Pós-Graduação em Ciência da Computação

Robson David Montenegro

RETIFICAÇÃO DE IMAGENS DE DOCUMENTOS CAPTURADOS POR DISPOSITIVOS MÓVEIS UTILIZANDO TRANSFORMADA DE HOUGH E HISTOGRAMAS DE GRADIENTES ORIENTADOS

Dissertação de Mestrado

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE
2015

Universidade Federal de Pernambuco
Centro de Informática
Pós-graduação em Ciência da Computação

Robson David Montenegro

RETIFICAÇÃO DE IMAGENS DE DOCUMENTOS CAPTURADOS POR DISPOSITIVOS MÓVEIS UTILIZANDO TRANSFORMADA DE HOUGH E HISTOGRAMAS DE GRADIENTES ORIENTADOS

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: *Carlos Alexandre Barros de Mello*

RECIFE
2015

Robson David Montenegro

Retificação de Imagens de Documentos capturados por Dispositivos Móveis utilizando Transformada de Hough e Histogramas de Gradientes Orientados/ Robson David Montenegro. – RECIFE, 2015-

123 p. : il. (algumas color.) ; 30 cm.

Orientador Carlos Alexandre Barros de Mello

Dissertação de Mestrado – Universidade Federal de Pernambuco, 2015.

1. Palavra-chave1. 2. Palavra-chave2. I. Orientador. II. Universidade xxx. III. Faculdade de xxx. IV. Título

CDU 02:141:005.7

À Julieta Montenegro.

Agradecimentos

À minha família, pelo apoio incondicional. Em especial, à minha mãe, pela vida dedicada a educação dos filhos.

Ao meu orientador, Professor Dr. Carlos Alexandre Barros de Mello, por acreditar neste trabalho e pelas orientações fundamentais.

À Bruna Montenegro, pela disposição e ajuda providencial neste trabalho e, principalmente, pelos momentos de descontração que dividimos. Estes fazem tudo valer a pena.

Aos colegas da Stefanini Document Solutions, que ajudaram diretamente para o desenvolvimento desta dissertação.

Às instituições responsáveis pela minha formação: Colégio Equipe, POLI/UPE e CIn/UFPE.

À Tia Bia, Vovó Esther e Tia Lila, por participarem diretamente da minha educação nos momentos mais importantes.

À Kenelly, pela ajuda operacional, disposição, paciência, por sempre acreditar em mim e, principalmente, por estar sempre presente.

Baby, you are gonna miss that plane.

—CELINE

Resumo

Diversas maneiras de armazenamento e transmissão de informação em meio digital estão disponíveis devido ao contínuo crescimento tecnológico. Porém, grande parte das informações relevantes permanece armazenada em meio físico, como: livros, certidões, contratos e documentos pessoais. Existe um grande esforço para realizar a transposição dessas informações para meios digitais de forma a facilitar o acesso e utilização de meios de comunicação mais modernos. Os *scanners* fornecem a maneira mais popular de se obter esta transposição, porém, estes dispositivos muitas vezes não oferecem portabilidade e custo adequados. A utilização de dispositivos móveis, tais como celulares, para captura de imagens de documento tem se mostrado uma alternativa viável aos tradicionais *scanners* de mesa. Isto se deve a sua facilidade de uso, portabilidade e barateamento de seu hardware que facilitou sua popularização. Porém, por se tratar de captura em um ambiente menos controlado, documentos digitalizados desta forma apresentam distorções que comprometem a sua legibilidade tais como: perspectiva, embassamento, baixa resolução, interação do conteúdo com o *background* e curvatura das linhas de texto. Trabalhos recentes tratam este problema utilizando diferentes abordagens, muitos destes, de maneira eficaz. Entretanto, estas técnicas são fortemente dependentes do conteúdo textual presente nos documentos. Esta dissertação apresenta uma nova abordagem para realizar correção de imagens capturadas por dispositivos móveis baseado-se apenas em características morfológicas do documento. O método é dividido em três etapas. A primeira é o pré-processamento responsável por ajustar o contraste da imagem. Depois, as bordas do documento são definidas utilizando uma abordagem híbrida do descritor de Gradiente de Histogramas Orientados com a Transformada de Hough. Por último, a superfície deformada do documento é mapeada para uma superfície retangular corrigida. O algoritmo foi experimentado em diversas configurações de seus parâmetros livres em uma base de documentos pessoais coletada para este trabalho. O menor erro obtido foi de 4,08% e, além disto, as imagens corrigidas foram processadas por sistemas de OCRs e os resultados quantitativos mostram que o algoritmo proposto melhorou substancialmente a legibilidade das imagens.

Palavras-chave: Processamento de Imagens, Visão Computacional, Retificação de Imagens

Abstract

Several ways of information storage and transmission in digital media emerged due to technological progress. However, much of the relevant information remains stored on physical media, such as books, certificates, contracts and personal documents. Much effort has been made to carry out the transposition of this information to digital media in order to facilitate access and use modern communication channels. The scanners provide the most popular way to obtain this transposition, however, these devices often do not offer adequate portability and are usually expensive. The use of mobile devices such as cell phones, for document imaging has proven to be a suitable alternative to traditional table scanners. This is due to its ease of use, portability and cheapness of their hardware which facilitated its popularization. However, documents acquired in a less controlled environment, have distortions that muddle its readability such as perspective, blur, low resolution, interaction of the content and the background and curled text lines. Recent works address this problems using different approaches, many of these, efficiently. However, these techniques are strongly dependent on the textual content in the document images. This dissertation presents a new algorithm to rectify images acquired by mobile devices based only on morphological features of the document image. The method is divided into three steps. First is the preprocessing when the image contrast is adjusted. Then, the document edges are determined using a hybrid approach of Hough Transform and Histogram of Oriented Gradients descriptor. Finally, the warped surface of the document is mapped to a rectangular surface. The algorithm has been tested in several configurations in a personal document image base collected for this work. The best error rate was 4.08 % and, moreover, the corrected images were processed by OCR systems and quantitative results shows that the proposed algorithm has significantly improved readability of the images.

Keywords: Image Processing, Computer Vision, Dewarping

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Lista de Acrônimos

Sumário

1

Introdução

No contexto desta dissertação, "documento"se refere a um conjunto de informações legíveis para humanos, normalmente, formado por papéis contendo textos e elementos gráficos. Não existem razões para crer que o armazenamento e veiculação de documentos deixem de existir em papel (**?**). Muitos livros existem apenas em papel; contratos – por questões legais – precisam ser preservados em meio físico; sofisticados métodos forenses de detecção de fraude foram desenvolvidos apenas para papel, como em cheques bancários.

No entanto, o papel é frágil, suscetível ao envelhecimento e os canais de transmissão convencionais são lentos. Por isso, a digitalização de documentos tem sido amplamente utilizada como forma de preservação da informação em formato eletrônico (**?**) e para aproveitar as tecnologias atuais de transmissão de informações por meio digital. Soma-se a isto, a possibilidade de armazenamento em grande escala, ocupando pouco espaço e acesso rápido aos dados armazenados.

Uma das grandes vantagens que surgiu a partir do armazenamento de documentos digitalizados é a capacidade de recuperação rápida das informações contidas nestes documentos. A busca por palavras chaves ou conteúdos textuais específicos se torna bem mais objetiva com ajuda de Sistemas de Reconhecimento Automático de Documentos (SRAD). Ao invés do humano procurar página a página em vários livros, delega-se isto ao computador. Estes documentos são processados por um sistema de Reconhecimento Óptico de Caracteres (OCR), que são capazes de transformar uma imagem contendo textos em informação textual indexável.

Porém, realizar o reconhecimento através de sistemas baseados em SRAD é um problema não trivial. O início deste processo é chamado de digitalização, quando ocorre a transformação do domínio do documento do meio físico para o formato digital. Esta transformação é realizada através de sensores que podem estar presentes em câmeras digitais, câmeras de dispositivos móveis e *scanners*. A partir daí, a versão digital do documento é gerada, o objeto básico de estudo de área de Processamento Digital de Imagens (PDI).

1.1 Processamento Digital de Imagens

O principal objetivo desta área é melhorar a percepção de informações visuais para humanos ou máquinas. Além disto, é utilizada para otimizar a transmissão de informação através de técnicas de compactação. No entanto, não existe um consenso entre os autores em relação onde o processamento de imagens termina e outras áreas, como visão computacional, começam (**?**).

Para Gonzalez (**?**), a melhor forma de organizar esta área é em três níveis de processo: baixo, médio e alto. Os processos de baixo nível contemplam as operações onde a entrada e saída são imagens, como por exemplo limiarização que é transformação de imagem para apenas dois níveis de intensidade. Os de médio nível envolvem a segmentação, ou seja, o resultado são subimagens, como componentes ou bordas. Por fim, os processos do nível alto têm o objetivo de dar sentido à imagem, como classificação de texto.

A partir do surgimento do armazenamento de imagens de documentos em grande escala, uma nova área de estudo surgiu em PDI: análise de processamento de documentos digitais (**?**). Remover ruídos, separar figuras de textos e reconhecer os textos na imagem estão entre os objetivos desta área. Desta forma, é possível definir o macro fluxo de Processamento de Imagens de Documentos, visando reconhecimento, pelo fluxograma da Figura 1.1.

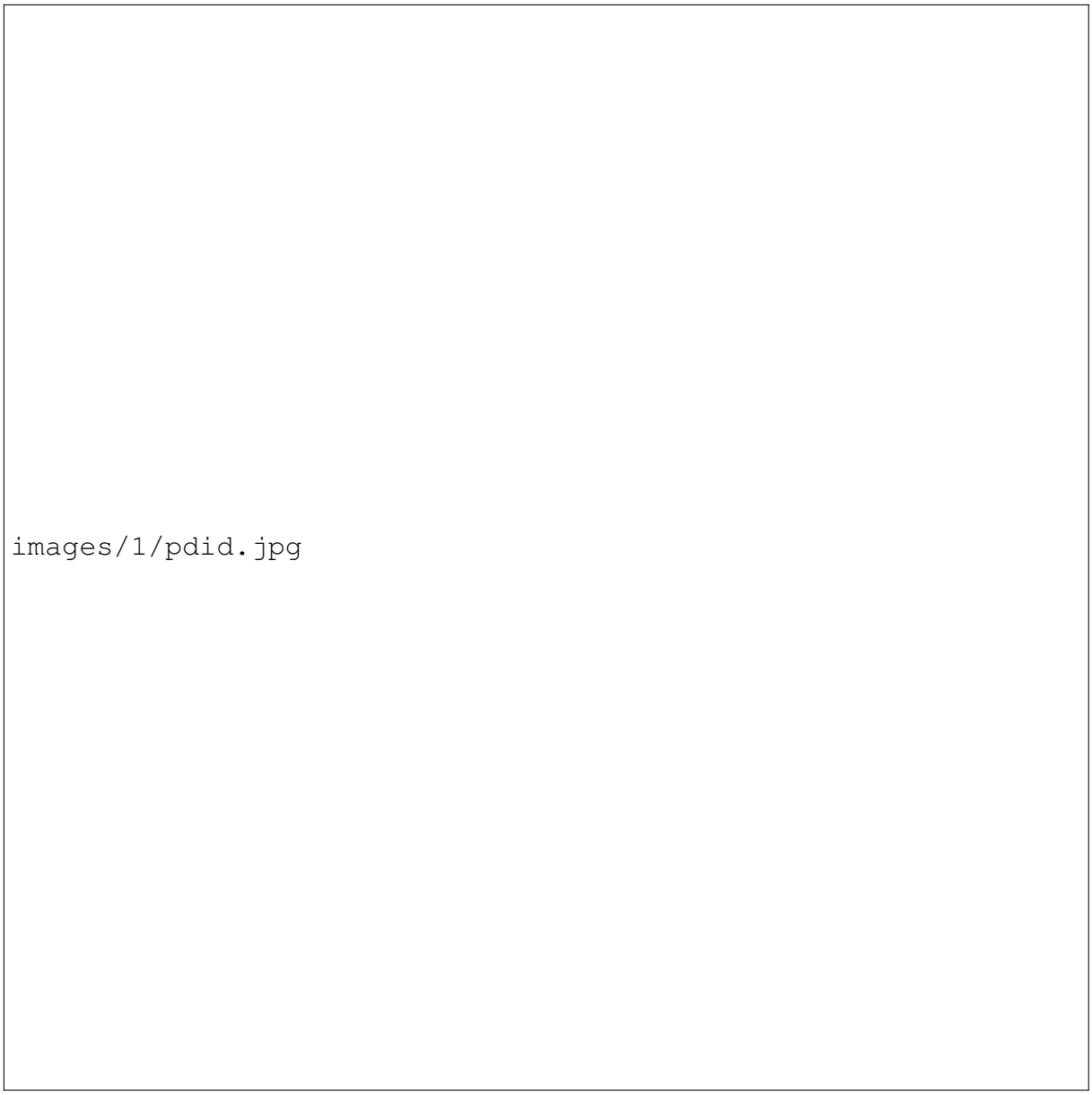


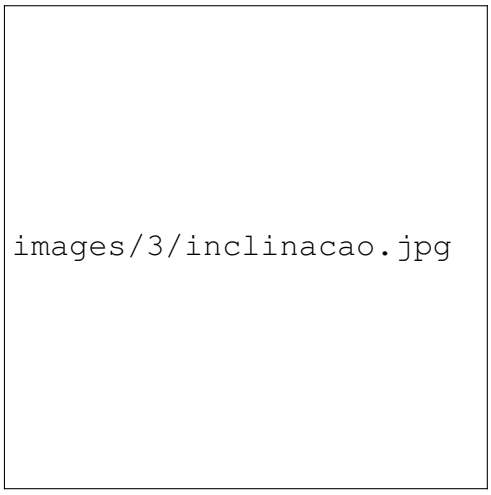
Figura 1.1: Fluxograma de Processamento de Imagens de Documentos

A *binarização* ou limiarização é a transformação de uma imagem em outra com apenas dois níveis de intensidade. Normalmente, os dois tons são preto, representando os objetos, e o branco, para o *background*. Para Mello (**?**), algoritmos de binarização para imagens de documentos podem ser organizados em quatro classes: técnicas baseadas em histograma, métodos iterativos, algoritmos baseados em entropia e técnicas *fuzzy*.

A fase de *pré-processamento* comporta uma variedade de possibilidades que dependem da finalidade da solução e das características das imagens utilizadas. Em geral, as fases de Segmentação e Reconhecimento pressupõem que as imagens estejam limpas. Por isto, o pré-processamento tem o objetivo de melhorar a qualidade da imagem por meio da remoção de ruídos e correção de distorções que são entraves para o processamento adequado das fases subsequentes.

