 2 one may conclude that Otsu presented better results than Kapur-Sahoo-Wong filter for that specific set of images.

Table 2: Otsu Filter

α	Threshold	P(b|b)%	P(b|f)%	P(f|f)%	P(f|b)%
0.1	145	94.19	5.81	100.00	0.00
0.2	145	94.57	5.43	100.00	0.00
0.3	145	95.05	4.95	100.00	0.00
0.4	149	95.24	4.76	100.00	0.00
0.5	149	96.00	4.00	100.00	0.00
0.6	146	97.51	2.49	100.00	0.00
0.7	138	99.87	0.13	99.54	0.46
0.8	138	99.94	0.06	99.56	0.44
0.9	138	99.97	0.03	99.53	0.47
1.0	140	99.95	0.05	99.55	0.45

3.3 Johannsen-Bille

This method [9] uses the entropy of the gray level histogram of the digital image. Essentially, it divides the set of gray into two parts, to minimize the interdependence between them. Table 3 presents the performance obtained by this filter for the test set. The results shown demonstrate that the Johanssen-Bille filter is very unstable depending on the opacity coefficient α, as when its values were 0.3, 0.6, 0.7, and 0.8 the output was completely black images. The Johanssen-Bille algorithm presented in some of the cases (α=0.5, 0.9, 1.0) an information loss, as over 10% of the foreground pixels were mapped onto background ones.

Table 3: Johanssen-Bille

α	Threshold	P(b|b)%	P(b|f)%	P(f|f)%	P(f|b)%
0.1	142	94.49	5.51	99.52	0.48
0.2	149	94.23	5.77	100.00	0.00
0.3	210	0.00	100.00	100.00	0.00
0.4	150	95.15	4.85	100.00	0.00
0.5	100	99.97	0.03	84.63	15.37
0.6	211	0.00	100.00	100.00	0.00
0.7	211	0.00	100.00	100.00	0.00
0.8	211	0.00	100.00	100.00	0.00
0.9	112	100.00	0.00	88.39	11.61
1.0	112	100.00	0.00	88.11	11.89

3.4 Yen-Chang-Chang

The binarization algorithm by Yen-Chang-Chang [10] follows the same ideas as the one by Kapur et al. [7] in respect to the entropy distributions. The result of applying Yen-Chang-Chang Method to the test set of document images is showed in Table 4.

Table 4: Yen-Chang-Chang

α	Threshold	P(b|b)%	P(b|f)%	P(f|f)%	P(f|b)%
0.1	210	0.00	100.00	100.00	0.00
0.2	210	0.00	100.00	100.00	0.00
0.3	210	0.00	100.00	100.00	0.00
0.4	210	0.00	100.00	100.00	0.00
0.5	178	92.14	7.86	100.00	0.00
0.6	211	0.00	100.00	100.00	0.00
0.7	211	0.00	100.00	100.00	0.00
0.8	211	0.00	100.00	100.00	0.00
0.9	176	98.47	1.53	100.00	0.00
1.0	183	98.23	1.77	100.00	0.00

The results presented in Table 4 show that Yen-Chang-Chang algorithm is not suitable to binarize the test set images as seven out of ten images were mapped onto completely black images.

3.5 The Wu-Lu algorithm

The Wu-Lu binarization algorithm [11] was also originally developed for ultrasound images and seems to work particularly well in images with few contrast values. It is based on Shannon entropy and uses the lower difference between the minimum entropy of the objects and the entropy of the background as threshold value. Table 5 presents the results obtained in using Wu-Lu algorithm in the binarization of the test set images.

Analyzing the results presented in Table 5, one may see that, although the value of the percentage of background-background mapping P(b|b) did not vary much and is either 100.00% or very close to that value for all the α´s, the P(f|f) value of foreground-foreground mapping varied between 36.61% and 59.72%, registering an error up to 63.39%, a strong loss of information in the text. That indicates that the Wu-Lu algorithm is possibly not suitable to binarize such set of document images.

Table 5: Wu-Lu

α	Threshold	P(b|b)%	P(b|f)%	P(f|f)%	P(f|b)%
0.1	75	99.13	0.87	62.81	37.19
0.2	75	99.00	1.00	62.45	37.55
0.3	74	99.96	0.04	61.00	39.00
0.4	73	100.00	0.00	59.72	40.28
0.5	72	100.00	0.00	57.70	42.30
0.6	71	100.00	0.00	55.86	44.14
0.7	70	100.00	0.00	54.23	45.77
0.8	68	100.00	0.00	50.21	49.79
0.9	66	100.00	0.00	45.99	54.01
1.0	62	100.00	0.00	36.61	63.39

3.6 Pun Algorithm

The algorithm proposed by Pun [19] takes as input a gray level image considered as produced by a source with an alphabet consisting of 256 statistically independent symbols. Pun considers the ratio between the *a posteriori* entropy and the total entropy as the image threshold. Table 6 presents the results of applying Pun's algorithm to the gray-level version of the synthetic images in the test set.

Table 6: Pun

α	Threshold	P(b|b)%	P(b|f)%	P(f|f)%	P(f|b)%
0.1	195	61.99	38.01	100.00	0.00
0.2	196	57.97	42.03	100.00	0.00
0.3	196	59.15	40.85	100.00	0.00
0.4	196	61.64	38.36	100.00	0.00
0.5	196	65.20	34.80	100.00	0.00
0.6	196	67.16	32.84	100.00	0.00
0.7	198	55.51	44.49	100.00	0.00
0.8	198	58.39	41.61	100.00	0.00
0.9	198	60.52	39.48	100.00	0.00
1.0	199	60.52	39.48	100.00	0.00

Pun algorithm is not suitable for the binarization of the test set of images although the P(f|f) was of 100.00% for all α's, the P(b|b) was around 60%, reaching 55.51 % for α = 0.7, meaning that are large number of background pixels were mapped onto black pixels of the monochromatic image.

3.7 The IsoData Method

Clustering is an unsupervised classification as no a priori knowledge (such as samples of known classes) is assumed to be available. The ISODATA Algorithm (Iterative Self-Organizing Data Analysis Technique Algorithm) [18] allows the number of clusters to be adjusted automatically during the iteration by merging similar clusters and splitting clusters with large standard deviations. The algorithm is highly heuristic. In the case of using the IsoData algorithm for binarizing document images the pixels in the image are iteratively sent to two clusters which will correspond to the black and white pixels. Table 7 presents the result of the binarization of the test set images using the IsoData algorithm.

Table 7: IsoData Clustering

α	Threshold	P(b|b)%	P(b|f)%	P(f|f)%	P(f|b)%
0.1	142	94.49	5.51	99.52	0.48
0.2	142	94.84	5.16	99.53	0.47
0.3	144	95.14	4.86	100.00	0.00
0.4	146	95.54	4.46	100.00	0.00
0.5	147	96.22	3.78	100.00	0.00
0.6	144	97.85	2.15	100.00	0.00
0.7	136	99.89	0.11	98.87	1.13
0.8	137	99.94	0.06	99.23	0.77
0.9	137	99.98	0.02	99.20	0.80
1.0	138	100.00	0.00	99.56	0.44

Analyzing the quality of the binarized images produced by the Isodata filter, it seems reasonable to consider important features for removing back-to-front interference: where the interference fade varied between $0.7 \leq \alpha \leq 1.0$, the value of the background-background mapping yielded an error of less than 0.11% as $99.89\% < P(b|b) < 100.00\%$. The foreground to foreground matching percentage P(f|f) had a small variation between 99.56% and 98.87%, a error less than 1.13%. It is interesting to notice that for very weak back-to-front interference (α=0.1, α=0.2) over 5% of the pixels from the paper texture were mapped onto the foreground, degrading the quality of the image. The filtering threshold varied between 136 and 147.

3.8 Mello-Lins Algorithm

The algorithm by Mello and Lins [12] is based on Shannon entropy to calculate a global threshold. It was developed with the aim of filtering out the back-to-front interference. The results obtained for the images in the test set are presented in Table 8.

Table 8: Mello-Lins

α	Threshold	P(b|b)%	P(b|f)%	P(f|f)%	P(f|b)%
0.1	174	91.19	8.81	100.00	0.00
0.2	183	89.76	10.24	100.00	0.00
0.3	181	90.58	9.42	100.00	0.00
0.4	180	91.21	8.78	100.00	0.00
0.5	178	92.14	7.86	100.00	0.00
0.6	176	93.14	6.86	100.00	0.00
0.7	174	94.47	5.53	100.00	0.00
0.8	170	97.30	2.70	100.00	0.00
0.9	165	99.19	0.81	100.00	0.00
1.0	181	98.45	1.55	100.00	0.00

All the pixels of the foreground in the test images were correctly mapped onto pixels of the foreground in the ground case images, as P(f|f)=100% for all values of α. The P(b|b) values were very high, reaching its best performance for α=0.9.

3.9 Silva-Lins-Rocha algorithm

The algorithm developed by Silva-Lins-Rocha [13] was developed to further improve the Mello-Lins algorithm. It considers the histogram distribution as the 256-symbol source (a priori source) distribution. It is assumed the hypothesis that all the symbols are statistically independent. In the case of real images one knows that this hypothesis does not hold. However, according to [13], this largely simplifies the algorithm and was supposed to yield better results than its predecessors.

The result of applying Silva-Lins-Rocha algorithm to the test images provided the results presented in Table 9.

As one may observe, considering the test set used, the Silva-Lins-Rocha actually performed better than the Mello-Lins algorithm for all values of fading coefficient but α=0.9, for some reason.

Table 9: Silva-Lins-Rocha

α	Threshold	P(b|b)%	P(b|f)%	P(f|f)%	P(f|b)%
0.1	89	97.60	2.40	78.73	21.27
0.2	95	97.77	2.23	82.80	17.20
0.3	105	97.94	2.06	86.73	13.27
0.4	115	98.17	1.83	90.60	9.40
0.5	126	98.44	1.56	94.96	5.04
0.6	137	98.80	1.20	99.22	0.74
0.7	150	98.80	1.20	100.00	0.00
0.8	161	98.98	1.02	100.00	0.00
0.9	167	99.07	0.93	100.00	0.00
1.0	165	99.26	0.74	100.00	0.00

3.10 Roe-Mello

The Roe-Mello [14] algorithm performs a local image equalization based on color constancy, and an extension to the standard difference of Gaussian edge detection operator, XDoG and Otsu binarization algorithm. The last two algorithms assessed are based on the entropy of the image, whereas the Roe-Mello one uses discriminator analysis. The threshold used by the algorithm showed very little variation, as may be observed in Table 10.

Table 10: Roe-Mello

α	Threshold	P(b|b)%	P(b|f)%	P(f|f)%	P(f|b)%
0.1	181	88.16	11.84	39.39	60.61
0.2	181	88.41	11.59	39.10	60.90
0.3	181	88.73	11.27	39.11	61.89
0.4	180	89.23	10.76	36.45	63.55
0.5	181	94.84	5.16	23.70	76.30
0.6	181	95.41	4.59	22.46	77.54
0.7	181	95.55	4.45	22.10	77.90
0.8	181	95.63	4.37	22.04	77.96
0.9	181	95.63	4.37	22.03	77.97
1.0	181	98.58	4.42	22.13	77.87

The results obtained by the Roe-Mello algorithm may be considered unsuitable for the binarization of the test set used.

3.11 The Almeida-Lins-Lima algorithm

The algorithm recently proposed by Almeida, Lins and Lima [15] is performed in four steps: filtering the image using a bilateral filter [16], splitting image into the RGB components, decision-making for each RGB channel based on an adaptive binarization method inspired by Otsu's method with a choice of the threshold level, and classification of the binarized images to decide which of the RGB components best preserved the document information in the foreground. It is far more computation intensive than its predecessors and involves training for the Decision-making block. Testing this algorithm with the same set of test images the automatically chosen threshold is equal to 126 and the channel that is chosen for providing the best results in binarizing the images is the Red channel. The results obtained are summarized in Table 11.

Table 11: Almeida-Lins-Lima

α	Threshold	P(b|b)%	P(b|f)%	P(f|f)%	P(f|b)%
0.1	126	96.49	3.51	100.00	0.00
0.2	126	96.93	3.07	100.00	0.00
0.3	126	97.66	2.34	100.00	0.00
0.4	126	99.60	0.40	100.00	0.00
0.5	126	99.87	0.13	100.00	0.00
0.6	126	99.91	0.09	100.00	0.00
0.7	126	99.94	0.06	100.00	0.00
0.8	126	99.97	0.03	100.00	0.00
0.9	126	99.99	0.01	100.00	0.00
1.0	126	100.00	0.00	100.00	0.00

The results presented for this algorithm show that for all the images in the chosen test set this algorithm performed better that its predecessors, exhibiting a steady “behavior” with the variation of the fading coefficient α. It is important to remark that this and the IsoData algorithms claim far more computational resources than the other algorithms assessed.