A Figura 1.2 ilustra distorções comuns encontradas em processamento de imagens de

documentos. Essas distorções são estreitamente ligadas a maneira da aquisição das imagens. No primeiro caso (Figura 1.2a), os limites do documento não estavam alinhados com a orientação da câmera no momento da captura, causando uma inclinação na representação do documento na imagem digital . No segundo caso (Figura 1.2b), além da inclinação, é possível perceber diferença de profundidade na imagem, causando o efeito de perspectiva. Estes fenômenos têm influência direta no desempenho da Segmentação e Reconhecimento, por isto, a importância de sua correção.



(a) Imagem com inclinação. Extraído de **?**).

(b) Imagem com distorção de perspectiva.

Figura 1.2: Imagens com distorções.

A *segmentação* é a separação dos objetos que compõem a cena. No processamento de imagens de documentos, a segmentação de documentos tem o objetivo de separar o conteúdo textual de outros elementos, como *background* e elementos gráficos sem informação de texto. Em muitos casos, faz-se uma análise de *layout* para identificar as regiões de interesse na imagem. Em seguida, as regiões de texto são novamente segmentadas para definir as linhas de texto e as palavras.

No *reconhecimento*, ocorre a identificação do texto através do OCR. Esta técnica consiste em transpor o conteúdo na imagem em informação textual, tal qual um humano faz durante a leitura. Ainda que em constante evolução, a capacidade de reconhecimento de texto automático ainda não é comparável a dos humanos para texto manuscrito, devido à grande variabilidade da escrita humana, e textos impressos com presença elevada de ruídos. Por isto, a qualidade da imagem nesta fase do processamento é crucial para obter bons resultados de reconhecimento.

1.2 Motivação

Como discutido no início deste capítulo, mesmo com grande avanço tecnológico em mídias digitais, o papel continua presente na rotina das pessoas. Um exemplo disto é a necessidade de guardar documentos tais quais: comprovantes de pagamentos, contratos e documentos pessoais. Da mesma forma, em ambientes corporativos, devido à grande quantidade de papel, foi necessário criar um setor para a gestão de documentos. Em ambos os casos, existem soluções que fornecem facilidades para organizar o conteúdo documental, tanto para pessoas físicas como para empresas.

Softwares como *Evernote* (**?**), oferecem solução de gestão de documentos pessoais com OCR. Para empresas, existem sistemas de Gestão Eletrônica de Documentos (GED) destinados principalmente a diminuir o espaço físico na guarda de papéis e recuperação de informação a partir de suas versões digitais. No Brasil, uma solução de GED disponível é o DSDOCs (**?**).

Em todos os contextos supracitados, o *scanner* tem sido o dispositivo de aquisição dominante. Existem diversos tipos de *scanners* disponíveis no mercado: desde os industriais capazes de realizar centenas de digitalizações por minuto até os de uso casual para captura de cartões de visita, por exemplo. Além disto, dependendo do calibre do *scanner*, o usuário tem acesso a um grande alcance de resolução e profundidade de cor da imagem resultante.

Porém, existem situações onde o uso do *scanner* tradicional não é eficaz. São dispositivos grandes e, normalmente, precisam estar em uma base fixa e ligados a energia. Por isto, não oferecem portabilidade e mesmo *scanners* portáteis tem a necessidade de estarem conectados a um computador, o que também não é uma solução ótima.

Outra restrição do uso do *scanner* é na digitalização de documentos históricos, utilizada como forma de preservação de informação e da herança cultural (**?**). Estes tipos de documentos são frágeis e precisam de um cuidado especial para manuseio, o que torna o uso de *scanners* de mesa tradicionais bastante arriscado, pois podem danificar o documento de maneira irreversível. Também não é possível a leitura automática de textos em cenas naturais através de *scanners*.

Em grandes empresas, como bancos e lojas, vários serviços exigem um cadastro prévio que precisam guardar versões digitalizadas de documentos do cliente como: identificação, comprovante de residência e comprovante de renda. Por este motivo, estes cadastros são feitos quase sempre dentro das dependências da empresa, como agências ou pontos de venda. De maneira semelhante, existem circunstâncias onde o documento original não pode ser retirado do local de guarda. Isto ocorre em tribunais onde os documentos originais de processos não podem sair do local ou mesmo em bibliotecas onde o livro só pode ser utilizado para consulta.

Estes são exemplos onde as soluções tradicionais utilizando *scanners* são ineficazes ou impossíveis de serem utilizadas. Por este motivo, a utilização de dispositivos móveis munidos de câmeras para captura de imagens de documento tem se mostrado uma alternativa interessante aos tradicionais *scanners* de mesa. Isto se deve a sua facilidade de uso, portabilidade, a não necessidade de contato com o objeto e barateamento de seu hardware que facilitou sua popularização.

Em janeiro de 2015, no Brasil, existiam 281,7 milhões de celulares onde mais da metade eram munidos de câmera (**?**).

Porém, por se tratar de captura em um ambiente menos controlado, documentos digitalizados desta forma apresentam distorções que comprometem a sua legibilidade tanto por humanos quanto por aplicações baseadas em OCR. Ambientes menos controlados trazem alguns fenômenos no resultado da digitalização: distorção de perspectiva, inclinação, baixa resolução, interação do conteúdo com o background e curvatura das linhas de texto. Os algoritmos tradicionais de Processamento de Imagem fornecem uma boa base para trabalhar com este tipo de problema e muito se tem evoluído neste sentido (**???**).

Entretanto, as soluções disponíveis para corrigir documentos capturados por dispositivos móveis são bastante dependentes do conteúdo textual. Em geral, as imagens de teste das técnicas do estado da arte são bastante limpas e com resolução alta. Além disto, é conveniente que a correção seja eficiente em tempo e custo computacional porque deve ser executada embarcada no dispositivo.

1.3 Objetivos

O principal objetivo deste trabalho é o desenvolvimento de uma abordagem para corrigir distorções de perspectiva causadas pela captura de documentos por dispositivos móveis sem dependência do conteúdo textual. Desta forma, os objetivos específicos são:

- Contribuir com estado da arte para um problema de difícil solução;
- Construção de uma base de documentos pessoais brasileiros;
- Propor solução eficiente para correção de distorção de perspectiva; e
- Comparação da técnica proposta com técnicas do estado da arte.

1.4 Estrutura da Dissertação

Este trabalho está organizado em seis capítulos. No Capítulo 2, os conceitos básicos para o entendimento deste trabalho são detalhados. No Capítulo 3, é realizada uma revisão do estado da arte de correção de imagens capturadas por câmera e algumas considerações estão presentes. O Capítulo 4 detalha o método proposto por este trabalho. A metodologia dos experimentos e discussão acerca dos resultados estão no Capítulo 5. Por fim, o Capítulo 6 traz as conclusões finais, as contribuições e os trabalhos futuros.

2

Conceitos Básicos

Este capítulo visa a explicar os principais conceitos e técnicas utilizados para alcançar os objetivos deste trabalho. As transformadas de Hough e o descritor Hog, alicerces do método proposto, são detalhadas nas Seções 2.1 e 2.2 respectivamente. A Seção 2.3 é destinada a apresentar as distorções abordadas neste trabalho.

2.1 Descritor de HOG

O Histograma de Gradientes Orientados (HOG) foi inicialmente descrito por Navneet Dalal e Bill Triggs para detecção de pedestres (**?**). O descritor parte do princípio de que o formato e aparência de um objeto em uma imagem podem ser bem descritos pela distribuição local dos gradientes ou pela direção de sua borda. A técnica pode ser dividida em quatro etapas como representado na Figura 4.5.

O primeiro passo é aplicar uma correção Gama (**?**) em cada canal da imagem, caso ela seja colorida. Dalal e Bill Triggs determinaram experimentalmente que esta correção causou uma modesta melhora nos testes de detecção de pedestres. Depois, as derivadas de primeira ordem para as componentes vertical I_x e horizontal I_y são calculadas pelas convoluções, representada pelo símbolo $*$, nas Equações. 2.1 e 2.2. Onde D_x e D_y são as máscaras abaixo

$$D_x = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

$$D_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

Com isto é possível calcular a magnitude (Eq. 2.3) e a direção do gradiente (Eq. 2.4).

$$I_x = I * D_x \tag{2.1}$$

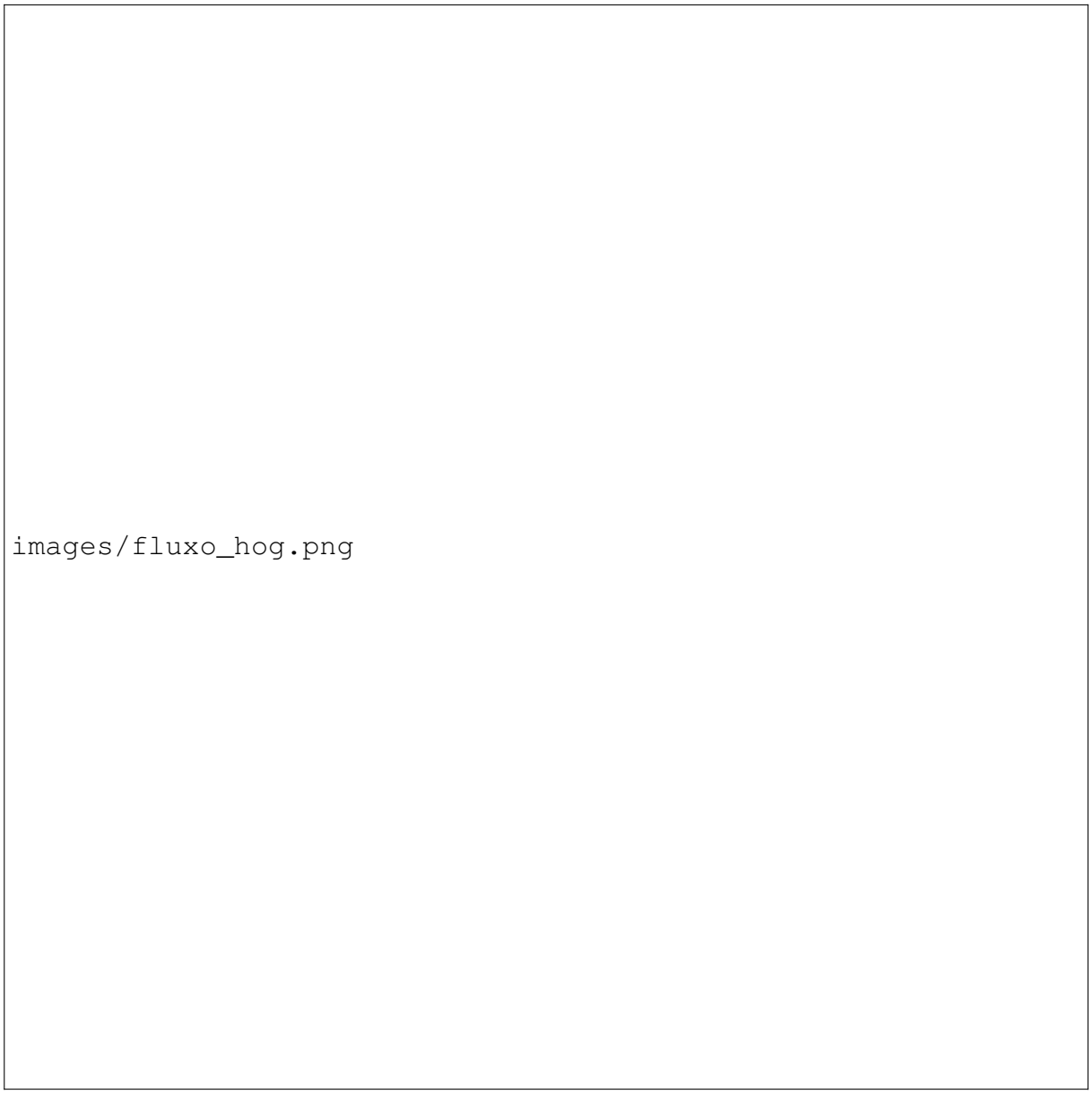


Figura 2.1: Cálculo descritor HOG.

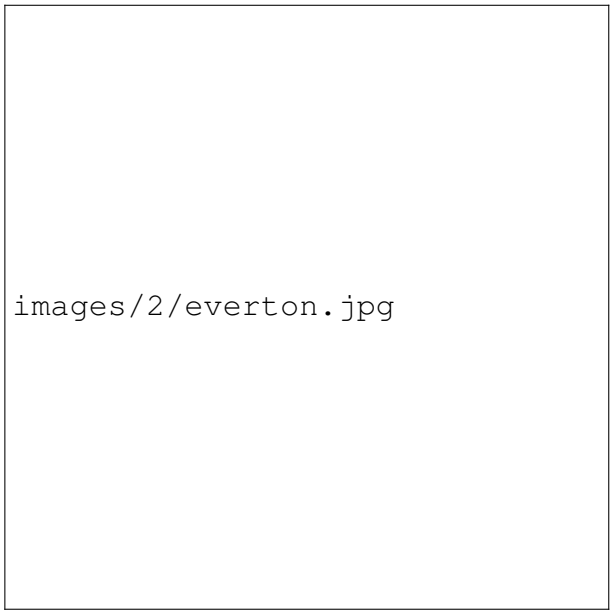
$$I_y = I * D_y \tag{2.2}$$

$$|G| = \sqrt{I_x^2 + I_y^2} \tag{2.3}$$

$$\theta = arctan\frac{I_y}{I_x} \tag{2.4}$$

Depois do cálculo da magnitude e da direção do gradiente, a imagem é dividida em células de tamanho *cellSize*, que é medido em *pixels* (Figura 2.2). Para cada uma das células (Figura 2.2b), um histograma é calculado. O valor da magnitude de cada pixel na imagem representa um peso para a orientação do próprio pixel; a este valor dá-se o nome de voto. Os

votos são acumulados nas suas respectivas posições no histograma, como ilustra a Figura 2.3. No seu trabalho, Dalal e Triggs determinaram experimentalmente que o histograma deveria ser igualmente espaçado no *range* 0º–180°, a quantidade de divisões do Histograma é dado pelo parâmetro *numBins*.



(a) Imagem Original.