4 GLOBAL RESULTS

The assessment presented in the last section for the ten selected binarization algorithms presented for one test set formed by ten synthetic images obtained with ten different fading coefficients α varying from 0.1 to 1.0 in steps of 0.1 showed that the performance of the algorithms is highly dependent of the features of the document image. Further testing was made with a larger set of 1,600 synthetic images with the coefficient α varying between 0 and 1 in steps of 0.01. The average of the results of P(b|b)% and P(f|f)% were taken for each of the filters assessed for each value of α. The filters that showed both P(b|b)% and

P(f|f)% average values higher than 99% and are presented in Table 12. The data presented in Table 12 corroborate the hypothesis formulated that the performance of binarization algorithms depends heavily on the "intrinsic nature" of the document image, and that a small variation in the image may yield completely different performance figures. In that sense, the data presented in this section must be read as a simple indicator of the quality of the images generated by those algorithms using a controlled test set, not being adequate to read the results as a quality classification rank for the compared algorithms.

Table 12: Overall algorithm classification for 1,600 synthetic images with 0<α<1 in steps of 0.1.

α	*P(b|b)%*	*P(f|f)%*	*Filter*	*Threshold*
1.0	100.00	100.00	Almeida-Lins-Lima	126
1.0	100.00	99.56	IsoData	138
1.0	99.95	99.56	Otsu	140
0.9	99.99	100.00	Almeida-Lins-Lima	126
0.9	99.98	99.20	IsoData	137
0.9	99.97	99.53	Otsu	138
0.9	99.07	100.00	Silva-Lins	167
0.8	99.97	100.00	Almeida-Lins-Lima	126
0.8	99.94	99.23	IsoData	137
0.8	99.94	99.56	Otsu	138
0.8	98.98	100.00	Silva-Lins	161
0.7	99.94	100.00	Almeida-Lins-Lima	126
0.7	99.25	100.00	Kapur SW	147
0.7	99.87	99.54	Otsu	138
0.6	99.91	100.00	Almeida-Lins-Lima	126
0.5	99.87	100.00	Almeida-Lins-Lima	126
0.4	99.60	100.00	Almeida-Lins-Lima	126

5 CONCLUSIONS

No binarization algorithm is an "all-kind-of-document" winner. Several factors such as paper texture, aging, thickness, tranlucidity, permability, the kind of ink, its fluidity, color, aging, etc., all may influence the performance of each algorithm. This paper presents an assessment methodology based on the controlled generation of a large set of synthetic images that allows identifying quality aspects of the binarized images.

Eleven different binarization algorithms presented in this paper were used to binarize the images in the test set database of 1,478,400 binary images that were compared with the 134,400 ground truth images, allowing to know for each of them the percentage and type of matching (P(b|b)% and P(f|f)%) and mismatched (P(b|f)% and P(f|b)%) pixels.

The authors plan to develop an image "matcher" or "classifier" that will be trained with the database developed of synthetic images. The aim of such classifier is that, given a real-world document, the platform will automatically find the closest synthetic document to it. Once that document is found, one knows the set of binarization algorithms that are more likely to provide the best results. One important point is worth remarking here is that binarization assessments tend only to consider the quality of the resulting image "for visual inspection". In the more global assessment methodology presented here, the user will be even able to choose to prioritize to minimize either the P(b|f)% or P(f|b)%) errors, depending on the "sensitiveness" of the target application. For instance, if the resulting binary image will go through an OCR it may be better to have P(f|b)%) < P(b|f)%.

Preliminary tests made in matching the synthetic images with "real world" documents for "visual inspection" provided very good results. The image shown in Figure 8 may witness the good quality of the binary image provided by using the Almeida-Lins-Lima algorithm in the document image presented in Figure 3. The document image in Figure 9 provides another evidence of that, using the same binarization algorithm.

The assessment strategy presented here is a generalization of the platform described in reference [20]. The current version of the assessment environment encompasses 5,554,00 images (231 groud-truth x 2 blur x 100 α-fading-coefficients x 3 offsets x 100 textures-patterns x 2 texture generation schemes x 2 color/grayscale). The authors of this paper consider this image set representative of the universe or "real world" text documents. At present, twenty-five binarization algorithms are being assessed. Another relevant aspect that should be taken into account is that the proposed binarizarion platform accounts now for the time elapsed by each algorithm to binarize each image. This allows the user to choose the lightest algorithm that provides the best results. For instance, the computational cost of Otsu is extremely small if compared with the IsoData or the Almeida-Lins-Lima algoritms. At a later stage, space consumed will also be considered.

It is most relevant to emphasize the computational challenge involved in the task proposed here, as each of the synthetic images is over 10 MB large. If one attempts to store the 5,554,000 images, over 50 TB of storage would be needed, a volume of data unreasonable to be used. Each image is generated a time and then binarized in a pipeline with the 25 filters currently tested against the ground-truth image and the data is collected and stored. A slice of the image that corresponds to central one-fifth of it is being saved as a lossless PNG image to later be used in the training of the image matcher. A cluster with 10 machines is being used in this platform, using the technology described in the BigBatch project [21]. Priority was given to four different values of alpha (α=1 no interference, α=0.8 weak interference, α=0.6 medium interference, α=0.4 strong back-to-front interference). The partial assessment results will be made publically available as they are obtained. The authors would like to remark that even processing in a dedicated cluster with ten nodes, several

months of processing are needed. The preliminary version of the DIB-Document Image Binarization platform and website is publically available at www.cin.ufpe.br/~dib.

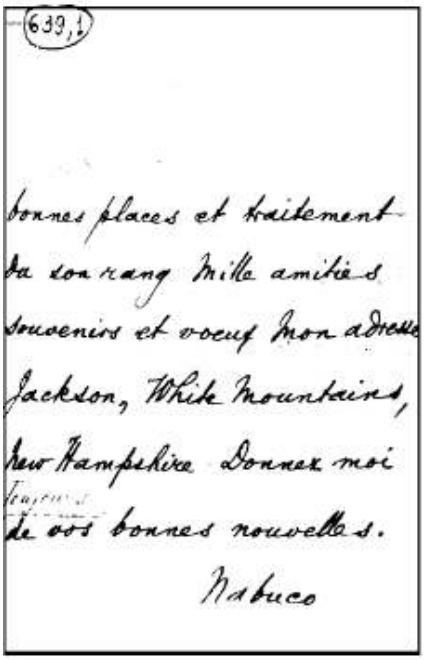


Figure 8 –Binarized version of the document shown in Figure 3 using the Almeida-Lins-Lima algorithm.

ACKNOWLEDGMENTS

The authors of this paper are grateful for those who made the code of their algorithms publically available for testing and performance analysis and to CNPq – Brazilian Government for sponsoring this research.

REFERENCES

[1] K. Ntirogiannis, B. Gatos and I. Pratikakis, Performance Evaluation Methodology for Historical Document Image Binarization, IEEE Trans. Image Proc., vol.22, no.2, pp. 595-609, Feb. 2013..

[2] R. D. Lins et al. An Environment for Processing Images of Historical Documents. Microproc. and Microprogramming, 111–121, 1995.

[3] G. Sharma. Show-trough cancellation in scans of duplex printed documents. IEEE Transaction Image Processing, v. 10, n. 5, p. 736–754, 2001.

[4] R. D. Lins. Nabuco – Two Decades of Processing Historical Documents in Latin America. Journal of Universal Computer Science. , March 2011.

[5] C. A. B. Mello and R. D. Lins. 2002. Generation of Images of Historical Documents by Composition. Symposium on Document Engineering, 127–133. 2002.

[6] M.Sezgin and B.Sankur. A Survey over Image Thresholding Techniques and Quantitative Performance Evaluation. Journal of Electronc Imaging, v. 1, n. 13, p. 146–165, 2004.

[7] J. N. Kapur, P. K. Sahoo, A. K. C. Wong. A New Method for Gray-Level Picture Thersholding Using the Entropy of the Histogram. C. Vision Graphics and Image Processing, v. 29, p. 273–285, 1985.

[8] N. Otsu. A Threshold Selection Method from Gray-Level Histograms. IEEE Transaction on Systems, Man and Cybernetics, v. SMC-9, n. 1, p. 62–66, 1979.

[9] G. Johannsen and J. A. Bille. A Threshold Selection Method Using Information Measure. ICPR'82 - Proceeding 6th International Conference on Pattern Recognition, 140–143. 1982.

[10] J. C. Yen, F. J. Chang, S. Chang. 1995. A New Criterion for Automatic Multilevel Thresholding. IEEE Transaction Image Process IP-4, 370–378.

[11] U. L. Wu, A. Songde, L. U. Haqing. 1998. An Effective Entropic Thresholding for Ultrasonic Imaging. International Conference Pattern Recognition, 1522–1524.

[12] C. A. B. Mello and R. D. Lins. Generation of Images of Historical Documents by Composition. Proceedings of the 2002 ACM symposium on Document engineering, 127–133, 2002.

[13] J. M. M. Silva, R. D. Lins, V. C. Rocha. Binarizing and Filtering Historical Documents with Back-to-Front Interference. ACM Symposium on Applied Computing, 853–858, 2006.

[14] E. Roe and C. A. B. Mello. Binarization of Color Historical Document Images Using Local Image Equalization and XDoG. 12th International Conference on Document Analysis and Recognition, August, p. 205–209, 2013.

[15] M. A. M. de Almeida, R. D. Lins, B. C. Lima, A New Binarization Algorithm for Images with Back-to-Front Interference. Submitted for publication, 2017.

[16] S. Paris, P. Kornprobst, J. Tumblin and F. Durand. Bilateral Filtering: Theory and Applications. Foundations and Trends in Computer Graphics and Vision. Vol. 4, No. 1, 1–73. 2008.

[17] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. SIGGRAPH '01 28th annual conference on Computer graphics and interactive techniques, 341-346. 2001.

[18] [N. Memarsadeghi, D. M. Mount, N. S. Netanyahu, J. Moigne. 2007. A Fast Implementation of the IsoData Clustering Algorithm. International Journal of Computational Geometry and Applications, 71–103.

[19] T. Pun. Entropic Thresholding, A New Approach. Computer Vision Graphics and Image Processing, 210–239, 1981.

[20] R. D. Lins and G. F. P e Silva. Assessing Strategies to Remove Back-to-Front Interference in Color Documents. IEEE International Telecommunications Symposium, 2010, IEEE Press, p. 1-6, 2010.

[21] G. G.Mattos, A. A. Formiga, R. D. Lins, F. M. J. Martins. BigBatch: a document processing platform for clusters and grids. ACM-SAC 2008. ACM Press, 2008. v. I. p. 434-441.

[22] Scikit-learn. http://scikit-learn.org/stable/ (visited: 31st May 2017)

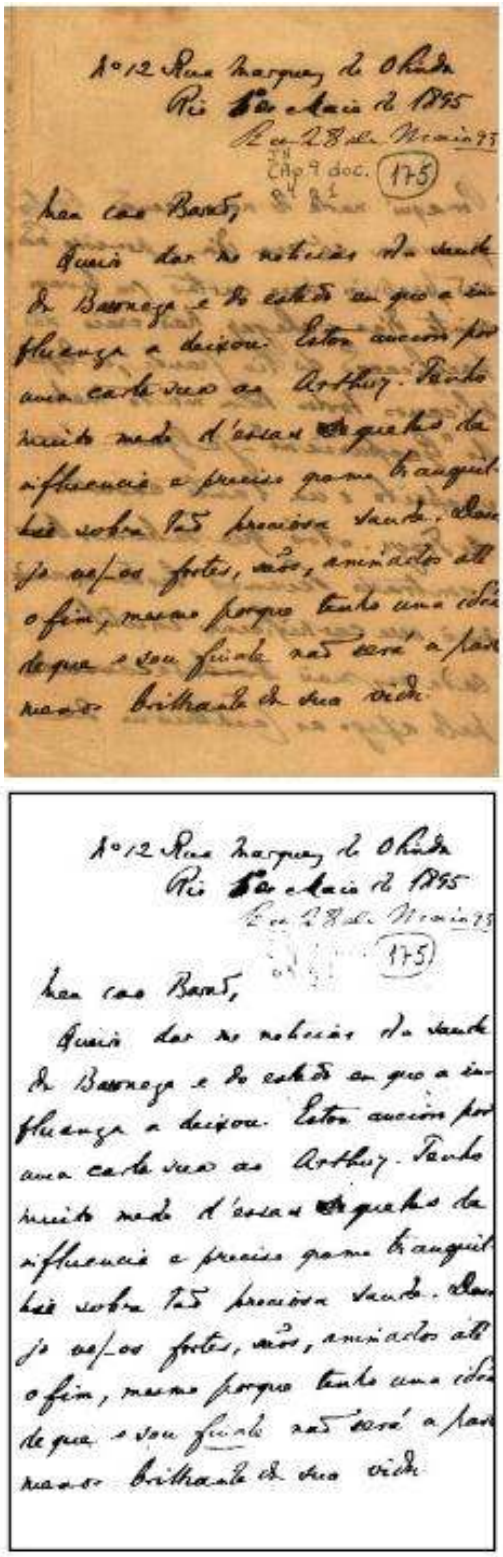


Figure 9 –Historic document and its binary version produced by Almeida-Lins-Lima algorithm.

Binarizing Document Images Acquired with Portable Cameras

Rafael Dueire Lins
UFRPE – UFPE
Recife, PE, BRAZIL
rdl.ufpe@gmail.com

Rodrigo Barros Bernardino,
Darlisson Jesus
DES – CTG – UFPE
Recife, PE, BRAZIL
{rbbernardino, dmj.ufpe}@gmail.com

José Mário Oliveira
IFPE – UFPE
Recife, PE, BRAZIL
josealexandre@recife.ifpe.edu.br

***Abstract* — Although made for "family photos" portable digital cameras, either in standalone models or embedded in cell phones, are often used to take photos of documents today. In general, such photos are sent via networks and either visualized in desktops, printed, or even transcribed via OCR. Binarization may play an important role in such a scheme. This paper follows the idea that "no binarization algorithm is good for all kinds of images". Non-uniform illumination, the possible interference of light sources from the environment, and non-uniform resolution are some of the problems found in photographed document images that are not present in their scanned counterparts. This paper presents a new methodology to assess binarization algorithms in different devices, taking into account the difficulties listed and the particularities of the cameras and documents.**

Keywords — portable digital cameras, binarization, image processing.

I. INTRODUCTION

Not long ago, document digitalization using cameras was restricted to very fragile, odd sized, bound, and illumination-sensitive historical documents; this is no longer true. Those cameras were high-cost professional models that were used in mechanical supports to avoid perspective distortion and were used in light-controlled environments to avoid interferences. For almost a decade, portable digital cameras have become a pervasive good, omnipresent in the life of most people. Portable models, including the ones embedded in cell phones, have already reached resolutions of 41 Mpixels, such as in the case of the cell phone Nokia Lumia 1020, a model released in July/2013, providing very good quality images. Professionals and students from different areas now use cell phones as a fast a practical way to acquire images of documents, taking advantage of its availability, low weight, portability, low cost, small size, etc.

Several "camera scanning" softwares are available both for Apple (iOS) and Android devices. Evernote Scannable is considered by [2] the best "foolproof" app for Apple devices as: *"All you need to do is open the app and point your camera at what you want to scan. The app does the rest of the work by searching the camera's field of view for a sheet of paper, automatically focusing the shot and taking a photo."* The app automatically saves any scans directly to the Evernote account of the user in the iCloud storage or manually export to the Photos app. In the case of Android devices, according to [2], the free app CamScanner is considered the best one as it: *"Once you've captured an image, there are tons of ways to edit it by adjusting the color, contrast or brightness, or by cropping the image."* The "document scan" apps minimize image skew, perspective distortion, and borders from the document surroundings. Despite all those facilities of such apps, they do not perform document binarization satisfactorily. Most users transfer the document photos to a desktop computer for later reading in a larger screen or printing. In all cases, being able to generate good quality binary images of documents is a key factor to store, transfer via networks and economically print such documents. But, the binarization of camera acquired document images is far from being a trivial task. Having a non-uniform illumination that may suffer the interference from light sources of the environment and also having a non-uniform image resolution are the two most important complicating factors, assuming that the document image has little perspective and skew distortion, the document had its borders removed, and that lens distortion is not perceivable.

This article focuses on the binarization of camera acquired text documents, the kind of document that is most often photographed by professionals and students. The background idea is that "no binarization algorithm is good for all kinds of images" [3]. This paper presents an assessment methodology to quantitatively evaluate the performance of binarization algorithms for photographed documents, taking as reference scanned test images. The assessment performed was made with images acquired with three different models of cell phones. The methodology presented here would allow knowing which algorithm or algorithms would be most suitable for a specific family of devices and could run either embedded in the phone or in a desktop computer in an environment such as PhotoDoc [4]. The resolution of the camera of the tested devices was around 10 Mpixels the one found in most models of cell phones today. The test image used here were stored in the jpeg format, with default setting (1% loss). Although some models of cell-phones, including the Nokia Lumia 1020 [1] and one of the devices used for tests here, allow saving photographed images in the raw format, the adoption of the jpeg images correspond to the most generally used format for saving photos today. Twenty-one binarization algorithms are assessed here. They were chosen for having the code available, being widely known and used.

II. Assessment Methodology

The degree of freedom brought to the users with portable digital cameras is proportional to the complexity of the algorithms to handle the acquired images. Features such as non-uniformity of colors, illumination, and depth bring a "natural" look to the photos. In the case of photographed text documents, however, one wants exactly the opposite: plain colors with minimum hue variation throughout the document image, as provided by scanners. The particularities of camera documents make impossible to have a ground-truth image to serve as reference, due to the variable resolution of the acquired image (caused by the non-fixed camera-document distance) and even non-uniform resolution (due to the lens-distortion).

The binarization of scanned documents is still a complex research topic. Much research efforts have been driven to assessing the performance of binarization algorithms. The DIB platform (http://dib.cin.ufpe.br/) [3] and the DIBCO (https://vc.ee.duth.gr/dibco2017/) [5] contest witness the complexity and relevance of such a problem. The complexity of assessing binarization algorithms for documents obtained with cameras is much higher as too many factors may influence the performance. Those factors may be grouped into three clusters. The first one is device related, such as camera resolution, position and intensity of the embedded flash, lens distortion, minimum focal distance, image stabilization, etc. The second group of factors is related to the image acquisition process *per se*, such as: the existence of interfering light sources, minimizing skew and perspective of the acquired photo, image focus, embedded flash activation, etc. The third set of factors depends on the features of the binarization algorithm: if local or global, threshold calculation method, if applied directly to the color image of if to the grayscale equivalent, etc.

The first author of this paper was possibly the first researcher to propose using OCRs as a way to assess the quality of photographed documents by measuring the Levenshtein distance [6] between the text in the original document and the OCR transcription made of the photographed document [4]. Although that may be considered as a valid method also for assessing the quality of binarization algorithms, if the purpose of binarization is automatic document transcription or indexing, that may not provide valid results in terms of human readability. Besides that, the interpretation of Levenshtein distance [6] is not straightforward as is states the number of character insertion, deletion, and replacements needed to generate the reference text taking as input the text transcribed from the binarized image.

In this paper, the assessment of binarization images is made comparing the difference between the ratio between the black to white pixels in the binary image generated by the algorithm and the same ratio of the ground truth scanned one. The whole idea of the assessment made here is that the closer the proportion between the number of the black and white pixels in the binary document and the one in the ground-truth image, the better the quality of the image of the binary document, as the closer to the original ratio between black and white pixels it is. This hypothesis was validated by the visual inspection of the resulting monochromatic documents.

Thus, given that one takes a photo of a printed or handwritten document, having as background a white sheet of non-glossy paper, opaque enough as to no back-to-front interference be observed [7], this paper establishes a methodology to aim to answer the following questions:

1. Given a camera, which is possibly the best algorithm to binarize the image of texts documents?
2. Should the embedded flash be on or off?

It is assumed here that all documents prior to binarization were cropped to remove its noisy border, leaving only the document page. The borders were automatically removed using the algorithm reported in reference [8]

III. The Tested Algorithms

In general, document image binarization is extremely challenging and, as described in reference [3], even in the much "well-behaved" case of scanned documents there is no chance of a specific algorithm to be an all case winner as many different features of real documents may interfere in the quality of the resulting image. As made clear above, the scope of the present assessment is to answer the two questions formulated. The assessment methodology may allow a criterious choice of binarization algorithm to be embedded into a device-specific document processing app. Twenty-one binarization algorithms were tested using the methodology described:

1. DaSilva-Lins-Rocha [9]
2. Intermodes [10]
3. IsoData-MODF [11]
4. IsoData-ORIG [11]
5. Johannsen-Bille [12]
6. Kapur-Sahoo-Wong [13]
7. Li-Tam [14]
8. Mean [15]
9. Mello-Lins [16]
10. MinError [17]
11. Minimum (variation of [10])
12. Mixture-Modeling [18]
13. Moments [19]
14. Otsu [20]
15. Percentile[21]
16. Pun [22]
17. RenyEntropy (variation of [13])
18. Shanbhag [23]
19. Triangle [24]
20. Wu-Lu [25]
21. Yean-Chang-Chang [26]

The basic criterion for the choice of the algorithms assessed was code availability.

IV. RESULTS

Three different models of cell phones were used in the tests reported here. The tests were performed with equipments considered "popular" among users:

- LG K4 – 5 Mpixel camera, f/2.8 aperture size, Flash "LED".
- IPhone 6 - 8 Mpixel camera, 1/3-inch sensor size, F2.2 aperture size, Flash "Dual LED".
- IPhone SE - 12 Mpixel camera, 1/3-inch sensor size, F2.2 aperture size, Flash "Dual LED".

The total number of documents used was 118 documents, of which 76 were printed and 42 handwritten. The documents in the test set were scanned in 300 d.p.i. resolution, true color. The scanned documents in the test set were binarized using the different algorithms tested. The images binarized using Otsu algorithm were chosen as ground-truth images, as the visual inspection of the documents provided good quality monochromatic images.

For each of the camera models above, the set of documents was digitized with and without the activation of the embedded flash. In both cases, there was no control of interfering external light sources, to simulate the real day-by-day operation performed by users in general.

A. *LG K4- 5 Mpixels*

The results obtained by the ten best binarization algorithms for the cell phone manufactured by LG model K4, for the images obtained without strobe flash for the 118 documents are presented in Figure 1. The blue bar stands for the mean value of the percent of the difference between the ratios between the black and white pixels from the photographed image in relation to the ground-truth one. The vertical black line in each vertical bar stands for the deviation from the mean obtained in the analysis.

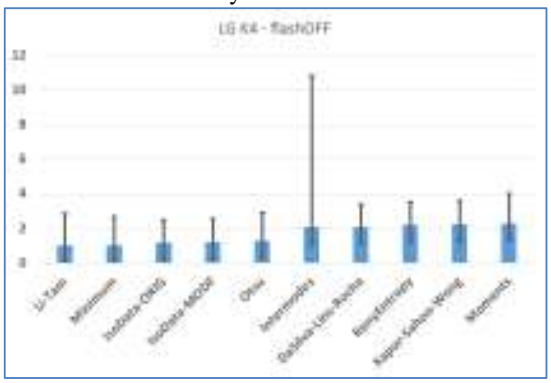


Figure 1 – LG K4 – Mean and deviation for the quality measure for the 118 test images

The analysis of the data in Figure 1 shows that the Li-Tam algorithm seems to be the most suitable algorithm for binarizing the images obtained with the LG K4 phone without flash, because it reaches the minimum mean of the error with good enough deviation interval. Figure 8 presents an example of a document image binarized with such algorithm.

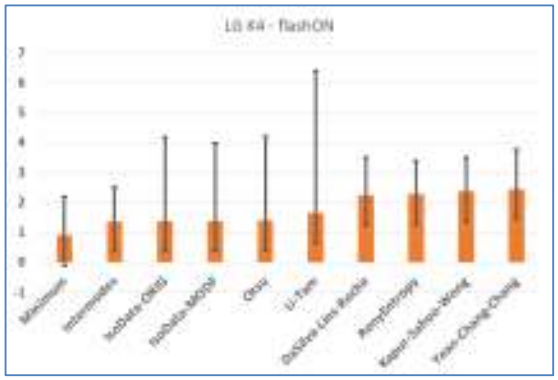


Figure 2 – LG K4 – Performance of the 10 best binarization algorithms for images with the flash on.