(b) Imagem dividida por células de tamanho *cellSize*

Figura 2.2: Divisão da imagem em *cellSize*

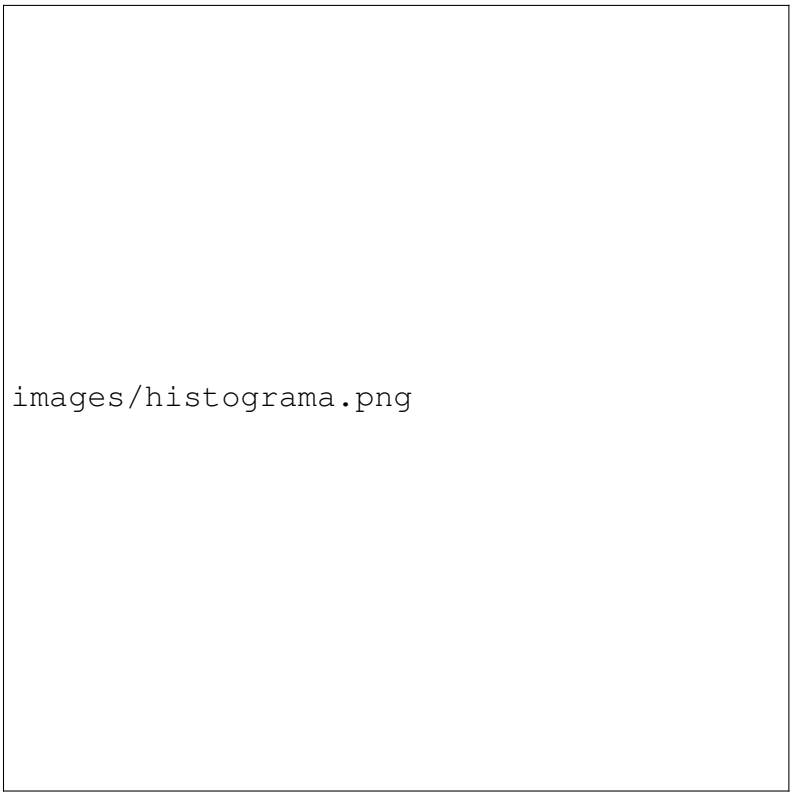
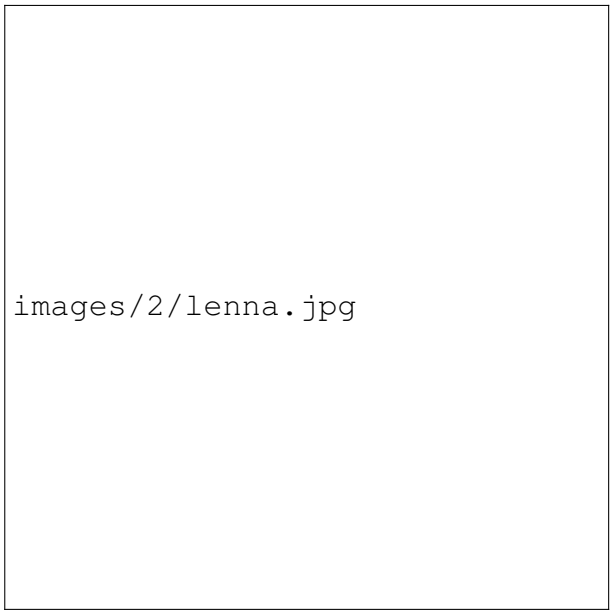


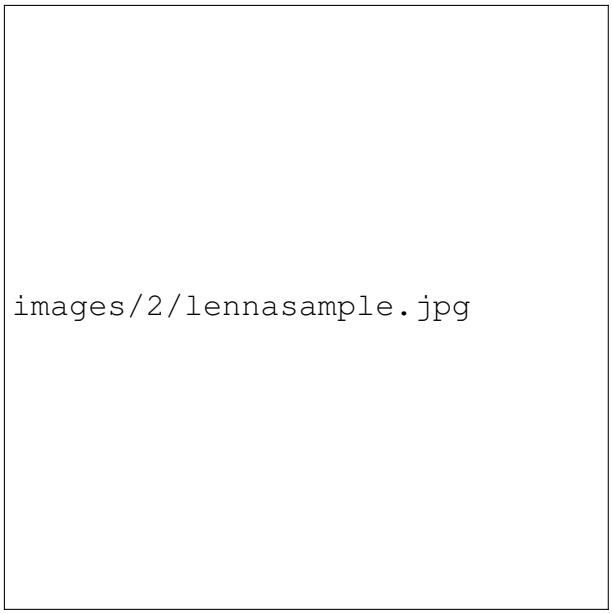
Figura 2.3: Construção do Histograma onde *numBins* = 9

A fim de mitigar variações de iluminação e sombra, uma normalização é realizada por agrupamentos de células, chamados blocos. A normalização é realizada deslizando uma janela bloco a bloco. Esta janela caminha de uma forma que haja sobreposição para que os valores das células possam contribuir na normalização de todos os seus vizinhos. A Figura 2.4 traz uma abstração visual dos histogramas de gradientes orientados de uma imagem.



(a) Imagem original

(b) HOGs da Imagem



(c) Amostra da imagem original

(d) Hogs da amostra

Figura 2.4: Exemplo visual dos histogramas da imagem.

2.2 Transformada de Hough

A Transformada de Hough (HT) foi proposta por Paul Hough para identificação de trajetórias de partículas em uma Câmara de Bolhas. Foi observado que estas trajetórias tinham comportamento uniforme e linear. Hough, então, desenvolveu uma maneira de identificar e isolar estas linhas retas em uma imagem digital (**?**). Mais tarde, Richard Duda e Peter Hart estenderam a HT para identificação de objetos com formas arbitrárias, não apenas de linhas retas, e a chamou de Transformada de Hough Generalizada (GHT) (**?**). Uma das contribuições deste trabalho é uma versão modificada da HT que utiliza uma imagem no espaço de HOG como entrada para identificação de linhas retas. Esta abordagem está detalhada na Seção 4.2. Por isto, esta seção se atem à utilização da HT na identificação de linhas retas em imagens digitais.

A HT é uma técnica de extração de características baseada em um processo de votação. Podemos dividir este processo em duas fases: (1) transformar uma imagem no plano cartesiano $I(x,y)$ para uma representação em um espaço de parâmetros $H(\rho,\theta)$ e (2) filtrar parâmetros de linhas retas mais representativas. A transformação de domínio é fundamentada na ideia de que uma linha reta, representada por uma função afim $y = \alpha x + \beta$, pode ser convertida em coordenadas polares. A Figura 2.5 ilustra esta relação formalizada pela Equação 2.5. Um rearranjo para forma $\rho = f(\theta)$ é mostrado na Equação 2.6.

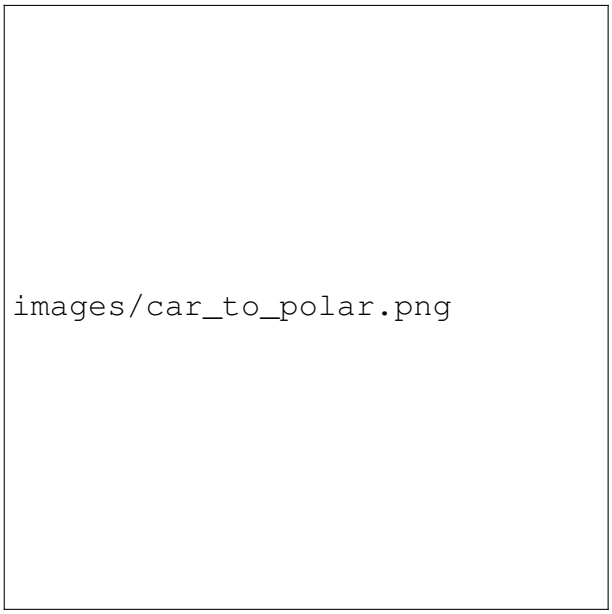


Figura 2.5: Relação cartesiano-polar

$$y = -\frac{cos\theta}{sen\theta}x + \frac{\rho}{sen\theta} \tag{2.5}$$

$$\rho = xcos\theta + ysen\theta \tag{2.6}$$

Portanto, cada par de linhas retas (α, β) tem seu respectivo par polar (θ, ρ), onde ρ é a distância entre a origem e a reta e θ é o ângulo complementar a inclinação da reta. A HT baseia-se neste princípio para construir o espaço de parâmetros $H(\theta, \rho)$, onde $\theta \in [0, \pi]$ e $\rho \in \mathbb{R}$. A transformação é realizada por uma varredura em $I(x, y)$: para cada pixel ativo $I(x_i, y_i)$ todos os pares (θ, ρ) de todas as possíveis retas que cortam o plano cartesiano em (x_i, y_i) são incrementados por 1 (um) em $H(\theta, \rho)$, como é mostrado pelo Algoritmo 1.

O resultado deste processo é a matriz acumulada $H(\theta, \rho)$. As figuras 2.6 e 2.7 mostram a Imagem de uma linha reta e a representação da HT respectivamente. A partir de $H(\theta, \rho)$ é possível ver o ponto máximo na imagem θ =-45º e o ρ = 0. Utilizando as fórmulas 2.7 e 2.8, colocando no formato de função afim $y = \alpha x + \beta$, obtemos a seguinte reta: $y = x$.

Algoritmo 1 Transformada de Hough

```
1:  procedure HOUGHTRANSFORM(I)                          ▷ I é uma imagem binária
2:      cols ← Im.cols
3:      rows ← Im.rows
4:      colsHough ← 2 * sqrt(cols^2 + rows^2)            ▷ O máximo ρ é a diagonal de I
5:      rowsHough ← 180                                  ▷ [0, π] em graus
6:      H ← CreateImage(rowsHough, colsHough)
7:      for i ← 1, rows do
8:          for j ← 1, cols do
9:              if IsObjectPixel(Im(i, j)) then          ▷ Avalia apenas pixels ativos
10:                 for k ← 0, rowsHough − 1 do
11:                     θ ← k * (π/180)
12:                     ρ ← i * cos(θ) + j * sin(θ)
13:                     H(k, ρ) ← H(k, ρ) + 1            ▷ Incremento de todas as retas
14:                 end for
15:             end if
16:         end for
17:     end for
18: end procedure
```

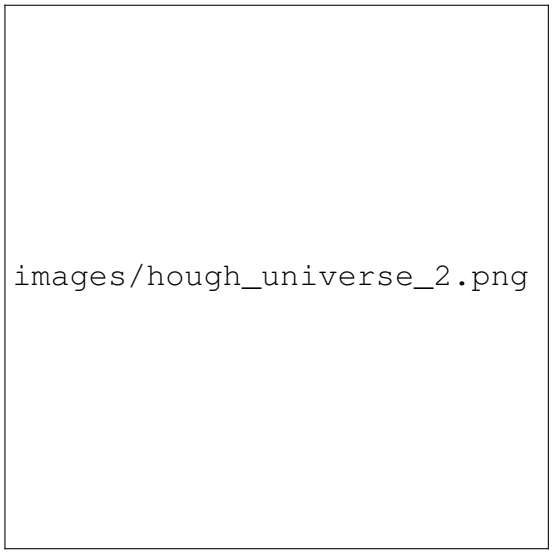


Figura 2.6: Linha reta no espaço cartesiano

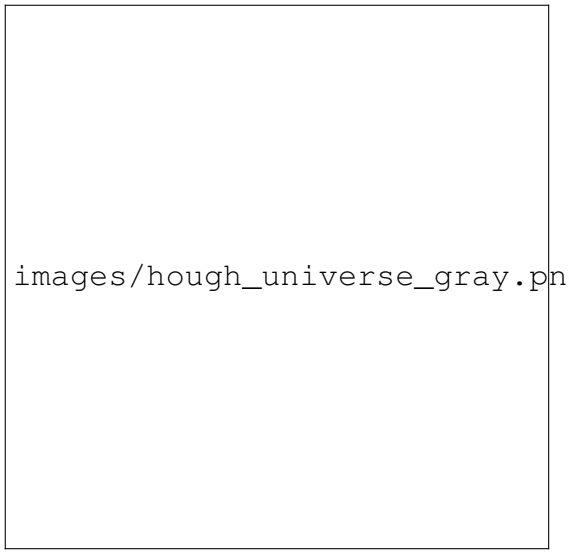


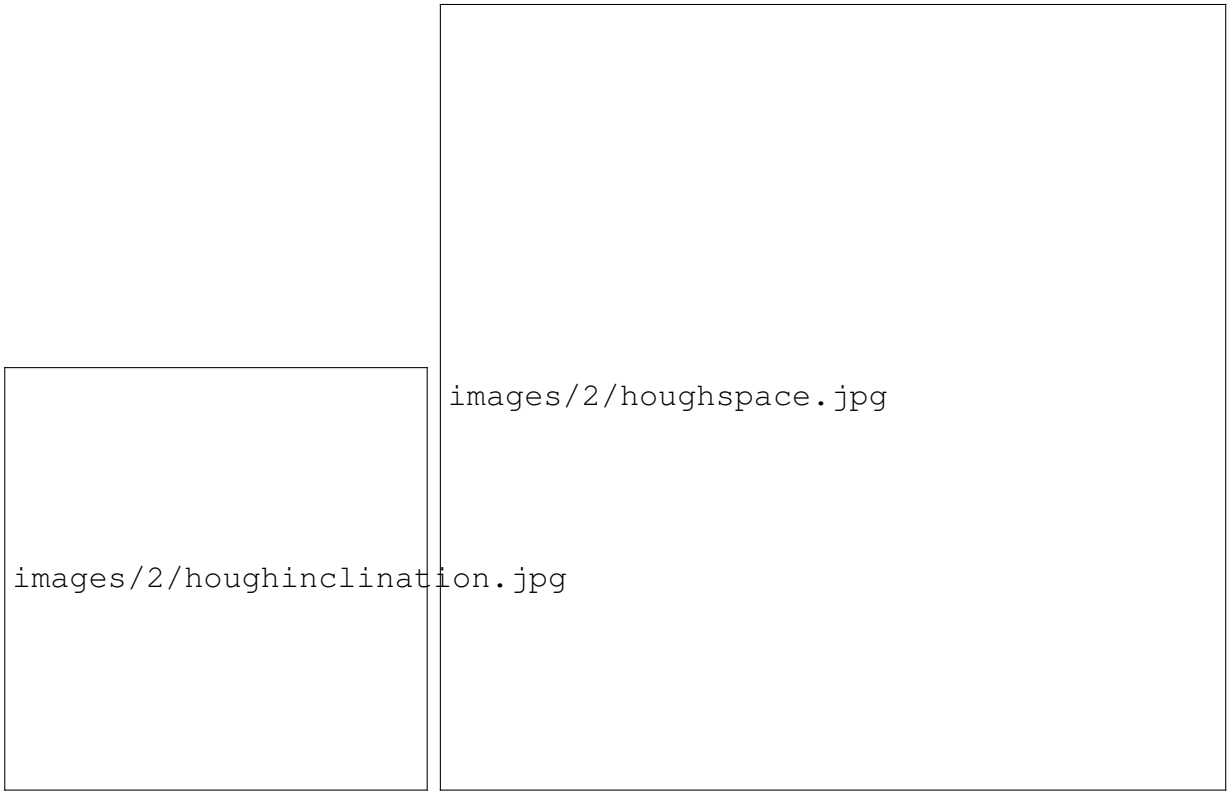
Figura 2.7: Representação no espaço de Hough

$$\alpha = \begin{cases} 0 & Se\theta = 90° \\ -\frac{sin\theta}{cos\theta} & \text{caso contrário} \end{cases} \tag{2.7}$$

$$\beta = \frac{\rho}{sin\theta} \tag{2.8}$$

A HT também pode ser aplicada para detecção de inclinação. Por exemplo, a Figura 2.8a mostra a imagem de um documento com 10º de inclinação no sentido anti-horário. Após aplicar a HT, obtém-se o respectivo espaço de Hough, ilustrado pela Figura 2.8b. É possível observar

vários pontos de convergência para $\theta = 80$, que é o complemento da inclinação do documento. A partir deste valor, é possível aplicar um algoritmo de rotação na imagem, corrigindo esta distorção.