Figure 2 shows that the digitalization of the images acquired with the embedded strobe flash on yielded a drastic variation to the performance of the binarization algorithms. The Minimum algorithm performed best. It is also important to observe that the deviation in this algorithm reached negative values indicating that some pixels of the foreground (printed or handwritten parts) were mapped into white pixels (background). In general, one could recommend keeping the embedded strobe flash off to take photos of documents using this cell phone. An example of a document image in the set binarized with the Mininum algorithm is shown in Figure 9.

B. *IPhone 6 – 8 MPixels*

Figure 3 presents the performance results obtained for the binarization algorithms for the images acquired without flash for an IPhone 6. The first aspect to be observed is that the algorithms that presented the best results were completely different from the ones for the LG K4. The performance figures of the algorithms were much better for this device as the scale of the mean error is much smaller than in Figure 1 or 2. Although the Li-Tam algorithm provided the smallest mean error is not the best choice for using in such a device in such a setup, as it is better to have added noise to the image than loss of information. In such a case, Mello-Lins may provide better results as it maximum information loss is much smaller than the one from Li-Tam algorithm. A representative image in this set may be found in Figure 10.

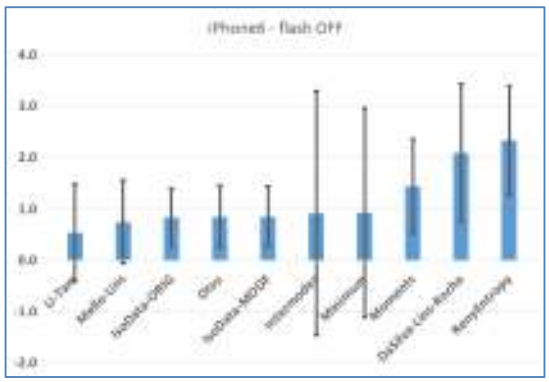


Figure 3 – The 10 best results for the images obtained with the iPhone 6 without flash.

Similarly, Figure 4 presents the results obtained for photos taken with the embedded strobe flash on.

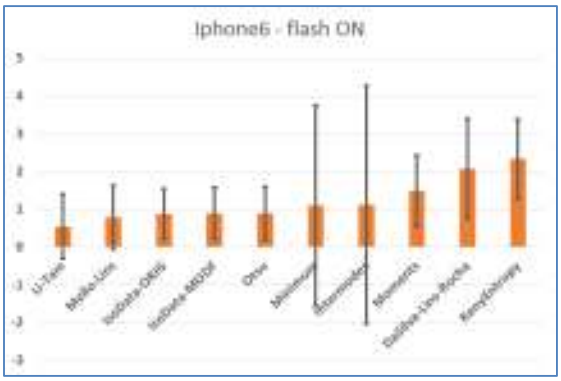


Figure 4 – The 10 best results for the 42 images obtained with the iPhone 6 – 8 Mpixels.

The results presented in Figure 4 show a different rank of the same 10 best algorithms and changes to their deviation and error. Again, the Mello-Lins algorithm is possibly the best one. In this device the flash on brought no significant variation to the error and deviation. Another example of a document image in this data set binarized with Mello-Lins is presented in Figure 11.

C. IPhone SE - 12 Mpixels

The results obtained for the 10 best binarization algorithms for the document photos taken with the iPhone SE without flash are shown in Figure 5, in which one may find a different set of algorithms as the top 10. Most possibly, one may point out Mello-Lins as the best one, as the error rate was smaller than 1% with a small deviation. No algorithm presented information loss. An example of document in this test set binarized with Mello-Lins algorithm is presented in Figure 12.

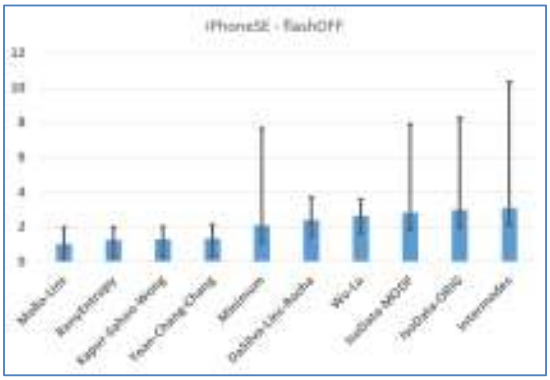


Figure 5 – Results for the iPhone SE with flash off.

Figure 6 presents the results for the 10 best binarization algorithms for the set of images taken with the flash on, in which one may observe a much higher percent of errors introduced and also a much larger error deviation interval for the best algorithms. The assessment made would not recommend having the strobe flash on, whenever digitalizing documents. Figure 13 presents an example of a document in this set binarized with RenyEntropy algorithm.

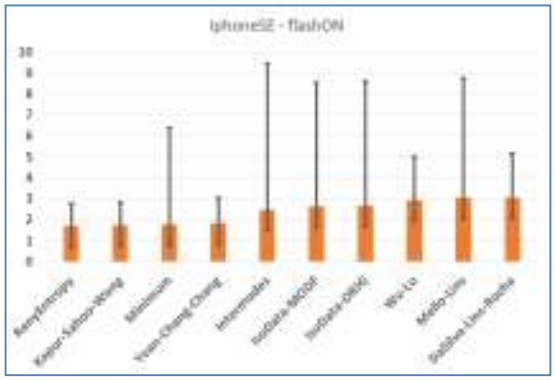


Figure 6 – Results for the iPhone SE with flash on.

D. Processing-time performance

Reference [3] remarks that the processing time of algorithms is an important feature to be analyzed. Two algorithms may produce similar quality images, but their processing times may show such a large difference that may make its use unviable. Such aspect is most relevant in the possibility of attempting to embed such algorithm in a portable device. Figure 7 presents the time elapsed to process all the data sets of the images acquired with the three devices studied here. The processing time is from a processor Intel i7-4510U @ 2.00GHz x 2, 8GB RAM, running Linux Mint 18.2 64-bit. All the algorithms were coded in Java, by their authors.

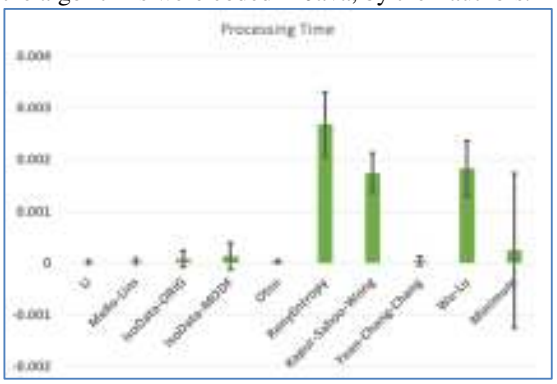


Figure 7 – Mean and deviation of the processing time elapsed per image by the 10 best binarization algorithms for the devices studied for all the document sets.

As one may observe in Figure 7, Li was the fastest among the tested binarization algorithms, being closely followed by the Mello-Lins one. The processing time of RenyEntropy, Kapur-Sahoo-Wong and Wu-Lu would hardly justify their adoption, even if the quality of the images generated were good ones, overall if the algorithm were to be app embedded.

V. CONCLUSIONS AND LINES FOR FURTHER WORK

Acquiring document images with portable cameras is widely done today. A number of factors, such as non-uniform resolution and illumination, make camera document images much harder to process than scanned ones. Document binarization saves storage space and computer bandwidth

for network transmission, thus having ways of making a judicious choice of which binarization algorithm to use is an important subject. No binarization algorithm is an "all kind of document winner", as each algorithm is intrinsically tuned to improve a number of features in the target images.

This paper presents a methodology to assess the performance of binarization algorithms for plain text camera documents, written or printed on matt white paper, the sort of document most often found today. It shows that the choice of the best binarization algorithm is not only device dependant, but also varies with the setup of the camera, particularly if the embedded strobe flash is activated. The tests performed here assessed 21 binarization algorithms and used a document set of 118 documents, both handwritten and printed, digitized using three different cell phones of two manufacturers, with models in the lower to mid price range in the market. The assessment made here also took into account the processing time of the algorithms, an important factor to analyze, if one considers the possibility of including binarization algorithms in embedded apps, a trend already observed today.

The code for the algorithms and test images are publically available at the site of the Document Image Binarization Platform at (http://dib.cin.ufpe.br/) .

ACKNOWLEDGMENTS

Research reported here was partly supported by CNPq and CAPES - Brazilian Government.

REFERENCES

[1] Nokia Lumia 1020. https://www.cnet.com/products/nokia-lumia-1020/specs/. Last visited on July, 15th 2017.

[2] The best scanning apps for Android and iPhone. https://www.cnet.com/how-to/best-scanning-apps-for-android-and-iPhone/. Last visited on July, 15th 2017.

[3] R.D.Lins, M.M. de Almeida, R.B. Bernardino, D. Jesus, J.M. Oliveira. 2017. Assessing Binarization Techniques for Document Images. In Proceedings of ACM Symposium on Document Engineering, Valetta, Malta, September 2017, (DocEng' 17), 10 pages. DOI: 10.1145/3103010.3103021

[4] G. F. P. Silva and R. D. Lins, PhotoDoc: A Toolbox for Processing Document Images Acquired Using Portable Digital Cameras, CBDAR´07, IAPR, 2007.

[5] K. Ntirogiannis, B. Gatos and I. Pratikakis, Performance Evaluation Methodology for Historical Document Image Binarization, IEEE Trans. Image Proc., vol.22, no.2, pp. 595-609, Feb. 2013.

[6] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. Soviet Phys. Dokl., pp. 707-710, February 1966.

[7] R.D.Lins, et al. An Environment for Processing Images of Historical Documents. Microprocessing & Microprogramming, pp. 111-121, North-Holland, 1995.

[8] G. F. P. Silva, R. D. Lins, and A.R.Silva. A New Algorithm for Background Removal of Document Images Acquired Using Portable Digital Cameras. ICIAR 2013, LNCS vol 7950, pp. 290-298, Springer Verlag, 2013.

[9] J. M. M. Silva, R. D. Lins, V. C. Rocha. Binarizing and Filtering Historical Documents with Back-to-Front Interference. ACM Symp. Applied Comp., 853–858, 2006.

[10] J. M. S. Prewitt and M. L. Mendelsohn, "The Analysis of Cell Images". Ann. N. Y. Acad. Sci., 128(3) :1035–1053, 1966.

[11] T.W. Ridler, S. Calvard, "Picture Thresholding Using an Iterative Selection Method" IEEE Trans. Systems, Man, and Cybernetics, vol. 8, no. 8., pp. 630–632, 1978.

[12] G. Johannsen and J. A. Bille. A Threshold Selection Method Using Information Measure. ICPR'82 - 6th International Conference on Pattern Recognition, 140–143. 1982.

[13] J. N. Kapur, P. K. Sahoo, A. K. C. Wong. A New Method for Gray-Level Picture Thersholding Using the Entropy of the Histogram. C. Vision Graphics and Image Processing, v. 29, p. 273–285, 1985.

[14] C. H. Li and P. K. S. Tam, "An iterative algorithm for minimum cross entropy thresholding" Pattern Recognition Letters, vol. 19, no. 8. Elsevier BV, pp. 771–776, Jun-1998.

[15] C. A. Glasbey, "An analysis of histogram-based thresholding algorithms" CVGIP: Graphical Models and Image Processing, vol. 55, pp. 532-537, 1993.

[16] C.A.Mello and R.D. Lins. "Image segmentation of historical documents." Visual2000, 2000.

[17] J. Kittler and J. Illingworth, "Minimum error thresholding" Pattern Recognition, vol. 19, no. 1. Elsevier BV, pp. 41–47, Jan-1986.

[18] https://imagej.nih.gov/ij/plugins/mixture-modeling.html

[19] W.-H. Tsai, "Moment-preserving thresolding: A new approach," Computer Vision, Graphics, and Image Processing, vol. 29(3). Elsevier BV, pp. 377–393, Mar-1985.

[20] N. Otsu. A Threshold Selection Method from Gray-Level Histograms. IEEE Transaction on Systems, Man and Cybernetics, v. SMC-9, n. 1, p. 62–66, 1979.

[21] W. Doyle, "Operation useful for similarity-invariant pattern recognition" Journal of the Association for Computing Machinery, vol. 9,pp. 259-267, 1962.

[22] T. Pun, "Entropic thresholding, a new approach," Comput. Graph. Image Process., vol. 16, no. 3, pp. 210–239, 1981.

[23] A. G. G. Shanbhag, "Utilization of Information Measure as a Means of Image Thresholding," CVGIP Graph. Model. Image Process., vol. 56, no. 5, pp. 414–419, 1994.

[24] G. W. Zack, W. E. Rogers, and S. A. Latt, "Automatic measurement of sister chromatid exchange frequency.," J. Histochem. Cytochem., vol. 25, no. 7, pp. 741–753, 1977.

[25] J. C. Yen, F. J. Chang, S. Chang. 1995. A New Criterion for Automatic Multilevel Thresholding. IEEE Transaction Image Process IP-4, 370–378.

[26] Lu Wu , Ma Songde , Lu Hanqing . An Effective Entropic Thresholding for Ultrasonic Imaging. International Conference Pattern Recognition, 1522–1524. 1998.

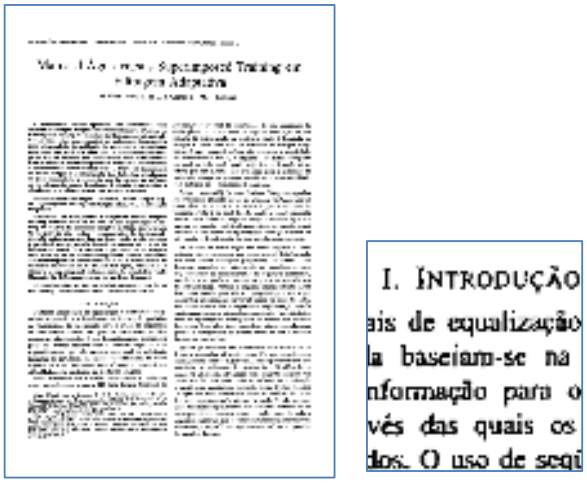


Figure 8 – Document LG K4 with flash off, binarized using Li-Tam [14] algorithm, part in zoom.

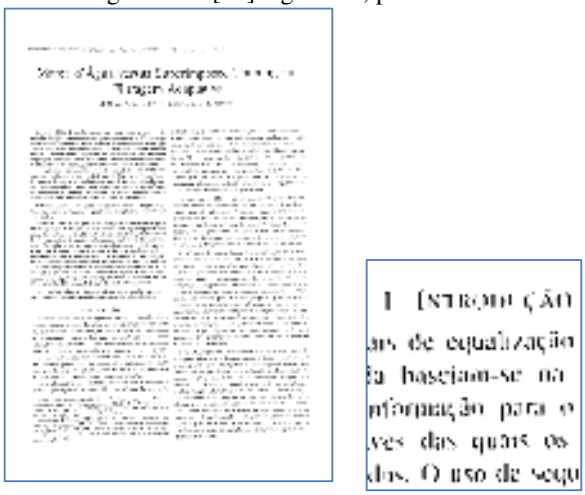


Figure 9 – LG K4 document image flash on, binarized using Minimum [11] algorithm.

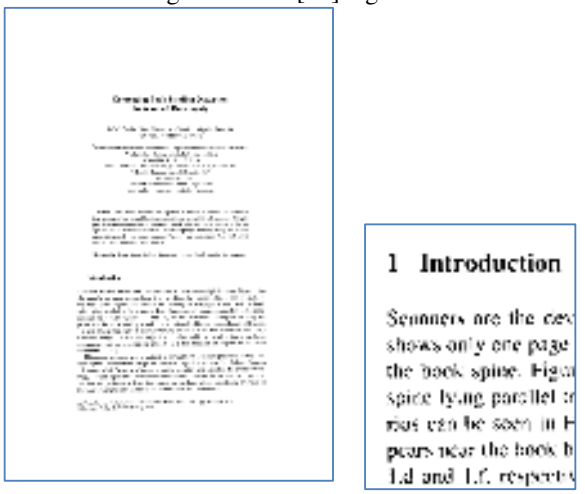


Figure 10 – Mello-Lins [16] binarization result for the iPhone 6 without flash.

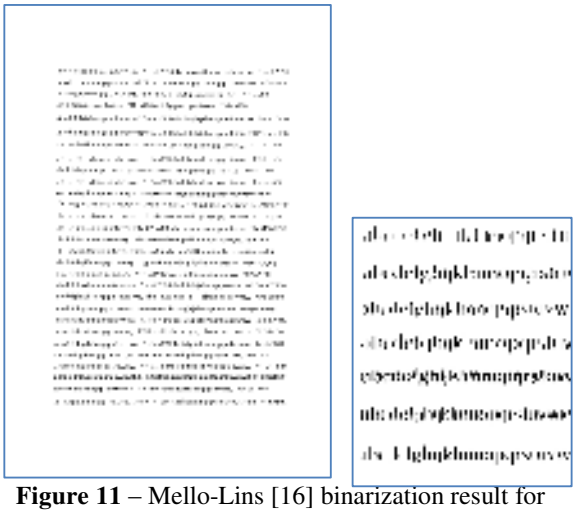


Figure 11 – Mello-Lins [16] binarization result for the iPhone 6 with flash on.

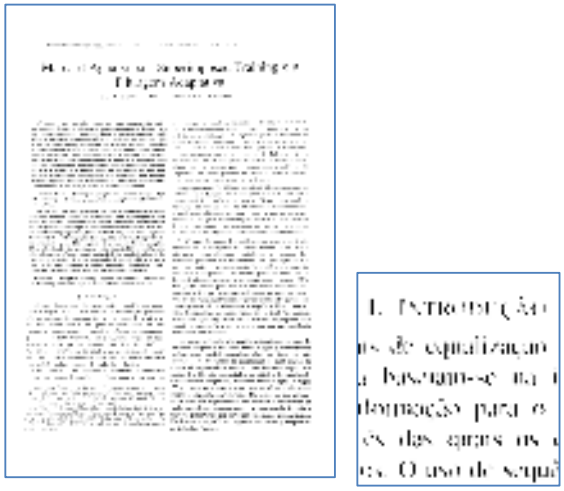


Figure 12 – Binarizatio result for an image iPhone SE with flash off binarized using Mello-Lins [16]

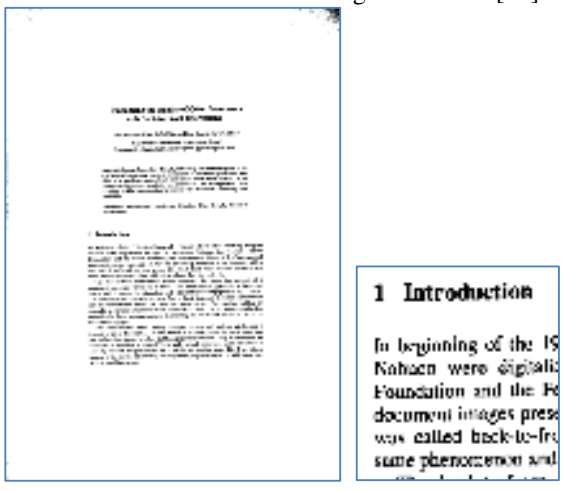


Figure 13 – Result for a iPhone SE photo document flash on binarized with RenyEntropy [13]

Article

A New Binarization Algorithm for Historical Documents

Marcos Almeida [1,*], Rafael Dueire Lins [1,2], Rodrigo Bernardino [1], Darlisson Jesus [1] and Bruno Lima [1]

[1] Federal University of Pernambuco, Recife-PE 50.740-560, Brazil; rdl.ufpe@gmail.com (R.D.L.); rbbernardino@gmail.com (R.B.); dmj.ufpe@gmail.com (D.J.); brunocesar182@hotmail.com (B.L.)
[2] Federal Rural University of Pernambuco; Recife-PE 52171-900, Brazil
* Correspondence: mmar@ufpe.br; Tel.: +55-81-2126-7129

Received: 31 October 2017; Accepted: 16 January 2018; Published: date

Abstract: Monochromatic documents claim for much less computer bandwidth for network transmission and storage space than their color or even grayscale equivalent. The binarization of historical documents is far more complex than recent ones as paper aging, color, texture, translucidity, stains, back-to-front interference, kind and color of ink used in handwriting, printing process, digitalization process, etc. are some of the factors that affect binarization. This article presents a new binarization algorithm for historical documents. The new global filter proposed is performed in four steps: filtering the image using a bilateral filter, splitting image into the RGB components, decision-making for each RGB channel based on an adaptive binarization method inspired by Otsu's method with a choice of the threshold level, and classification of the binarized images to decide which of the RGB components best preserved the document information in the foreground. The quantitative and qualitative assessment made with 23 binarization algorithms in three sets of "real world" documents showed very good results.

Keywords: documents; binarization; back-to-front interference; bleeding

1. Introduction

Document image binarization plays an important role in the document image analysis, compression, transcription, and recognition pipeline [1]. Binary documents claim for far less storage space and computer bandwidth for network transmission than color or grayscale documents. Historical documents drastically increase the degree of difficulty for binarization algorithms. Physical noises [2] such as stains and paper aging affect the performance of binarization algorithms. Besides that, historical documents were often typed, printed or written on both sides of sheets of paper and the opacity of the paper is often such as to allow the back printing or writing to be visualized on the front side. This kind of "noise", first called back-to-front interference [3], was later known as bleeding or show-through [4]. Figure 1 presents three examples of documents with such a noise extracted from the three different datasets used in this paper in the assessment of the proposed algorithm. If the document is exhibited either in true-color or gray-scale, the human brain is able to filter out that sort of noise keeping its readability. The strength of the interference present varies with the opacity of the paper, its permeability, the kind and degree of fluidity of the ink used, its storage, age, etc. Thus, the difficulty for obtaining a good binarization performance capable of filtering-out such a noise increases enormously, as a new set of hues of paper and printing colors appear. The direct application of binarization algorithms may yield a completely unreadable document, as the interfering ink of the backside of the paper overlaps with the binary one in the foreground. Several document image compression schemes for color images are based on "adding color" to a binary image. Such compression strategy is unable to handle documents with back-to-front interference [5]. Optical Character Recognizers (OCRs) are also unable to work properly for such documents. Several algorithms were developed specifically to binarize documents with back-to-front interference [3] [4, 6–9]. There is no binarization technique to be an all case winner as many parameters may interfere in the quality of the resulting image [9]. The development of new binarization algorithms is still an important research topic. International competitions on binarization algorithms, such as DIBCO - Document Image Binarization Competition [10], are an evidence of the relevance of this area.

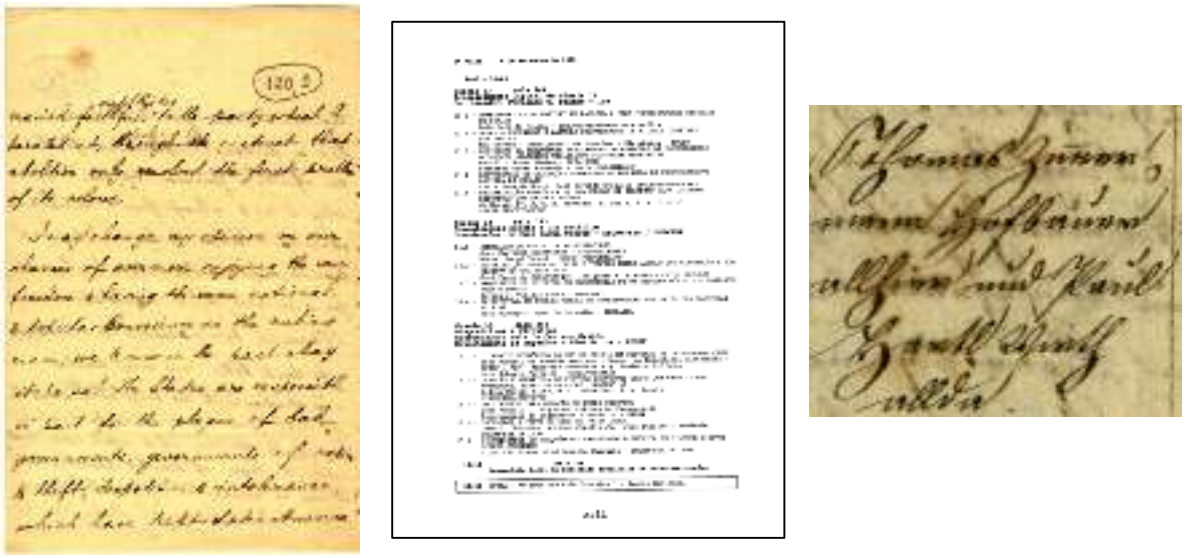


Figure 1. Images with back-to-front interference from the three test sets used in this paper: Nabuco bequest (**left**), LiveMemory (**center**) and DIBCO (**right**).

This paper presents a new global filter [1] to binarize documents, which is able to remove the back-to-front noise in a wide range of documents. Quantitative and qualitative assessments made in a wide variety of documents from three different "real-world" datasets (typed, printed and handwritten, using different kinds of paper, ink, etc.) allow to witness the efficiency of the proposed scheme.