(a) Imagem de um documento com a de 10º inclinação

(b) Representação no espaço de Hough da imagem inclinada

Figura 2.8: Correção de Inclinação usando Transformada de Hough.

2.3 Distorções

A utilização de câmeras facilita o processo de aquisição de imagens, entretanto, a adoção destes dispositivos traz alguns desafios no momento da captura. Isto se deve a maneira de uso destes dispositivos que dependem da habilidade manual do operador. Ao contrário do que ocorre na captura por *scanner* de mesa, o enquadramento e a distância focal são calibrados pelo manuseio do dispositivo pelo operador, no momento da captura.

Por isto, a captura por dispositivos móveis pode gerar imagens com distorções. Nesta seção, as principais distorções tratadas nesta Dissertação são analisadas: Inclinação (Seção 2.3.1) e Perspectiva (Seção 2.3.2).

2.3.1 Inclinação

A inclinação é um problema frequente e, em geral, é adquirida durante a aquisição, quando o documento é posicionado com um ângulo diferente de zero grau sobre o eixo do

scanner (**?**). A correção de inclinação cumpre papel crucial em várias aplicações de Visão Computacional. Uma pequena inclinação no documento prejudica a análise da diagramação e, por consequência, todos os processos posteriores, como segmentação e classificação . A Figura 2.9 mostra um exemplo de documento digitalizado com inclinação.

(a) Imagem com 8º de inclinação. **(b)** Imagem corrigida.

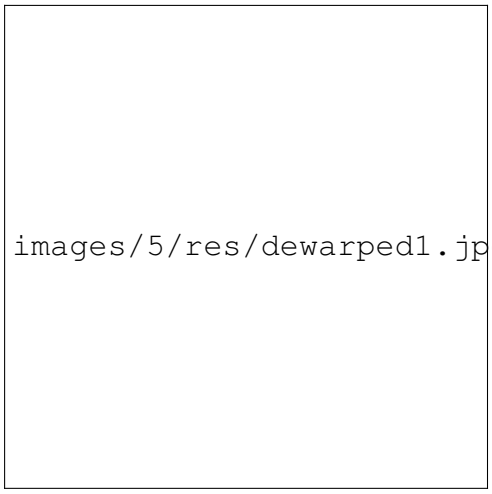
Figura 2.9: Exemplo de imagem inclinada e corrigida. Extraído de **?**).

A solução para este tipo de distorção é estimar o ângulo da inclinação dos documentos. Dentre as técnicas conhecidas para este problema estão: técnicas baseadas em análise de projeção (**?**); correção por HT (**?**) e solução baseada em agrupamento de vizinhos mais próximos (**??**).

2.3.2 Perspectiva

Documentos que não são posicionados no mesmo plano na qual a lente da câmera está no momento da captura, irão sofrer deste tipo de distorção. Mesmo que o documento esteja sobre uma superfície plana, posicionar a câmera paralelamente ao plano do documento é uma responsabilidade do operador do dispositivo. O resultado disto é que, na imagem digitalizada, algumas propriedades geométricas são preservadas, como colinearidade: uma linha reta permanece sendo uma linha reta após a captura. Por outro lado, linhas paralelas, em geral, não preservam esta propriedade (**?**). A Figura 2.10 ilustra esta distorção.

(a) Imagem Original.



(b) Imagem Corrigida.

Figura 2.10: Imagem com distorção de perspectiva (a) e sua correção (b).

No Capítulo 3, as abordagens do estado da arte de correção de perspectiva são discutidas. Parte destas técnicas, inclusive a proposta neste trabalho, se baseia na determinação de pontos específicos no universo da imagem para mapear a superfície distorcida à superfície corrigida. A este mapeamento dá-se o nome de *Dewarping* ou Retificação.

Dewarping, pode ser definido como a projeção da imagem, segundo seu próprio feixe perspectivo, para um plano horizontal. Na prática, o processo de retificação é um mapeamento do pixel no plano da imagem (X,Y) ao pixel (x,y) no plano real 2D, como pode ser visualizado na Figura 2.11. Esta transformação projetiva é uma transformação linear que utiliza uma matriz 3x3 não-singular (Equação 2.9).

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{2.9}$$

A Equação 2.9 pode ser representada pela Equação 2.10, onde λ é a matriz da transformação, e X e x são as coordenadas original e corrigida, respectivamente. Esta equação pode ser resolvida utilizando a pseudo-inversa para resolver os sistemas de equações lineares. Assim, podemos expressar λ pela Equação 2.12. Com λ, é possível mapear os pixels na superfície distorcida à superfície corrigida, como mostra a Figura 2.10b.

$$X\lambda = x \tag{2.10}$$

$$X^T X\lambda = X^T x \tag{2.11}$$

$$\lambda = (X^T X)^{-1} X^T x \tag{2.12}$$

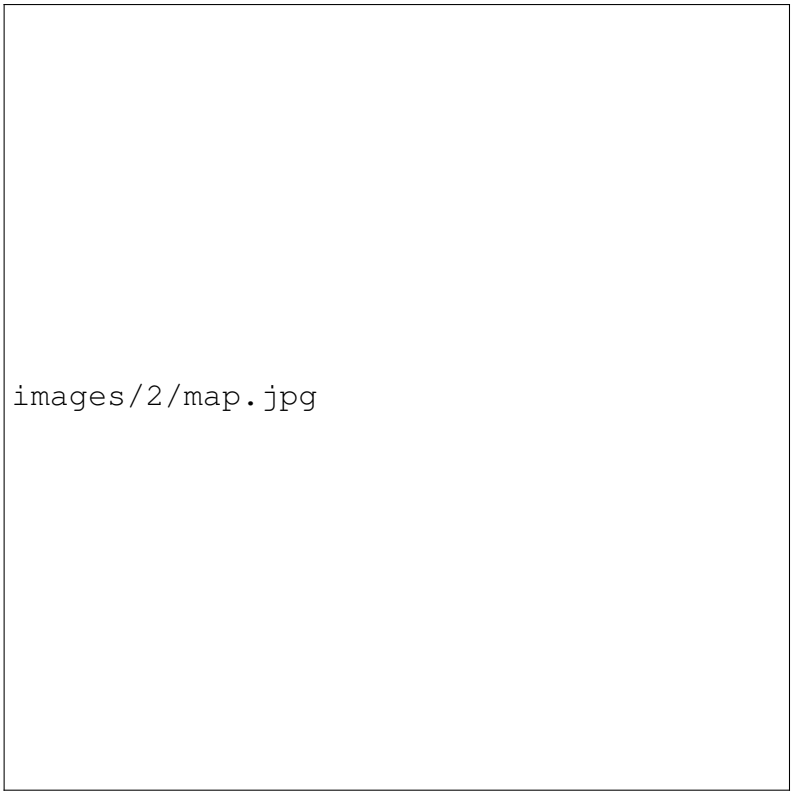


Figura 2.11: Mapeamento de pixel do plano da imagem ao plano 2D real. Modificado de (**?**).

3

Correção de Imagens Capturadas por Câmera

O objetivo deste capítulo é apresentar os principais conceitos e desafios na área de correção de imagens capturadas por câmera além de seu estado da arte. Os trabalhos propostos até então (**?**) podem ser divididos em dois tipos de abordagem: correção baseada na reconstrução de modelos 3D (Seção 3.1) e processamento de imagens 2D (Seção 3.2). O presente trabalho se enquadra no segundo tipo de abordagem. Desta forma, três propostas são detalhadas: a primeira baseada em retificação por *Vanishing Points* (Seção 3.2.1), a segunda utiliza contornos adaptativos (*snakes*) para segmentar e corrigir o texto encontrado (Seção 3.2.2) e, por último, uma abordagem baseada na estimativa da superfície distorcida a partir das linhas de texto encontradas na imagem (Seção 3.2.3).

3.1 Correção baseada em reconstrução do modelo 3D

Os trabalhos desta categoria dependem da obtenção de informações 3D da imagem. Estas informações podem ser obtidas durante a captura, utilizando dispositivos específicos. Um exemplo são as câmeras estereoscópicas, dotadas de duas ou mais lentes com sensores de imagem próprios, capazes de simular a visão binocular humana e capturar imagens tridimensionais. Ulges *et al.* (**?**) fizeram uso deste dispositivo, em **?**) são utilizados *scanners* à *laser*, enquanto **?**) utilizaram técnicas de luzes estruturadas.

Outra abordagem é a construção do modelo 3D a partir de informações presentes na imagem. Por exemplo, Cao *et al.* (**?**) propuseram uma abordagem baseada em um modelo cilíndrico, assumindo a visão frontal de um livro aberto. Liang *et al.* (**?**) estimaram o formato 3D do documento a partir das texturas presentes na imagem. Enquanto, Tan *et al.* (**?**) usaram as informações de sombra presentes na imagem para reconstruir o modelo tridimensional do documento.

A obrigatoriedade de utilizar equipamentos específicos para realizar a correção das imagens implica em perda de portabilidade e, muitas vezes, ter um controle maior sobre o

ambiente. Da mesma forma, as soluções baseadas em construção do modelo 3D realizam pressuposições que tornam as propostas vulneráveis: o método apresentado em (**?**) assume que a superfície distorcida é cilíndrica; já o algoritmo introduzido em (**?**) funciona apenas para curvaturas suaves; conhecimento sobre a luz no momento da captura é requerido no método descrito em (**?**).

3.2 Correção baseada em 2D

As soluções desta categoria se baseiam exclusivamente em informações presentes na imagem para realizar a correção. A maioria destas propostas é baseada na detecção das linhas distorcidas (**?**). Porém, existem soluções que realizam o mapeamento da superfície distorcida para uma superfície retificada com base em informações das bordas do documento ou pontos de referências, como *Vanishing Points* ou os cantos do documento.

Encontrar as linhas do texto em um documento distorcido não é uma tarefa fácil. Várias pesquisas baseiam-se na representação das linhas de textos não lineares através de *splines* ou aproximações polinomiais (**?????**). Estas técnicas sofrem de várias limitações como espaçamento entre linhas heterogêneo, alto custo de processamento e pressuposições que não tem respaldo em situações reais.

A seguir, três abordagens distintas do estado-da-arte são apresentadas. A primeira investe na busca por *Vanishing Points* a partir de projeções inclinadas e características extraídas das letras do texto. Na seção seguinte, algoritmos baseados em *snakes* são utilizados para representar a linhas distorcidas do texto do documento. Por fim, uma abordagem baseada em dois passos, um grosseiro e outro ajuste fino, capaz de descobrir a superfície distorcida que contém o texto em uma superfície corrigida.

3.2.1 *Fujimoto*

A abordagem de **?**) é baseado na correção de deformação de perspectiva a partir dos *Vanishing Points* [1] da imagem. Um *Vanishing Point* (VP) é o ponto de encontro das projeções de uma imagem e pode ser utilizado para realizar a correção de perspectiva, desde que se tenham os VPs vertical e horizontal. Fujimoto propõe uma abordagem híbrida para realizar esta tarefa que pode ser dividida em duas partes: uma direta e outra indireta.

A abordagem direta utiliza análise das projeções da imagem para determinar os VPs. Este método é bastante preciso, porém, tem um custo computacional alto. Em oposição, a abordagem indireta é bastante eficiente computacionalmente, mas menos assertivo. Esta última, baseia-se em heurísticas para determinar qual VP é o correto dentre todos os VPs conhecidos.

O Fluxograma da Figura 3.1 mostra os passos da abordagem de Fujimoto. O primeiro passo é realizar o pré-processamento da imagem: (1) transformação da imagem para tons de

[1] Em português do Brasil, "Pontos de Fuga"

cinza; (2) limiarização e (3) extração das bordas. Depois, a partir de heurísticas aplicadas a componentes conectados, as linhas retas, linhas de base e componentes verticais dos caracteres são identificados. A partir destas informações, a abordagem híbrida agrupa e detecta os VPs horizontais e verticais.

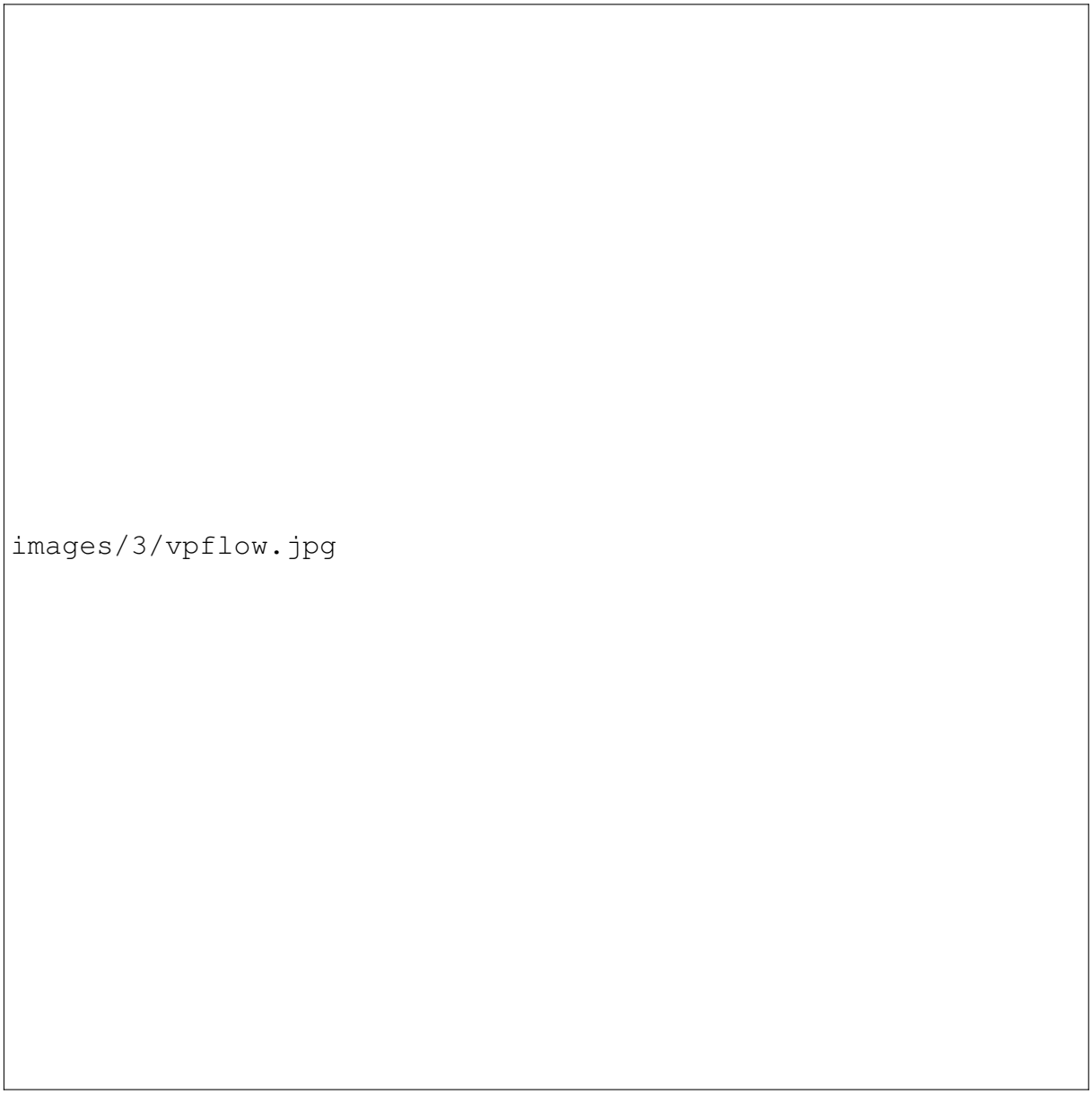


Figura 3.1: Fluxograma da Solução de *Fujimoto*

Para determinação dos VPs supõe-se que as linhas estão orientadas horizontalmente. As linhas retas são calculadas a partir de uma heurística baseada na análise de Componentes Conectados e em uma análise estatística. Após detectar as bordas da imagem (**?**), os Componentes Conectados (CC) são agrupados nas direções horizontal e vertical, de acordo com tamanho e forma. Estes agrupamentos são candidatos a linha.

Para um CC C_i a sua linha correspondente LC_i é determinada utilizando o Método dos Mínimos Quadrados. Considerando LC_i uma linha representada da forma $ay + bx + c = 0$, a

distância de um ponto (x,y) em C_i à linha é dado pela Equação 3.1. A partir da Função 3.2, é possível determinar se LC_i é uma linha reta, caso $f(LC_i)$ resulte em 1. Para isto, é necessário o tamanho da linha $Len(LC_i)$ e uma distribuição gaussiana $N(x,\mu,\sigma)$ (Equações 3.3, 3.4 e 3.5). Os parâmetros μ_{line} e σ_{line} foram determinados experimentalmente a partir de imagens diferentes.

$$DIS_i(x,y) = \frac{|a_i y + b_i x + c|}{\sqrt{a^2+b^2}} \tag{3.1}$$

$$f(LC_i) = \begin{cases} 1 & Len(LC_i) > len\_thres, N_{LC_i} > n\_thres\_line \\ 0 & \text{caso contrário} \end{cases} \tag{3.2}$$

na qual *len_thres* é o tamanho mínimo que uma linha deve ter e *n_thres_line* é o mínimo valor que o somatório de todos os I_{LC_i} da linha deve ter.

$$P_{LC_i}(x,y) = N(DIS_i(x,y), \mu_{line}, \sigma_{line}) \tag{3.3}$$

$$I_{LC_i}(x,y) = \begin{cases} 1 & P_{LC_i}(x,y) > p\_thres\_line \\ 0 & \text{caso contrário} \end{cases} \tag{3.4}$$

na qual *p_thres_line* é o valor mínimo que $P_{LC_i}(x,y)$ deve ter para que (x,y) seja considerado um *pixel* válido.

$$N_{LC_i} = \sum_{(x,y)\in C_i} I_{LC_i}(x,y) \tag{3.5}$$

Outro grupo de linhas é calculado utilizando uma estratégia de borramento baseada em **?**). Este processo pode produzir blocos contendo mais de uma linha devido a distorção de perspectiva. Para resolver este problema, os contornos de cada linha borrada são calculados a partir das projeções verticais. Portanto, os contornos superior e inferior são expressos como $(x_1,y_1)^U,(x_2,y_2)^U,...,(x_N,y_N)^U$ e $(x_1,y_1)^L,(x_2,y_2)^L,...,(x_N,y_N)^L$, respectivamente.

A partir destes valores, a distância média entre os contornos superiores e inferiores, *contour_thres*, é calculada. Se uma distância for menor que um ponto de corte, *contour_thres*, este ponto é descartado. Os pontos remanescentes são descritos pelas Equações 3.6 e 3.7, onde M é o número de pontos remanescentes. Os pontos médios são representados pela Equação 3.8.

$$Set(U) = (x_1,y_1)^U,(x_2,y_2)^U,...,(x_M,y_M)^U \tag{3.6}$$

$$Set(L) = (x_1,y_1)^L,(x_2,y_2)^L,...,(x_M,y_M)^L \tag{3.7}$$

$$Set(C) = (x_1,\frac{y_1^U+y_1^L}{2}),(x_2,\frac{y_2^U+y_2^L}{2}),...,(x_M,\frac{y_M^U+y_M^L}{2}) \tag{3.8}$$

As linhas horizontais eleitas são as que respeitam as condições descritas nas Equações 3.9

(a) Imagem Original. **(b)** Bordas da Imagem. **(c)** Componentes alongados.

Figura 3.2: Extração dos componentes verticais. Extraído de **?**).

e 3.10. U e L representam as linhas de base superior e inferior, respectivamente, enquanto *cross_angle* calcula o ângulo formado pelas duas linhas e *ave_height* a distância média entre as linhas. Os limiares *angle_thres* e *height_thres* são determinados experimentalmente.

$$cross\_angle(U,L) < angle\_thres \tag{3.9}$$

$$ave\_height(U,L) < height\_thres \tag{3.10}$$

Fujimoto sugere que a inclinação dos caracteres fornece pistas suficientes para determinar o VP vertical. Baseado nisto, ele aplica o processo ilustrado na Figura 3.2. Primeiro, detecta as bordas da imagem (Figura 3.2b) e depois faz análise dos CC e seleciona os que apresentam características de alongamento vertical (Figura 3.2c).

Estes CC incluem componentes verticais, horizontais e inclinados. Assumindo que o ângulo de perspectiva na vertical não é maior que 45º, os componentes verticais são escolhidos se a altura do componente é muito maior que sua largura. De maneira similar a classificação das linhas retas, Fujimoto define se um CC é vertical pela Equação 3.11.

Na Equação 3.12, $N(x,\mu,\sigma)$ é uma distribuição Gaussiana para LC com média μ e desvio padrão σ, os valores μ_{stroke} e σ_{stroke} são a média e o desvio padrão da distribuição Gaussiana que determina se um componente é ou não alongado. Estes valores são determinados experimentalmente. Se $f(LC_i)$ for igual a 1, então C_i é um componente vertical. Para filtrar apenas CCs verticais e retilíneos, *p_thres_stroke* assume valores altos – próximos a 1, uma vez que $N(x,\mu,\sigma)$ tem contra-domínio $[0,1]$ – e *n_thres_stroke* tem valor próximo ao número de pixels do CC. Cada um dos componentes selecionados representa uma linha reta com orientação vertical que é utilizada para calcular o VP vertical.

$$f(LC_i) = \begin{cases} componente\ vertical & N_{LC_i} > n\_thres\_stroke \\ componente\ não\ vertical & \text{caso contrário} \end{cases} \tag{3.11}$$

na qual *n_thres_stroke* é o valor mínimo que N_{LC_i} deve obter para que a componente C_i seja

considerada vertical.

$$P_{LC_i}(x,y) = N(DIS_i(x,y), \mu_{stroke}, \sigma_{stroke}) \tag{3.12}$$

$$I_{LC_i}(x,y) = \begin{cases} 1 & P_{LC_i}(x,y) > p\_thres\_stroke \\ 0 & \text{caso contrário} \end{cases} \tag{3.13}$$

na qual p_thres_stroke é o valor mínimo que P_{LC_i} deve obter para que o pixel em questão seja considerado válido.

$$N_{LC_i} = \sum_{(x,y)\in C_i} I_{LC_i}(x,y) \tag{3.14}$$

O primeiro passo para determinar o VP horizontal é calcular as intersecções de todas as linhas encontradas. Estes pontos são agrupados utilizando o algoritmo *k-means*. A Equação 3.15 define o número de *clusters* utilizados, onde N_P é igual ao número de interseções. Cada *cluster* C tem um centro que tem peso definido pela Equação 3.16, onde N_i é o número de pontos do agrupamento i. Esta equação pode ser interpretada como uma função de custo para a abordagem indireta (Equação 3.17).

$$N_{cluster} = max([ln(N_P)], 10) \tag{3.15}$$

$$w_i^C = \frac{N_i}{\sum_{i=1}^{N_{cluster}} N_i} \tag{3.16}$$

$$f_{inderect}(x_i, y_i) = w_i^C(x_i, y_i) \tag{3.17}$$

Fujimoto refina sua proposta adicionando uma abordagem direta na análise dos VPs candidatos. Como mostrado em **?**), é possível calcular as projeções dos textos em imagens com distorção de perspectiva. As projeções a partir de um ponto $H(x,y)$ são armazenadas em uma estrutura $B(x,y)$, como mostra a Figura 3.3. A função de custo da abordagem direta pode ser definida como mostrado na Equação 3.18, onde f'_{direct} é a soma derivativa quadrática (Equação 3.19).

$$f_{direct}(c_i(x,y)) = \frac{f'_{direct} c_i(x,y)}{\sum_{i=1}^{N} f'_{direct} c_i(x,y)} \tag{3.18}$$

$$f'_{direct}(c_i(x,y)) = \sum_{j=1}^{N_B-1} (B_{j+1}(x,y) - B_j(x,y))^2 \tag{3.19}$$

A abordagem direta baseia-se no comportamento do resultado das projeções a partir de VPs válidos. Como mostra a Figura 3.4, quanto mais próximo ao VP real, mais bem comportado se apresenta a projeção em perspectiva. Fujimoto combina as duas abordagens numa equação

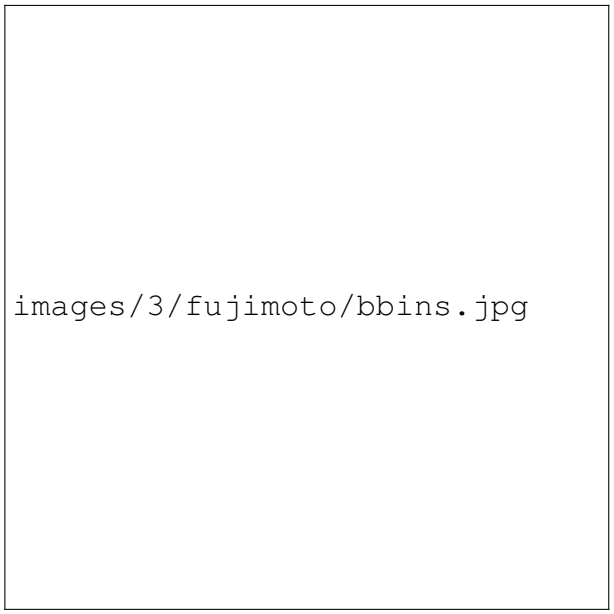


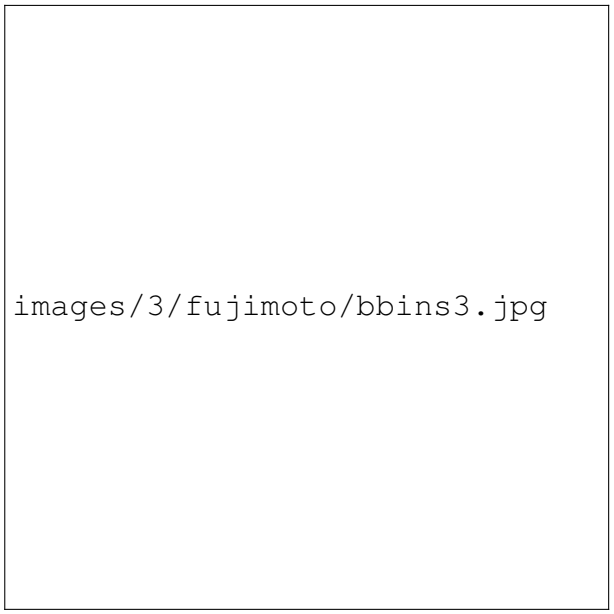
Figura 3.3: Cálculo de projeção $B(x,y)$ a partir do ponto $H(x,y)$. Extraído de **?**).

linear (Equação 3.20), onde foi definido experimentalmente que $\alpha = \beta = 0.5$. Por fim, o VP horizontal é dado pela Equação 3.21.