2. The New Algorithm

The algorithm proposed here is performed in four steps: 1. decision-making for finding the vector of parameters of the image to be filtered, 2. filtering the image using a bilateral filter, 3. splitting the image into the RGB components, and performing their binarization using a method inspired by Otsu's algorithm for each RGB channel, and 4. choice of which of the RGB components best preserved the document information in the foreground, which is considered the final output of the algorithm. Figure 2 presents the block diagram of the proposed algorithm. The functionality of each block is detailed as follows.

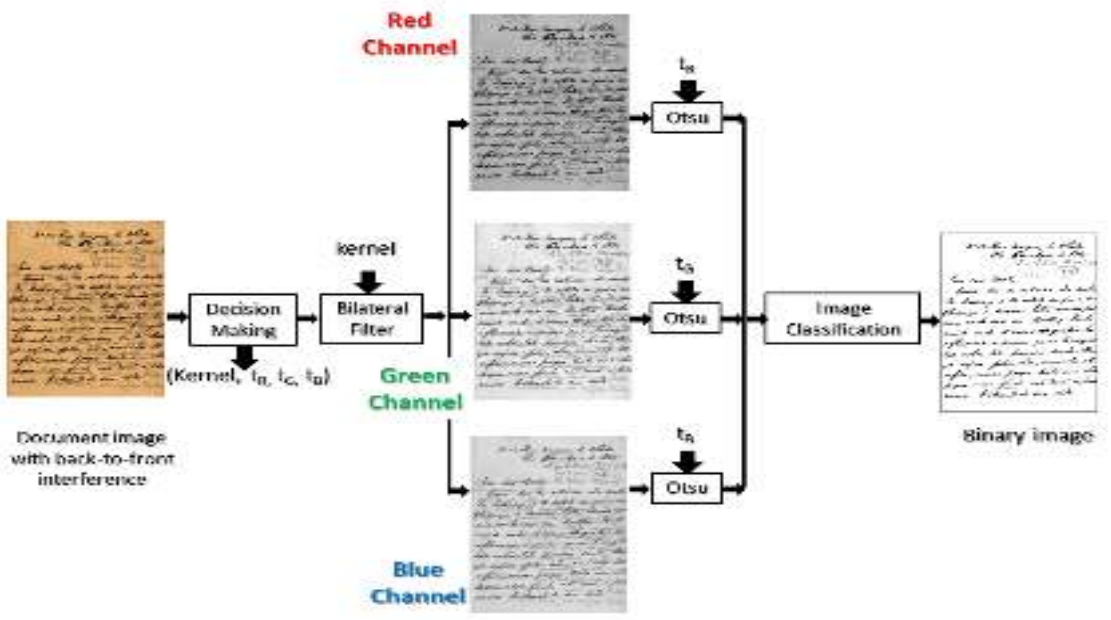


Figure 2. Block diagram of the proposed algorithm.

2.1. The Decision Making Block

The decision making block takes as input the image to be binarized and outputs a vector with four parameters: the value of the kernel (*kernel*) for the bilateral filter and three threshold values (t_R, t_G, t_B) that will be later used in the modified Otsu filtering.

The training of the binarization process proposed here is made with synthetic images which were generated as explained in Section 2.2. After filtering, the matrix of co-occurrence probabilities between the original image and of the binary image was calculated for each of the images in the document training set, whose generation is explained below.

The probabilistic structure applied in the analysis to each of the images in the training set is similar to the transmission of binary data in a Binary Asymmetric Channel, as shown in Figure 3. The probabilities P(f/b) and P(b/f) represent an additive noise in communication channels in information theory, here it represents the inability of the algorithm to correct the back-to-front interference of the image tested in the binarization process. The probabilities P(b/b) and P(f/f) are calculated from the pixel-to-pixel comparison of the binarized image generated by the proposed algorithm with the ground-truth image.

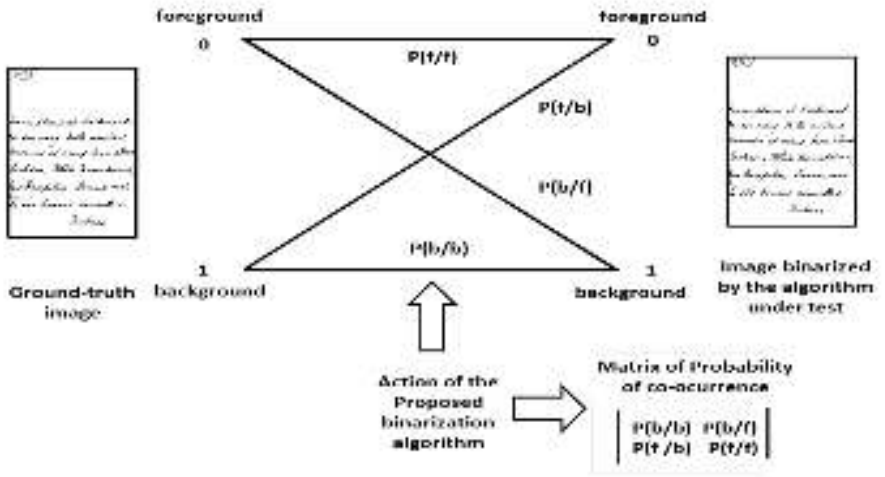


Figure 3. Generation of the co-occurrence matrix for each of the images in the training set.

The background-background probability is a function that needs to be optimized in the decision-making block, mapping background pixels (paper) from the original image onto white pixels of the binary image. It depends of all the parameters of the original image texture, strength of the back to front interference (simulated by the coefficient α), paper translucidity, etc. for each RGB channel. Thus, one can represent this dependence as:

$$P(b/b) = f(\alpha,R,G,B). \quad (1)$$

The optimal threshold t_c^* for each channel is calculated in the decision-making block, the index c can be R, G or B, maximizing P(b/b):

$$t_c^* = MaxP(b/b), \quad (2)$$

subject to a given criterion $P(f/f) \geq M$. The criterion used here was M = 97%, that is at most 3% of the foreground pixels may be incorrectly mapped. During the training phase, the best t_c^* will be chosen from the three channels, which best maximizes the P(b/b) for each of the images in the training set. The matrix of co-occurrence probability is calculated and the decision maker chooses the best binary image. The decision-making block was trained with 32,000 synthetic images in such a way to, given a real image to be binarized, it finds the optimal threshold parameters.

2.2. Generating Synthetic Images

The Decision-Making Block needs training to "learn" about the optimal threshold parameters and the value of the kernel to be used in the bilateral filter. Such training must be done using controlled images which are synthesized to mimic the different degrees of back-to-front interference, paper aging, paper translucidity, etc. Figure 4 presents the block diagram for the generation of synthetic images. Two binary images of documents of different nature (typed, handwritten with different pens, printed, etc.) are taken: F—front and V—verso (back). The front image is blurred with a weak Gaussian filter to simulate the digitalization noise [1], the hues that appear in after document scanning.

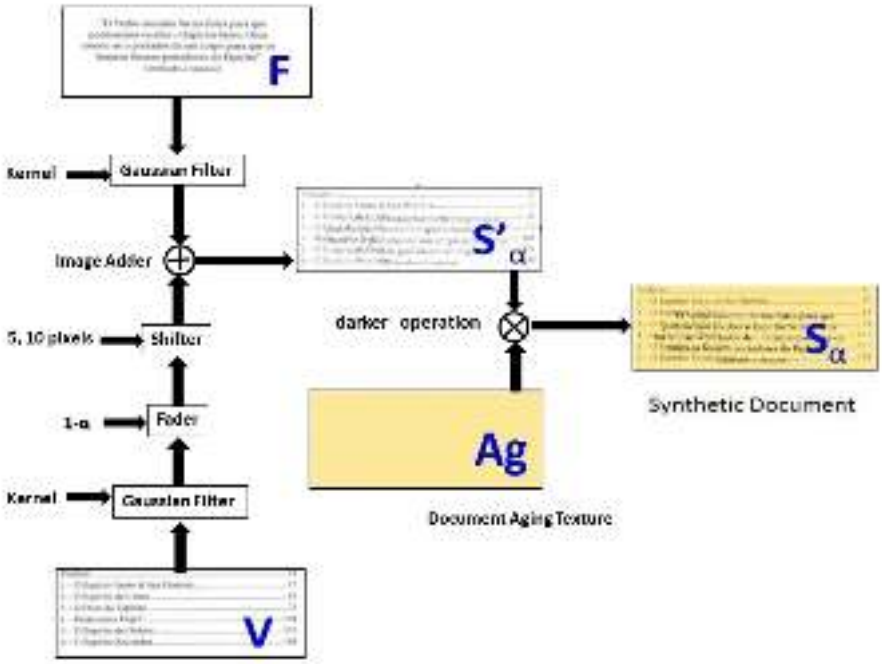


Figure 4. Block diagram of the scheme for the generation of synthetic images for the Decision-Making Block.

The verso image is "blurred" by passing through two different Gaussian filters that simulate the low-pass effect of the translucidity of the verso as seen in the front part of the paper. Two different parameters were used to simulate two different classes of paper translucidity. The "blurred" verso image is now faded with a coefficient α varying between 0 and 1 in steps of 0.01. Then, a circular shift of the lines of the document is made of either 5 or 10 pixels, to minimize the chances of the front and verso lines coincide entirely. Finally, the two images are overlapped by performing a "darker" operation pixel-by-pixel in the images. Paper texture is added to the image to simulate the effect of document aging. The texture pattern was extracted from document from late 19th century to the year 2000. The analysis of 3450 documents representative of a wide variety of documents of such a period was analyzed yielding 100 different clusters of textures. The synthetic texture to be applied to the image to simulate paper aging is generated using those 100 clusters by image quilting [11] and randomly, as explained in reference [9]. The training performed in the current version of the presented algorithm was made with 16 of those 200 synthetic textures. The total number of images used for training here was thus 16 (textures), times 10 ($0 < \alpha < 1$ in steps of 0.10), times 2 blur parameters for the Gaussian filters, times 100 different binary images, totaling 32,000 images. Details of the full generation process of the synthetic image database are out of the scope of this paper and may be found in reference [9].

2.3. The Bilateral Filter

The bilateral filter was first introduced by Aurich and Weule [12] under the name "nonlinear Gaussian filter". It was later rediscovered by Tomasi and Manduchi [13] who called it the "bilateral filter" which is now the most commonly used name according to reference [14].

The bilateral filter is a technique to smoothen images while preserving their edges. The filter output at each pixel is a weighted average of its neighbors. The weight assigned to each neighbor decreases with both the distance values among pixels of the image plane (the spatial domain S) and the distance on the intensity axis (the range

domain R). The filter applies spatial weighted averaging without smoothing the edges. It combines two Gaussian filters; one filter works in the spatial domain, while the other filter works in the intensity domain. Therefore, not only the spatial distance but also the intensity distance is important for the determination of weights. The bilateral filter combines two stages of filtering. These are the geometric closeness (i.e., filter domain) and the photometric similarity (i.e., filter range) among the pixels in a window of size N × N. Let I(x,y) be a 2D discrete image of size N × N, such that {x,y} ∈ {0, 1, ..., N − 1} X {0, 1, ..., N − 1}. Assume that I(x,y) is corrupted by an additive white Gaussian noise of variance σ_n^2. For a pixel (x,y), the output of a bilateral filter can be as described by Equation (1):

$$I_{BF}(\mathrm{x},\mathrm{y}) = \frac{1}{K}\sum_{i=\mathrm{x}-\mathrm{d}}^{x+d}\sum_{j=\mathrm{y}-\mathrm{d}}^{\mathrm{x}+d} G_s(i;x,j;y)\, G_r[I(i,j),I(x,y)]I(i,j), \tag{3}$$

where I(x,y) is the pixel intensity in the image before applying the bilateral filter, I_{BF}(x,y) is the resulting pixel intensity after applying the bilateral filter and d is a non-negative integer such that (2d + 1) × (2d + 1) stands for the size of the neighborhood window. Let G_s and G_r be the domain and the range components, respectively, which are defined as:

$$G_s(i;x,j;y) = e^{-\frac{\left|(i-x)^2+(j-y)^2\right|}{2\sigma_s^2}} \tag{4}$$

and

$$G_r\big(I(i,j);I(x,y)\big) = e^{-\frac{|I(i,j)-I(x,y)|^2}{2\sigma_r^2}} \tag{5}$$

The normalization constant K is given as:

$$\mathrm{K} = \frac{1}{\sum_{i=\mathrm{x}-\mathrm{d}}^{x+d}\sum_{j=\mathrm{y}-\mathrm{d}}^{\mathrm{x}+d} G_s(i;x,j;y)\, G_r[I(i,j),I(x,y)]} \tag{6}$$

Equations (4) and (5) show that the bilateral filter has three parameters: σ_s^2 (the filter domain), σ_r^2 (the filter range), and the third parameter is the window size N × N [15].