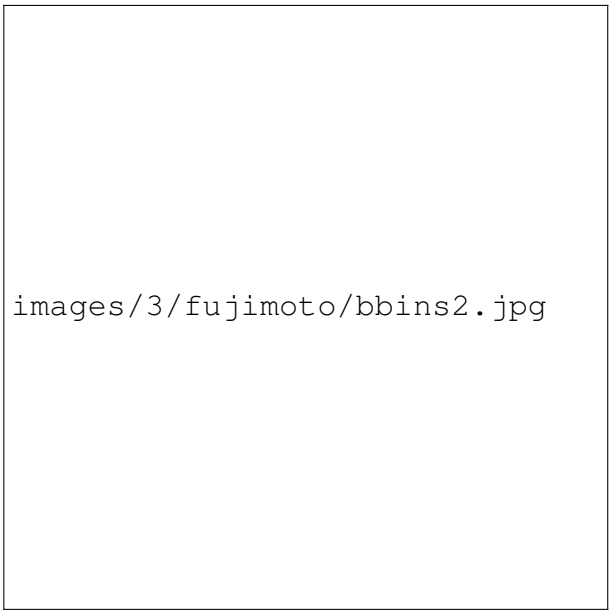
$$g(x_i, y_i) = \alpha f_{indirect}(x_i, y_i) + \beta f_{direct}(x_i, y_i) \tag{3.20}$$

$$(V_x, V_y) = argmax\ g(x_i, y_i) \tag{3.21}$$

O VP vertical é calculado de maneira similar. A projeção vertical é realizada linha-a-linha, já que não existe um padrão de espaçamento do texto na perspectiva vertical. Dado os VPs vertical e horizontal, a transformação da superfície distorcida para superfície corrigida é realizada como mostra a Figura 3.5. Resultados deste processo estão na Figura 3.6.

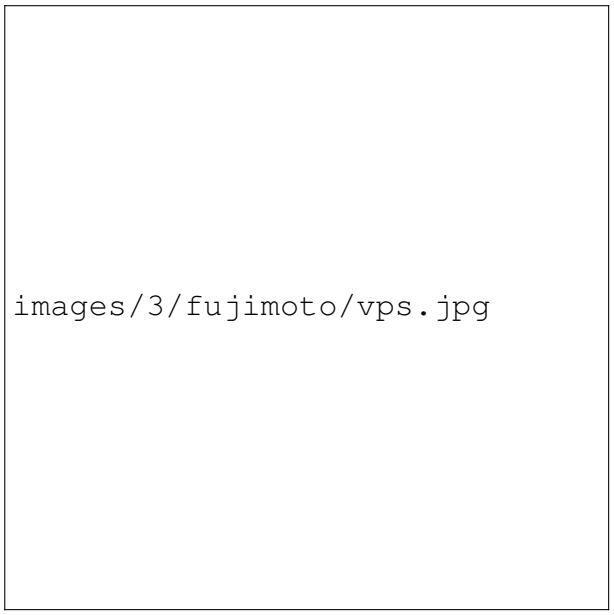


(a) Projeções em perspectiva em ângulo diferentes.



(b) Resultado da aplicação da projeção em perspectiva.

Figura 3.4: Projeções em perspectiva. Extraído de **?**).



(a) Determinação da superfície distorcida.

(b) Transformação da superfície distorcida para corrigida.

Figura 3.5: Correção da imagem. Modificado de **?**).

Figura 3.6: Resultados da abordagem de Fujimoto. Extraído de **?**).

3.2.2 *Bukhari et al.*

Bukhari *et al.* desenvolveram uma técnica de segmentação de linhas de texto distorcidas (**?**) baseados em contornos adaptativos – *snakes* (**?**). Eles adicionaram características ao modelo de *snakes* original e a chamaram de *coupled snakelets*. Algumas das melhorias introduzidas foram:

- ***Snake* com Curva Aberta**: um contorno adaptativo é representado, normalmente, por uma curva fechada. Para representar uma linha, foi necessário desenvolver uma *snake* com curva aberta.
- **Inicialização Automática das *Snakes***: para cada CC, um par de *snakes* é inicializado, uma para o topo e outra para a base.
- **Deformação Direcionada**: as *snakes* deformam apenas a componente vertical enquanto a componente horizontal se mantém estática. Esta característica é devido ao fato das escritas de origem latina serem horizontais.
- ***Snakes* Evolutivas**: cada *snake* é inicializada em um ponto do CC. Depois, iterativamente, ela adapta-se com relação ao *Gradient Vector Flow* (GVF) (**?**) dentro de um retângulo pequeno. Na próxima iteração, este retângulo aumenta um pouco e a nova

posição da *snake* é calculada. Este processo adaptativo auxilia ao contorno mover-se apenas sobre CCs que estejam contidos na mesma linha.

Os passos para a abordagem proposta por Bukhari *et al.* estão na Figura 3.7. Cada um desses passos está detalhado a seguir.

Figura 3.7: Fluxograma da Solução de Bukhari *et al.*

Pré-processamento: A imagem de entrada cinza é binarizada por um algoritmo adaptativo baseado em **?**). A imagem resultante pode conter ruído de sal-e-pimenta, por isto, uma heurística de limpeza é realizada na imagem. Se ao menos uma das condições nas Equações 3.22, 3.23 e 3.24 for verdadeira, o CC com altura H_{cc} e espessura W_{cc}, é descartado. H_{doc} e W_{doc} são a altura e a largura da imagem; H_{avg} e W_{avg} são a altura e espessura média dos CCs, enquanto σ_H e σ_W são os desvios padrão da altura e da espessura dos CCs.

$$H_{cc} > 0,1 \times H_{doc} \quad or \quad H_{cc} > 7 \times \sigma_H \tag{3.22}$$

$$W_{cc} > 0,1 \times W_{doc} \quad or \quad W_{cc} > 7 \times \sigma_W \tag{3.23}$$

$$H_{cc} \times W_{cc} < \frac{1}{3} \times H_{avg} \times W_{avg} \tag{3.24}$$

Inicialização das *Snakes*: Todos os CCs remanescentes após o pré-processamento são marcados como não processados. Então, um CC é escolhido aleatoriamente e duas *snakes* – topo e base – são inicializadas para o CC com tamanho L. Uma pequena região retangular de tamanho $W_R \times H_R$ é criada centralizada no CC de forma que cubra alguma parte dos vizinhos do componente. W_R, H_R e L são definidos pelas Equações 3.25, 3.26 e 3.27. Um exemplo de uma região retangular inicial pode ser encontrada na Figura 3.8a.

$$L = W_{cc} + 2 \times W \tag{3.25}$$

$$W_R = W_{cc} + 4 \times W \tag{3.26}$$

$$H_R = H_{cc} + 2 \times H \tag{3.27}$$

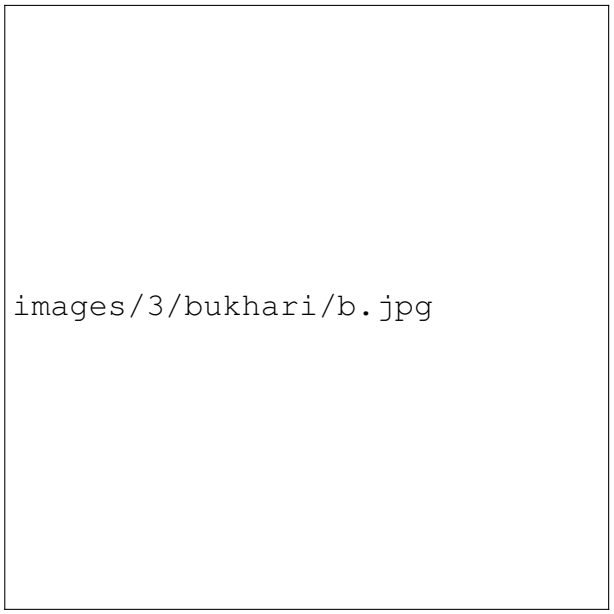
no qual W e H são espessura e altura média dos CCs respectivamente.

Deformação das *Snakes*: Para *snake* do topo, o GVF é calculado utilizando todos os topos de todos os CCs dentro da região. Então, esta *snake* é deformada usando os componentes verticais do GVF. Para a *snake* de base, o GVF é calculado utilizando os pontos de base de todos os CCs.

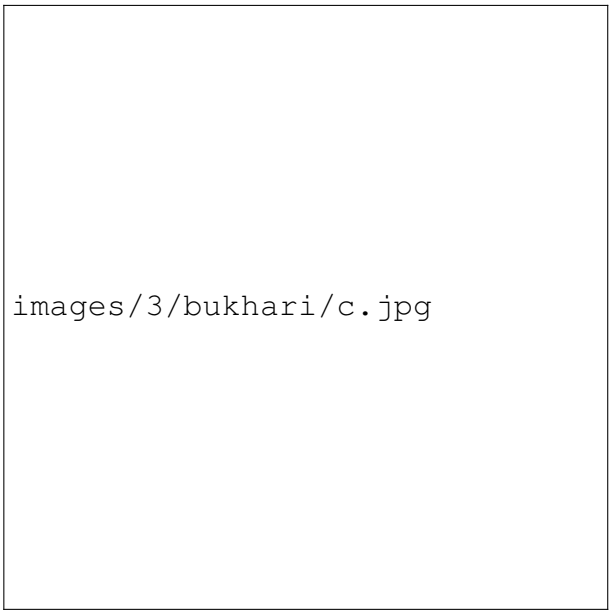
Sincronização das *Snakes*: As *snakes* de topo e de base têm o mesmo número de pontos. Porém as distâncias verticais entre seus pontos correspondentes não são as mesmas. Por isto, a distância média é calculada e todas as distâncias são normalizadas.

Alongamento das *Snakes*: Primeiro, a média do coeficiente de inclinação α do par de *snakes* é calculada. Depois, cada *snake* é alongada para ambas direções (direita e esquerda) por um tamanho W e com inclinação α. W é o tamanho médio da espessura das componentes direita e esquerda de cada *snake*. O retângulo que rodeava o CC também é alongado, de forma que fique o dobro do seu tamanho anterior. Todos os CCs tocados pelo alongamento da *snake* são marcados como processados. As Figuras 3.8b e 3.8c ilustram o processo neste estado no segundo e terceiro ciclo, respectivamente.

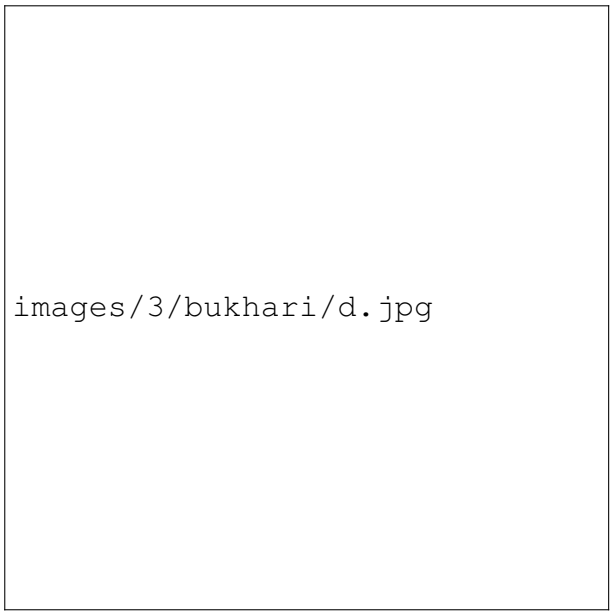
Pós-processamento: Duas características de cada par de *snake* são avaliadas: a inclinação média e a distância entre a *snake* de topo e a de base. Os pares que fogem dos *thresholds* definidos para estas duas medidas, são eliminados da segmentação final.



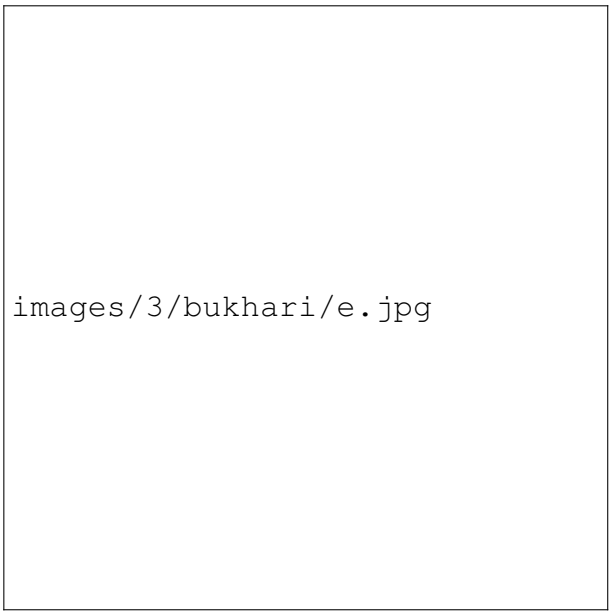
(a) Par de *snakes* iniciais.



(b) Par de *snakes* iniciais após primeiro ciclo de deformação.



(c) Par de *snakes* iniciais após segundo ciclo de deformação.



(d) Par de *snakes* iniciais após terceiro ciclo de deformação.

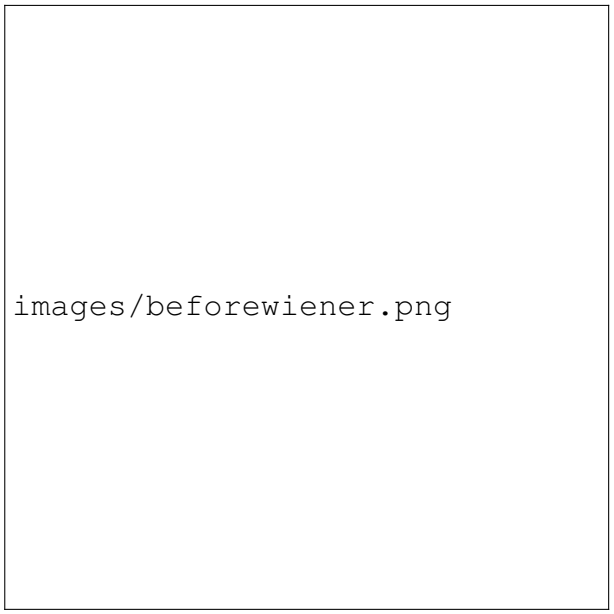
Figura 3.8: Ciclo de deformação das *snakes*. Extraído de (**?**).

Rotulação: Depois de todo o processo, cada grupo de *snakes* com sobreposição são consideradas como uma linha de texto. As componentes removidas no pós-processamento são atribuídas às linhas mais próximas.

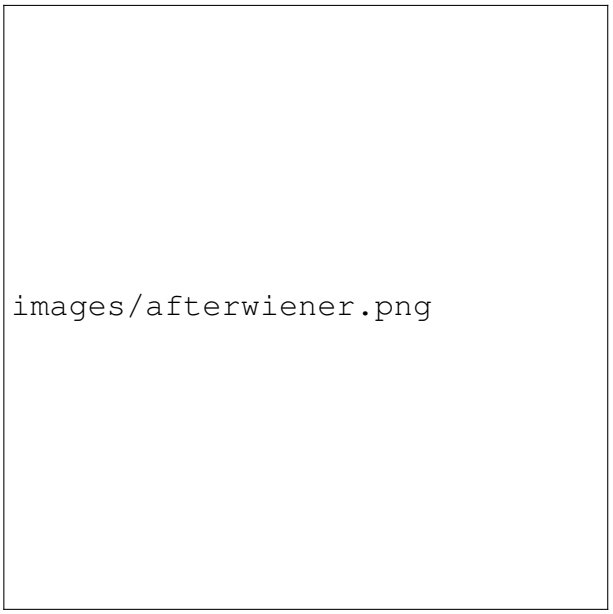
O resultado da segmentação final do método proposto por Bukhari *et al.* pode ser encontrado na Figura 3.9, onde as linhas de texto segmentadas estão exibidas com cores diferentes entre si. Para realização dos seus experimentos foi utilizada a base **?**).

Figura 3.9: Segmentação de linhas distorcidas pela proposta de Bukhari *et al.* Extraído de **?**)

.



(a) Imagem Original



(b) Imagem após aplicação do filtro de Wiener

Figura 3.10: Filtro de Wiener 3x3 (Extraído de **?**)).

3.2.3 *Stamatopoulos et al.*

Stamatopoulos *et al.* propuseram uma metodologia de *dewarping* baseada no conteúdo do documento. A correção da imagem é realizada em dois passos: a correção grosseira e o ajuste fino. O primeiro passo tem o objetivo de determinar a superfície curvada do documento, baseando-se apenas no conteúdo textual. A superfície curvada detectada é então projetada em um plano retangular 2D. Depois, é realizado um ajuste em cada palavra visando corrigir as distorções locais.

Antes de aplicar o método de correção, uma limiarização adaptativa capaz de preservar e melhorar o texto em imagens degradadas e de baixa qualidade é aplicado na imagem (**?**). Este pré-processamento pode ser dividido em quatro etapas: Suavização, Estimativa do *Foreground*, Estimativa do *Background* e Limiarização.

A Suavização é a aplicação do filtro de Wiener (**?**) que é utilizado em imagens degradadas e de baixa qualidade. A função de mapeamento da imagem cinza de entrada $I_s(x,y)$ para imagem cinza de saída $I(x,y)$ é dada pela Equação 3.28, onde μ é a média local, σ^2 é a variância em uma janela 3x3 na vizinhança do pixel e v^2 é a média de todas as variância calculadas na vizinhança. A Figura 3.10 ilustra o resultado da aplicação do filtro.

$$I(x,y) = \mu + \frac{[(\sigma^2 + v^2)(I_s(x,y) - \mu)]}{\sigma^2} \tag{3.28}$$

Para melhor compreensão, consideraremos *pixels* com valor "1"*foreground* e "0"*background*. O *foreground* é estimado com a aplicação do método de binarização adaptativo de Sauvola (**?**). A imagem binária resultante $S(x,y)$ desta binarização é o *foreground*. O *background* $B(x,y)$ é calculado com base em $S(x,y)$. Para todos os *pixels* "0"em $S(x,y)$, $B(x,y)$ recebe o

valor correspondente de $I(x,y)$. Para os demais *pixels*, é aplicada a regra de interpolação descrita pela Equação 3.29, onde dx x dy é uma janela definida tal que consiga cobrir dois caracteres.

$$B(x,y) = \frac{\sum_{\substack{x-dx<ix<x+dx \\ y-dy<iy<y+dy}} I(ix,iy)[1-S(x,y)]}{\sum_{\substack{x-dx<ix<x+dx \\ y-dy<iy<y+dy}} 1-S(x,y)} \tag{3.29}$$

A imagem binária final $T(x,y)$ é calculada com base em $B(x,y)$ e $I(x,y)$:

$$T(x,y) = \begin{cases} 1 & \text{Se } B(x,y) - I(x,y) > d(B(x,y)) \\ 0 & \text{caso contrário} \end{cases} \tag{3.30}$$

$$d(B(x,y)) = q\delta\left(\frac{1-p2}{1+exp\left(\frac{-4B(x,y)}{b(1-p1)}+\frac{2(1+p1)}{1-p1}\right)}+p2\right) \tag{3.31}$$

$$\delta = \frac{\sum_y^x B(x,y) - I(x,y)}{\sum_y^x S(x,y)} \tag{3.32}$$

$$b = \frac{\sum_y^x B(x,y)(1-S(x,y))}{\sum_y^x 1-S(x,y)} \tag{3.33}$$

na qual $q = 0,6$, $p1 = 0,5$ e $p2 = 0,8$.

Antes da correção da imagem, o algoritmo de Stamatopoulos *et al.* realiza a detecção de palavras e linhas com objetivo de estimar a superfície distorcida. Nesta fase existe uma tolerância aos erros no agrupamento das palavras em linhas, já que o objetivo é obter uma estimativa grosseira da distorção. Para isto, é necessário determinar a altura majoritária entre todos os caracteres (AH) para remoção de ruído, imagens e gráficos. Para isto, os CC da imagem são calculados e um histograma de todas as alturas é construído. O valor de AH é o valor máximo do histograma. Os componentes que não satisfizerem as condições descritas nas Equações 3.34, 3.35 e 3.36 são removidos.

$$h > 3 * AH \tag{3.34}$$

$$h < \frac{AH}{4} \tag{3.35}$$

$$w < \frac{AH}{4} \tag{3.36}$$

na qual w e h são largura e altura do componente, respectivamente.

O próximo passo é formar as palavras. Para isto, utiliza-se o *Run Length Smoothing Algorithm* (RLSA) para ligar os caracteres próximos. Com os caracteres ligados, os CC da imagem são calculados novamente e obtém-se as palavras. As linhas são construídas ligando

(a) Detecção da superfície distorcida **(b)** Transformação de Correção

Figura 3.11: Cálculo da superfície (Extraído de **?**)).

as palavras adjacentes. Seja (x_1, y_1, x_2, y_2) o *bounding-box* da palavra base e (x'_1, y'_1, x'_2, y'_2) o *bounding-box* de uma candidata a vizinha, entre todas as palavras a direita da palavra base que satisfazem a condição $[y_1, y_2] \bigcap [y'_1, y'_2] \neq \varnothing$, a de menor distância $(x'_1 - x_2)$ é escolhida apenas se $0 < D < 6 * AH$, onde D é o valor da distância. Esta última condição assegura que a palavra mais próxima seja a escolhida para compor a linha. Depois de fazer para a direita, o mesmo processo para coletar as palavras à esquerda é realizado.

A estratégia utilizada no método proposto por Stamatopoulos *et al.* é realizar a correção das grandes distorções na fase "grosseira"e depois realizar os ajustes mais precisos na fase posterior. Para realizar a transformação é necessário estimar a superfície curvada do texto, como mostrado em verde na Figura 3.11a . Para isto, é necessário encontrar os pontos A,B,C e D. Primeiro, as linhas menores que 80% do tamanho médio das linhas são excluídas. Os segmentos AD e BC que correspondem aos limites da esquerda e da direita da superfícies são encontrados por um processo iterativo.

O começo e o fim de cada linha são coletados. As linhas que não começam (ou terminam) na mesma identação que as demais (início de parágrafo, por exemplo) precisam ser eliminadas do cálculo da estimativa de AD e BC. Se o desvio da linha estimada for maior que AH, então o ponto (de começo ou de fim) de maior distância é eliminado. O segmento é recalculado até que a condição seja satisfeita ou restem duas linhas. As curvas AB e DC são calculadas a partir das linhas superior e inferior restantes da fase anterior, respectivamente. Depois de estimar a superfície curvada, dada pelas curvas AB e DC e pelos segmentos AD e BC, o próximo passo é realizar a transformação. Como mostra a Figura 3.11b , é preciso mapear os pixels da projeção ABCD para o retângulo A'B'C'D'.

Para encontrar A'B'C'D', consideramos $A'(x'_1, y'_1) = A(x_1, y_1)$. Os demais pontos são

calculados levando em consideração a espessura W e a altura H do retângulo (Equações 3.37 e 3.38). O valor de $arclen(A,B)$ é o comprimento do arco que liga os pontos na curva AB e |AD| é a distância euclidiana entre A e D. Depois de definir o retângulo alvo, cada ponto $O(x,y)$ na superfície curvada é mapeado para um $O'(x',y')$ em $A'B'C'D'$. Cada $O(x,y)$ é definido por dois pontos (E e G) calculados a partir das curvas AB e CD. Os pontos E e G devem satisfazer a condição descrita em Eq. 3.39. O ponto $O'(x',y')$ no retângulo alvo é calculado de modo que preserve a razão entre a superfície curvada e a área retangular nas direções x e y. Baseando-se nisto, os pontos $Z'(x',y'_1)$ e $H(x'_1,y')$ são calculados pelas Equações 3.40 e 3.41.

$$Width = min(arclen(A,B), arclen(D,C)) \tag{3.37}$$

$$Height = min(|AD|, |BC|) \tag{3.38}$$

$$\frac{arclen(A,E)}{arclen(A,B)} = \frac{arclen(D,G)}{arclen(D,C)} \tag{3.39}$$

$$|AZ| = \frac{Width}{arclen(A,B)} arclen(A,E) \tag{3.40}$$

$$|A'H| = \frac{Height}{arclen(E,G)} arclen(E,O) \tag{3.41}$$

O último passo é o ajuste fino de cada uma das palavras. Para isto, uma nova detecção de palavras e linhas é executada na imagem corrigida. A inclinação de todas as palavras é determinada em relação ao eixo horizonal. Cada palavra é rotacionada de modo que sua linha de base fique paralela ao eixo horizonal. Depois, todas as palavras de todo o texto precisam ser verticalmente movidas para que haja um alinhamento horizontal. Este processo é ilustrado na Figura 3.12.

Apesar de ter conseguido valores altos de taxa de acerto nos seus experimentos, o método proposto por Stamatopoulos é bastante dependente da presença de texto bem estruturado nas imagens. As várias análises de CC tornam a heurística proposta bastante custosa. Além disto, ao ser apresentada a layouts um pouco mais complexos, como presença de coluna dupla, o método não apresenta bons resultados.

3.3 Considerações

As abordagens para correção de imagens capturadas por câmera podem ser divididas em duas: correção baseada na reconstrução do modelo 3D que, geralmente, requer dispositivos específicos ou pressuposições que nem sempre são válidas; e abordagens de correção 2D, que se baseiam exclusivamente no conteúdo da imagem para realizar a correção. Estas são, na maioria

Figura 3.12: (a) Palavras distorcidas. (b) Cálculo da inclinação das palavras. (c) Correção de Inclinação. (d) Alinhamento das palavras.

das vezes, bastante dependentes do texto contido na imagem do documento.

Neste capítulo foram apresentadas três abordagens do estado da arte de correção de imagens capturadas por câmeras baseados apenas em informações contidas na imagem. A abordagem de Fujimoto objetiva encontrar os *Vanishing Points* da imagem, porém para encontrar os VPs, é necessário segmentar as linhas de texto e as componentes verticais dos caracteres. Além disso, o método pressupõe que a as linhas sempre estarão dispostas horizontalmente e a inclinação máxima nunca será maior que 45º.

A técnica proposta por Bukhari *et al.* é baseada exclusivamente na segmentação das linhas de texto, e portanto, pressupõe que o documento terá o formato de texto corrido. Além disto, Bukhari *et al.* utilizam um par de *snakes* – algoritmo computacionalmente caro – para cada CC de texto.

De maneira semelhante aos dois exemplos citados, Stamatopoulos *et al.* também desenvolveram uma técnica de correção de distorção de perspectiva baseado no mapeamento do texto distorcido para um mapeamento não-distorcido. Além disto, esta abordagem não funciona para textos com mais de uma coluna e é necessário fornecer o tamanho médio dos caracteres, portanto, pressupõe que o texto terá tamanho de fonte uniforme.

4

Correção de Imagens Utilizando a Transformada de Hough e o Descritor de HOG

Este capítulo explica uma nova abordagem proposta para correção de imagens capturadas por dispositivos móveis. A seguir, cada etapa do método é detalhada, ilustrada e justificada. Importante observar que muitas das imagens utilizadas ao longo deste capítulo são documentos pessoais e, a fim de preservar informações confidenciais presentes nestes documentos, algumas informações foram censuradas com tarjas pretas.

Como explicado no Capítulo 3, as correções de documentos capturadas por dispositivos móveis podem ser baseadas em reconstrução do modelo 3-D da imagem ou no processamento de imagens 2-D. A técnica proposta por este trabalho utiliza apenas informações contidas na imagem 2-D sem nenhuma dependência de conhecimentos *a priori*, como iluminação e tipo de lente, ou uso de dispositivos específicos.

As propostas de retificação apresentadas na Seção 3.2 são bastante dependentes de informações textuais contidas nos documentos digitalizados. Estas técnicas são experimentadas, muitas vezes, em imagens de páginas de livros ou documentos com grande presença de textos com fonte uniforme. Mesmo a abordagem de **?**), baseada em *vanishing points*, tem uma dependência de encontrar as linhas de texto e do formato dos caracteres para determinar a distorção da imagem.

O objetivo deste trabalho é realizar a correção de imagens com *background* complexo, iluminação heterogênea, variação de tamanho da fonte nos textos para melhorar a legibilidade por humanos ou máquinas. Fotografias de documentos pessoais são um bom exemplo deste desafio. Como pode ser visto na imagem de uma Carteira de Habilitação brasileira (Figura 4.1a), as informações textuais do documento estão espalhadas em uma organização própria tendo característica de um formulário estruturado do que de um texto corrido. Além disto, é possível perceber que as fontes do texto do documento diferem de tamanho, espessura e tonalidade.

Quando a imagem é capturada por um dispositivo móvel, em um ambiente não controlado, vários outros desafios ficam evidentes. A Figura 4.1b mostra o mesmo documento capturado por uma câmera de celular. É possível perceber um plano de fundo não comportado e distorções de

(a) Captura por escaner **(b)** Capturada por celular

Figura 4.1: CNH brasileira. (a) Layout complexo e mudança de fonte. (b) Evidentes distorções de perspectiva, inclinação e perda de parte do documento.

inclinação e leve perspectiva. Além disto, parte do documento é perdida no enquadramento da fotografia.