The geometric spread of the bilateral filter is controlled by σ_s^2. If the value of σ_s^2 is increased, more neighbours are combined in the diffusion process yielding a "smoother" image, while σ_r^2 represents the photometric spreading. Only pixels with a percentage difference of less than σ_r^2 are processed [13].

2.4. Otsu Filtering

After passing through the bilateral filter, the image is split into its original (non-gamma corrected) Red, Green and Blue components, as shown in the block diagram in Figure 2. The kernel of the bilateral filter alters the balance of the colors in the original image in such a way to widen the differences between the color of the front and back-to-front interference. A modified version of Otsu [16] algorithm is applied to each RGB channel using the thresholds determined by the Decision Making Block, which may be considered as the "optimal" threshold for each RGB channel, and then three binary images are generated.

2.5. Image Classification

The image classification block was also trained with the synthetic images in such a way to analyze the three binary images generated in each of the channels and outputs the one that is considered the best one. This decision was also made by a naïve Bayes automatic classifier which was trained using the calculated co-occurrence matrix for each of the 32,000 synthetic images by comparing each of them with the original ground truth image, the Front image.

3. Experiments and Results

As already explained, the enormous variety of kinds of text documents makes extremely improbable that one single algorithm is able to satisfactorily binarize all kinds of documents. Depending on the nature (or degree of complexity) of the image several or no algorithm will be able to provide good results. This paper follows the assessment methodology proposed in reference [9], in which one compares the numbers of background and foreground pixels correctly matched with a ground-truth image. Twenty-three binarization algorithms were tested using the methodology described:

1. Mello-Lins [5]
2. DaSilva-Lins-Rocha [6]
3. Otsu [16]
4. Johannsen-Bille [17]
5. Kapur-Sahoo-Wong [18]
6. RenyEntropy (variation of [18])
7. Li-Tam [19]
8. Mean [20]
9. MinError [21]
10. Mixture-Modeling [22]
11. Moments [23]
12. IsoData [24]
13. Percentile [25]
14. Pun [26]
15. Shanbhag [27]
16. Triangle [28]
17. Wu-Lu [29]
18. Yean-Chang-Chang [30]
19. Intermodes [31]
20. Minimum (variation of [31])
21. Ergina-Local [32]
22. Sauvola [33]
23. Niblack [34]

A ground-truth image for each "real" world one is needed to allow a quantitative assessment of the quality of the final binary image. Only the DIBCO dataset [10] had ground-truth images available. This makes the assessment task of real-world images extremely difficult [35]. All care must be taken to guarantee the fairness of the process. The ground-truth images for the other datasets were generated by applying the 23 algorithms above and the bilateral algorithm to all the test images in the Nabuco [7] and LiveMemory [36] datasets. Visual inspection was made to choose the best binary image in a blind process, a process in which the people who selected the best image did not know which algorithm generated it. To increase the degree of fairness and the number of filtering possibilities, the three component images produced by the Decision Making block were all analyzed. The binary images chosen using the methodology above went through salt-and-pepper filtering and were used as ground-truth image for the assessment below. All the processing time figures presented in this paper are from Intel i7-4510U@ 2.00 GHzx2, 8 GB RAM, running Linux Mint 18.2 64-bit. All algorithms were coded in Java, possibly by their authors.

3.1. The Nabuco Dataset

The Nabuco bequest encompasses about 6,500 letters and postcards written and typed by Joaquim Nabuco [7], totaling about 30,000 pages. Such documents are of great interest to whoever studies the history of the Americas, as Nabuco was one of the key figures in the freedom of black slaves, and was the first Brazilian Ambassador to the U.S.A. The documents of Nabuco were digitalized by the second author of this paper and the historians of the Joaquim Nabuco Foundation using a table scanner in 200 dpi resolution in true color (24 bits per pixel), back in 1992 to 1994. Due to serious storage limitations then, images were saved in the jpeg format with 1% loss. The historians in the project concluded that 150 dpi resolution would suffice to represent all the graphical elements in the documents, but choice of the 200-dpi resolution was made to be compatible with the FAX devices widely used then. About 200 of the documents in the Nabuco bequest exhibited back-to-front interference. The 15 document images used in this dataset were chosen for being representative of the diversity of documents in such a universe.

Table 1. Binarization results for images from Nabuco bequest.

AlgName	P(f/f)	P(b/b)	Time (s)
IsoData	98.08 ± 3.39	99.38 ± 0.60	0.0171
Otsu	98.08 ± 3.39	99.36 ± 0.63	0.0159

Bilateral	99.57 ± 1.23	99.29 ± 0.93	1.0790
Huang	99.40 ± 2.14	98.69 ± 0.88	0.0200
Moments	99.39 ± 1.34	98.40 ± 1.70	0.0160
Ergina-Local	99.99 ± 0.03	98.13 ± 0.64	0.3412
RenyEntropy	100.00	97.56 ± 1.17	0.0188
Kapoo-Sahoo-Wong	100.00	97.51 ± 1.07	0.0172
Yean-Chang-Chang	100.00	97.38 ± 1.26	0.0161
Triangle	100.00	95.94 ± 1.46	0.0160
Mello-Lins	98.61 ± 5.14	89.63 ± 24.43	0.0160
Mean	100.00	81.77 ± 5.99	0.0168
Johannsen-Bille	98.87 ± 2.97	59.77 ± 48.80	0.0164
Pun	100.00	55.44 ± 2.57	0.0185
Percentile	100.00	53.21 ± 1.33	0.0185
Sauvola	85.51 ± 12.93	99.95 ± 0.11	1.2977
Niblack	99.75 ± 0.34	77.06 ± 5.63	0.2135

Table 1 presents the quantitative results obtained for all the documents in this dataset. P(f/f) stands for the ratio between the number of foreground pixels in the original image mapped onto black pixels and the number of black pixels in the ground-truth image. Similarly, P(b/b) is proportion between the number of background pixels in the original image mapped onto white pixels of the binary image and the number of white pixels in the ground-truth image. The figures for P(b/b) and P(f/f) are followed by "±" and the value of the standard deviation. The time corresponds to the mean processing time elapsed by the algorithm to process the images in this dataset. The results were ranked in P(b/b) decreasing order.

The results presented in Table 1 shows the bilateral filter in third place for this dataset in terms of image quality, however the standard deviation is much lower than the two first. That implies that its quality is more stable for the various document images in this dataset. Figure 5 presents the document for which the bilateral filter presented the best and the worst results in terms of image quality with two zoomed areas from the original and the binarized document.

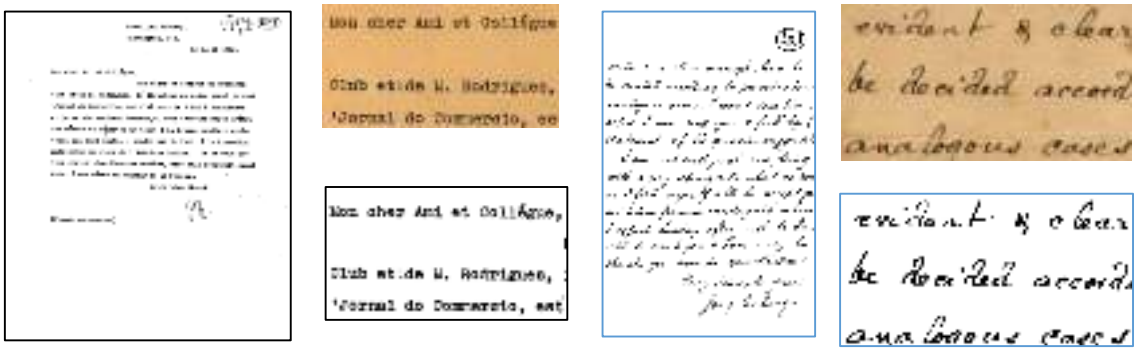


Figure 5. Historical documents from Nabuco bequest with the best ((**left**)—P(f/f) = 100, P(b/b) = 99.99) and the worst ((**right**)—P(f/f) = 89.76, P(b/b) = 99.98) binarization results for the bilateral filter with zooms from the original (**top**) and binary (**bottom**) parts.

3.2. The LiveMemory Dataset

This dataset encompasses 15 documents with 200 dpi resolution selected from the over 8,000 documents from the LiveMemory project that created a digital library with all the proceedings of technical events from the Brazilian Telecommunications Society. The original proceedings were offset printed from documents either typed or electronically produced. Table 2 presents the performance results for the 12 best ranked algorithms. The bilateral filter obtained the best results in terms of image filtering. It is worth observing that in the case of the worst quality image (Figure 6 right) the performance degraded for all the algorithms. This behavior is due to the shaded area in the hard-bound spine of the volumes of the proceedings.

Table 2. Binarization results for images from the LiveMemory project.

AlgName	P(f/f)	P(b/b)	Time (s)
Bilateral	100.00	98.90 ± 1.07	3.3325
IsoData-ORIG	99.56 ± 0.69	98.61 ± 1.99	0.0734

Otsu	99.60 ± 0.68	98.57 ± 2.08	0.0735
Moments	99.99 ± 0.03	97.91 ± 1.87	0.0716
Ergina-Local	98.98 ± 2.82	97.62 ± 1.04	0.9917
Huang	99.93 ± 0.27	96.42 ± 4.20	0.0865
Triangle	100.00	94.24 ± 2.15	0.0728
Mean	100.00	83.58 ± 5.59	0.0747
Niblack	99.76 ± 0.76	78.31 ± 2.97	0.6710
Pun	100.00	55.28 ± 3.60	0.0800
Percentile	100.00	53.91 ± 1.96	0.0795
Kapur-Sahoo-Wong	98.62 ± 4.92	97.15 ± 1.44	0.0729

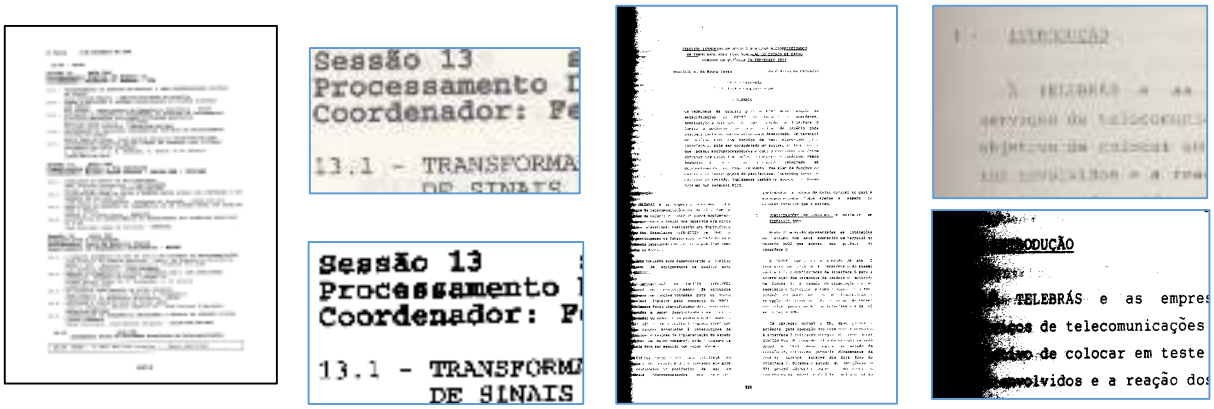


Figure 6. Images from LiveMemory with the best ((**left**)—P(f/f) = 100.00, P(b/b) = 99.99) and the worst ((**right**)—P(f/f) = 100.00, P(b/b) = 95.97) binarization results for the bilateral filter with zooms from the original (**top**) and binary (**bottom**) parts.

3.3. The DIBCO Dataset

This dataset has all the 86 images from the Digital Image Binarization Contest from 2009 to 2016. Table 3 presents the results obtained. The performance of the bilateral filter in this set may be considered good, in general. The overall performance of the bilateral filter was strongly degraded by the single image shown in Figure 7 (right) in which the P(f/f) of 25.93 drastically dropped the average result of the algorithm in this test set. It is important to remark that such an image is almost unreadable even for humans and that it degraded the performance of all the best algorithms.

Table 3. Binarization results for images from Document Image Binarization Competition (DIBCO).

AlgName	P(f/f)	P(b/b)	Time (s)
Ergina-local	91.37 ± 6.25	99.88 ± 1.89	0.1844
RenyEntropy	90.13 ± 14.19	96.77 ± 3.50	0.0125
Yean-Chang-Chang	90.61 ± 14.44	96.16 ± 4.35	0.0112
Moments	90.75 ± 9.91	95.80 ± 5.19	0.0112
Bilateral	92.99 ± 9.06	90.78 ± 16.01	0.6099
Huang	95.62 ± 6.37	84.22 ± 18.36	0.0147
Triangle	96.40 ± 5.72	80.80 ± 23.32	0.0113
Mean	99.35 ± 1.14	78.99 ± 9.35	0.0115
MinError	92.79 ± 23.46	74.29 ± 19.36	0.0115
Pun	99.68 ± 0.82	56.20 ± 6.18	0.0122
Percentile	99.71 ± 0.72	55.06 ± 3.58	0.0121
Sauvola	59.75 ± 30.06	99.58 ± 079	0.6933
Niblack	95.91 ± 2.31	78.61 ± 5.69	0.1241

4. Conclusions

Historical documents are far more difficult to binarize as several factors such as paper texture, aging, thickness, translucidity, permeability, the kind of ink, its fluidity, color, aging, etc. all may influence the performance of the algorithms. Besides all that, many historical documents were written or printed on both sides of translucent paper, giving rise to the back-to-front interference.

This paper presents a new binarization scheme based on the bilateral filter. Experiments performed in three datasets of "real world" historical documents with twenty-three other binarization algorithms. Image quality and processing time figures were provided, at least for the top 10 algorithms assessed. The results obtained showed that the proposed algorithm yields good quality monochromatic images that may compensate its high computational cost. This paper provides evidence that no binarization algorithm is an "all-kind-of-document" winner, as the performance of the algorithms varied depending of the specific features of each document. A much larger test set of synthetic about 250,000 images is currently under development, such a test set will allow much better training of the Decision Making and Image Classifier blocks of the bilateral algorithm presented. The authors are currently

attempting to integrate the Decision Making and Image Classifier blocks in such a way to anticipate the choice of the best component image. This would highly improve the time performance of the proposed algorithm.

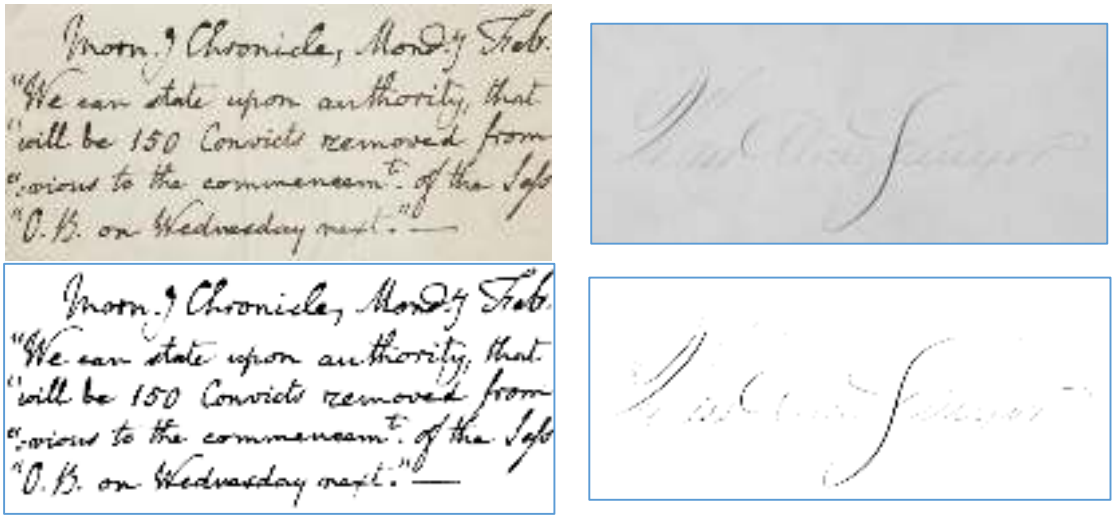


Figure 7. Two documents from DIBCO dataset: (**left-top**) original image (**left-bottom**) binary image obtained using the bilateral filter best result (P(f/f) = 97.05, P(b/b) = 99.88); (**right-top**) original image. (**right-bottom**) the worst binarization results for the bilateral filter (P(f/f) = 25.93, P(b/b) = 99.99).

The authors of this paper are promoting a paramount research effort to assess the largest possible number of binarization algorithms for scanned documents using over 5.4 million synthetic images in the DIB-Document Image Binarization platform. An image matcher, a more general and complex version of the Decision Making block, is also being developed and trained with that large set of images, in order to whenever fed with a real world image, to be able to match with the most similar synthetic one. Once that match is made, the most suitable binarization algorithms are immediately known. If this paper were accepted, all the test images and algorithms will be included in the DIB platform. The preliminary version of the DIB-Document Image Binarization platform and website is publicly available at https://dib.cin.ufpe.br/.

Acknowledgments: The authors of this paper are grateful for the referees whose comments much helped in improving the current version of this paper and to those researchers who made the code of their algorithms publicly available for testing and performance analysis and to the DIBCO team from making their images publicly available. The authors also acknowledge the partial financial support of to CNPq and CAPES—Brazilian Government.

Author Contributions: Marcos Almeida and Rafael Dueire Lins contributed in equal proportion to the development of the algorithm presented in this paper, which was written by the latter author. Bruno Lima was responsible for the first implementation of the algorithm proposed. Rodrigo Bernardino and Darlisson Jesus re-implemented the algorithm and were also responsible for all the quality and time assessment figures presented here.

Conflicts of Interest: The authors declare no conflict of interest

References

1. Chaki, N.; Shaikh, S.H.; Saeed, K. *Exploring Image Binarization Techniques*; Springer: New Delhi, India, **2014**.
2. Lins, R.D. A Taxonomy for Noise in Images of Paper Documents-The Physical Noises. In Proceedings of the International Conference Image Analysis and Recognition, Halifax, NS, Canada, 6–8 July 2009; Springer Verlag, Germany, **2009**; Volume 5627, pp. 844–854.
3. Lins, R.D. An Environment for Processing Images of Historical Documents. *Microprocess. Microprogr.* **1995**, *40*, 939–942.

4. Sharma, G. Show-through cancellation in scans of duplex printed documents. *IEEE Trans. Image Process.* **2001**, *10*, 736–754.
5. Mello, C.A.B.; Lins, R.D. Generation of Images of Historical Documents by Composition. In Proceedings of the 2002 ACM Symposium on Document Engineering, New York, NY, USA, 8–9 November **2002**; pp. 127–133.
6. Silva, M.M.; Lins, R.D.; Rocha, V.C. Binarizing and Filtering Historical Documents with Back-to-Front Interference. In Proceedings of the 2006 ACM Symposium on Applied Computing, New York, NY, USA, 23–27 April **2006**; pp. 853–858.
7. Lins, R.D. Nabuco – Two Decades of Processing Historical Documents in Latin America. *Journal of Universal Computer Science.* **2011,** *17*, 151–161.
8. Roe, E.; Mello, C.A.B. Binarization of Color Historical Document Images Using Local Image Equalization and XDoG. In Proceedings of the 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, 25–28 August **2013**; 205–209.
9. Lins, R.D.; Almeida, M.A.M.; Bernardino, R.B.; Jesus, D.; Oliveira, J.M. Assessing Binarization Techniques for Document Images. In Proceedings of the ACM Symposium on Document Engineering, Valletta, Malta, 4–7 September **2017**.
10. Pratikakis, I. ICDAR 2017 Competition on Document Image Binarization (DIBCO 2017). In Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition, Kyoto, Japan, 13–15 November **2017**; pp.2379–2140.
11. Efros, A.A.; Freeman, W.T. Image quilting for texture synthesis and transfer. In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01), New York, NY, USA, **2001**; pp. 341–346.
12. Aurich, V.; Weule, J.B. Non-Linear Gaussian Filters Performing Edge Preserving Diffusion. In Proceedings of the DAGM Symposium, London, UK, 13–15 September **1995**; pp. 538–545.
13. Tomasi, C.; Manduchi, R. Bilateral Filtering for Gray and Color Images. In Proceedings of the 6th International Conference on Computer Vision, Washington, DC, USA, 4–7 January **1998**; pp. 836–846.
14. Paris, P.; Kornprobst, P.; Tumblim, J.; Durand, F. Bilateral Filtering: Theory and Applications. *Found. Trends Comput. Graph. Vis.* **2008**, *4*, 1–73.
15. Shyam Anand, C.; Sahambi, J.S. Pixel Dependent Automatic Parameter Selection for Image Denoising with Bilateral Filter. *Int. J. Comput. Appl.* **2012**, *45*, 975–8887.
16. Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66.
17. Johannsen, G.; Bille, J.A. A Threshold Selection Method Using Information Measure. In Proceedings of the 6th International Conference on Pattern Recognition (ICPR'82), Munich, Germany, 19–22 October **1982**; pp. 140–143.
18. Kapur, N.; Sahoo, P.K.; Wong, A.K.C. A New Method for Gray-Level Picture Thresholding Using the Entropy of the Histogram. *Comput. Vis. Graph. Image Process.* **1985**, *29*, *273–285*.
19. Li, C.H.; Tam, P.K.S. An iterative algorithm for minimum cross entropy thresholding. *Pattern Recognit. Lett.* **1998**, *19*, 771–776.
20. Glasbey, C.A. An analysis of histogram-based thresholding algorithms. *Graph. Models Image Process.* **1993**, *55*, 532–537.
21. Kittler, J.; Illingworth, J. Minimum error thresholding. *Pattern Recognit.* **1986**, *19*, 41–47.
22. Mixture Modeling. ImageJ. Available online: http//imagej.nih.gov/ij/plugins/mixture-modeling.html (accessed on 20 January **2018**).
23. Tsai, W.H. Moment-preserving thresholding: A new approach. *Comput. Vis. Graph. Image Process.* **1985**, *29*, 377–393.
24. Doyle, W. Operation useful for similarity-invariant pattern recognition. *J. Assoc. Comput. Mach.* **1962**, *9*, 259–267.
25. Pun, T. Entropic Thresholding, A New Approach. *Comput. Vis. Graph. Image Process.* **1981**, *16*, 210–239.
26. Shanbhag, A.G.G. Utilization of Information Measure as a Means of Image Thresholding. *Comput. Vis. Graph. Image Process.* **1994**, *56*, 414–419.
27. Zack, G.W.; Rogers, W.E.; Latt, S.A. Automatic measurement of sister chromatid exchange frequency. *J. Histochem. Cytochem.* **1977**, *25*, 741–753.
28. Wu, U.L.; Songde, A.; Haqing, L.U.A. An Effective Entropic Thresholding for Ultrasonic Imaging. In proceedings of the International Conference Pattern Recognition, Brisbane, Australia, 16–20 August **1998**; pp. 1522–1524.
29. Yen, J.C.; Chang, F.J.; Chang, S. A New Criterion for Automatic Multilevel Thresholding. *IEEE Trans. Image Process.* **1995**, *4*, 370–378.

30. Ridler, T.W.; Calvard, S. Picture Thresholding Using an Iterative Selection Method. *IEEE Trans. Syst. Man Cybern.* **1978**, *8*, 630–632.
31. Prewitt, M. S. and Mendelsohn, M. L. The Analysis of Cell Images. *Ann. N. Y. Acad. Sci.* **1996**. *Volume 128, N. 3, pp. 836-846*.
32. Kavallieratou, E.; Stamatatos, S.; Adaptive binarization of historical document images. In Proceedings of the 18th International Conference on Pattern ICPR 2006, Hong Kong, China, 20–24 August **2006**; Volume 3.
33. Sauvola, J.; Pietikainen, M. Adaptive document image binarization. *Pattern Recognit.* **2000**, *33*, 225–236, doi:10.1016/S0031-320300055-2.
34. Niblack, W. An introduction to Digital Image Processing. Prentice-Hall: Upper Saddle River, NJ, USA, **1986**
35. Ntirogiannis, K.; Gatos, B.; Pratikakis, I. Performance Evaluation Methodology for Historical Document Image Binarization. *IEEE Trans. Image Process.* **2013**, *22*, 595–609.
36. Lins, R. D., Silva, G. F. P., Torreão, G., Alves, N. F. Efficiently Generating Digital Libraries of Proceedings with The LiveMemory Platform. *In: IEEE International Telecommunications Symposium, IEEE Press* **2010**. *pp. 119-125.*