A Transformada de Hough (HT) tem sido utilizada para estimar *vanishing points* (**?**) e para retificação de imagens (**??**). Usualmente, algoritmos de detecção de borda são utilizados antes da aplicação da HT, buscando diminuir o espaço de processamento, o que torna o método bastante dependente do resultado da extração de borda. Ainda assim, a HT requer uma grande quantidade de memória e tem um custo alto de processamento (**?**).

Por este motivo, no método proposto, a HT utiliza os HOGs ao invés da imagem original. Como exposto na seção 2.1, o descritor de HOG subdivide a imagem em células de tamanho n. Portanto, para uma imagem $1024x1024$, com $n = 16$, uma representação de tamanho $64x64$ é gerada, ou seja, uma redução de 16 vezes da dimensão da imagem. Esta nova maneira de utilizar a HT foi batizada de Transformada de Hough com Histogramas de Gradientes Orientados (HT-HOG) e os detalhes desta implementação estão na seção 4.2.

Após a aplicação da HT-HOG, o desafio é encontrar as linhas que descrevem os limites dos documentos. Em posse das linhas, é possível determinar os cantos do documento e efetuar o *dewarping*.

A Figura 4.2 mostra as etapas do método proposto. O primeiro passo é o aumento do contraste da imagem. Em seguida, o algoritmo de HOG é aplicado gerando os histogramas. A imagem no espaço de Hough é gerada a partir destes histogramas e uma análise é realizada para detectar as linhas. A última etapa é a aplicação da correção na imagem. As etapas do método proposto serão detalhadas nas demais seções deste capítulo.

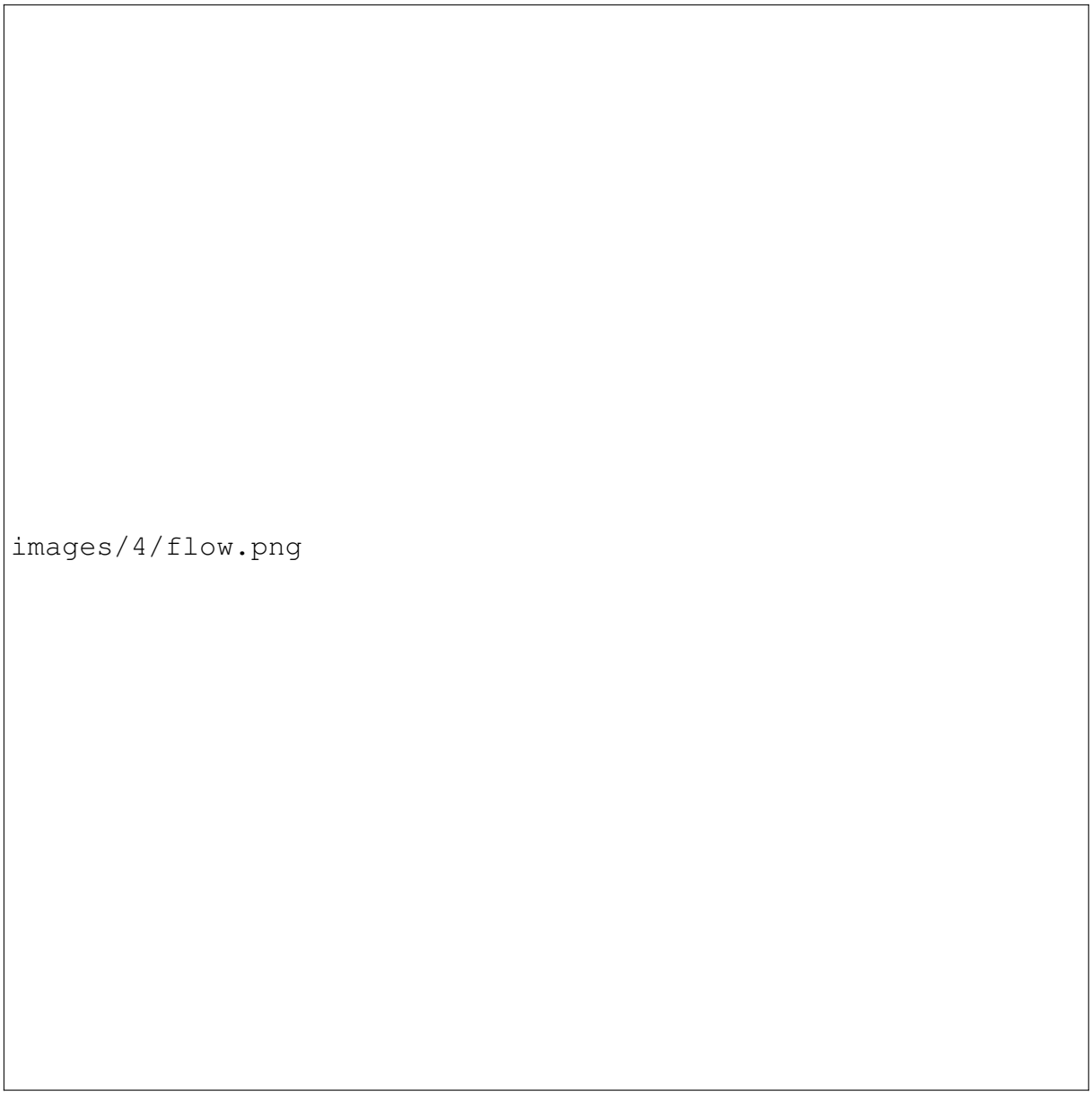
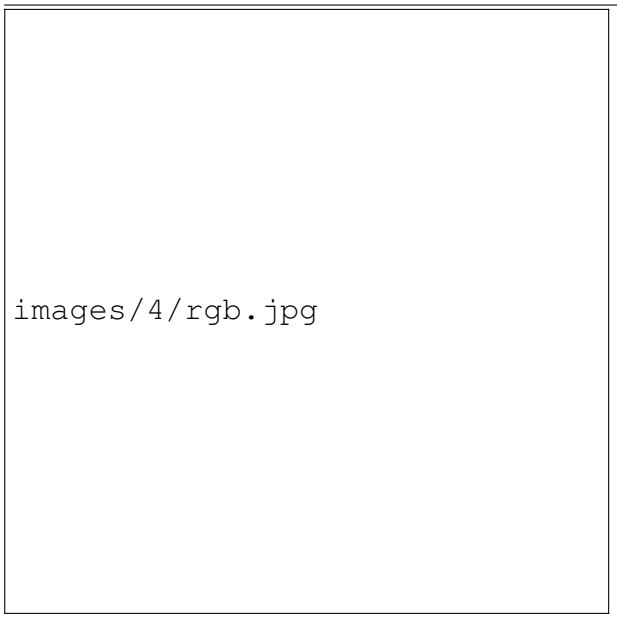


Figura 4.2: Diagrama de alto nível do método proposto

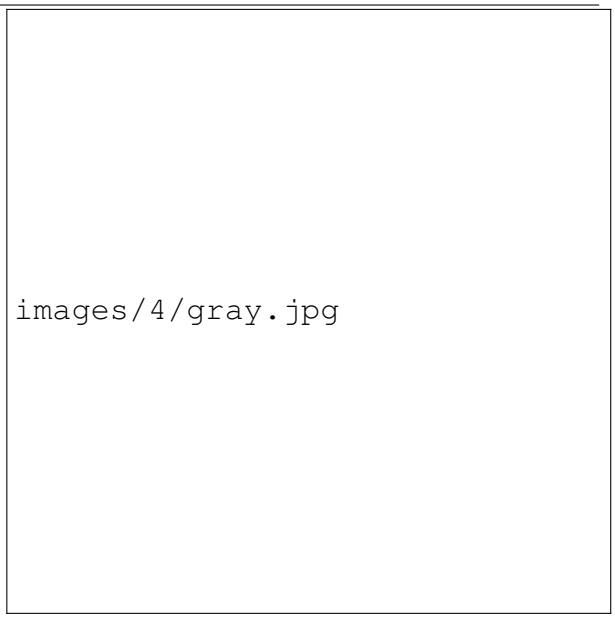
4.1 Ajuste do Contraste

A presença desta etapa traz resultados positivos, pois, o aumento do contraste realça as regiões de transição de intensidade trazendo informações relevantes para o descritor de HOG. O primeiro passo do ajuste é mapear a imagem colorida em uma imagem em tons de cinza utilizando a Equação 4.1. Para cada pixel na imagem original, a intensidade I é calculada pelos valores dos canais R, G e B. O resultado deste mapeamento está ilustrado na Figura 4.3.

$$I = 0.2989 * R + 0.5870 * G + 0.1140 * B \tag{4.1}$$



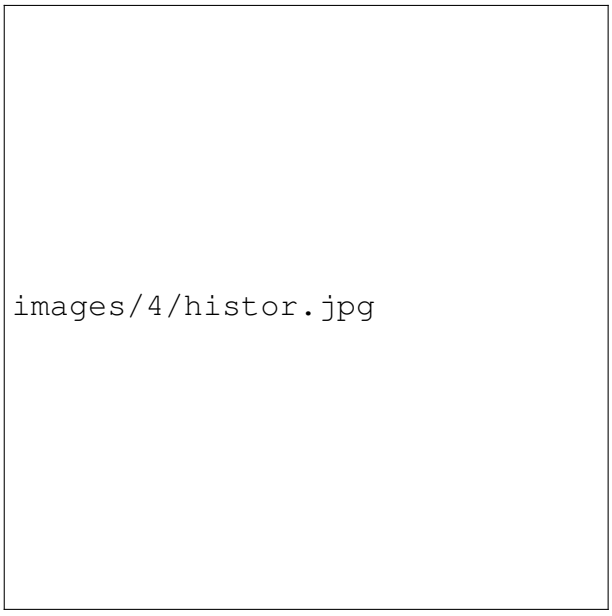
(a) Imagem Colorida.



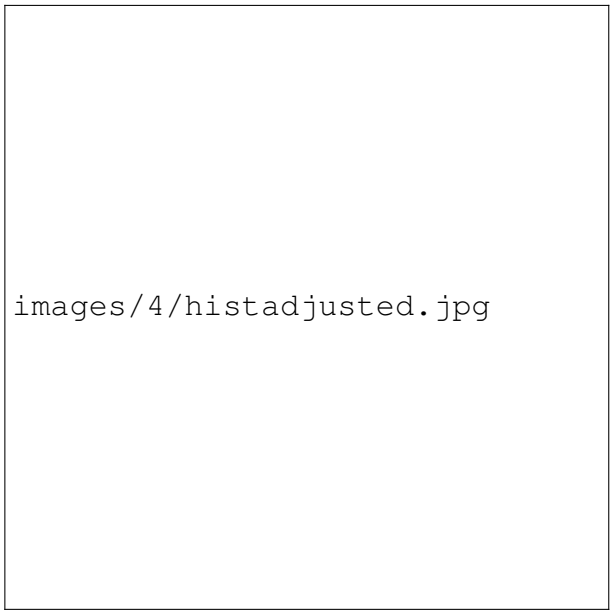
(b) Imagem em Tons de Cinza.

Figura 4.3: Transformação da imagem colorida em cinza.

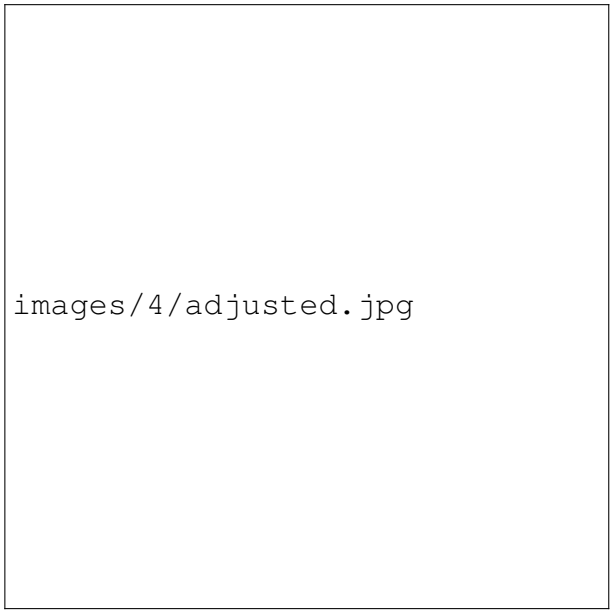
A partir do histograma da imagem cinza, os valores de L e U são definidos de forma que sejam os limites inferior e superior, respectivamente, tal que o somatório dos pixels de 0 até L e U até 255, seja 1% do total de pixels (Figura 4.4a). Então, uma função de transformação é definida tal que, todos os valores no intervalo $[0, L]$ são mapeados para 0 enquanto todos os valores em $[U, 255]$ são mapeados para 255. Os valores em $[L, U]$ são mapeados para o intervalo $[1, 254]$ equivalentemente (Figura 4.4b). O resultado deste processo está presente na Figura 4.4c. Esta procedimento é mesmo realizado pela função *imadjust* do MATLAB (**?**).



(a) Histograma Original.



(b) Histograma Ajustado.



(c) Imagem Ajustada.

Figura 4.4: Ajuste do Contraste da Imagem

4.2 Aplicação da HT-HOG

O objetivo desta etapa é gerar o espaço de parâmetros $H(\theta,\rho)$ para posterior seleção das linhas (Seção 4.3). A Figura 4.5 apresenta os passos deste processo. Primeiro, os HOGs da imagem são calculados e apenas os histogramas verticais e horizontais são analisados. Após filtrar HOGs de ruído, a HT é aplicada. Caso a imagem apresente deformação, a HT é aplicada nos HOGs inclinados para geração do $H(\theta,\rho)$.

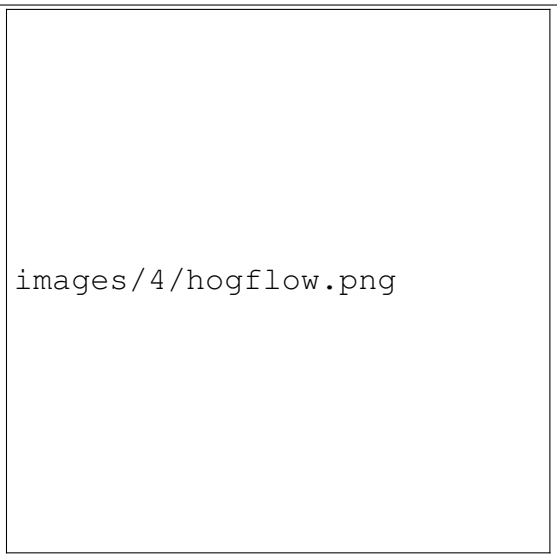


Figura 4.5: Diagrama do HT-HOG

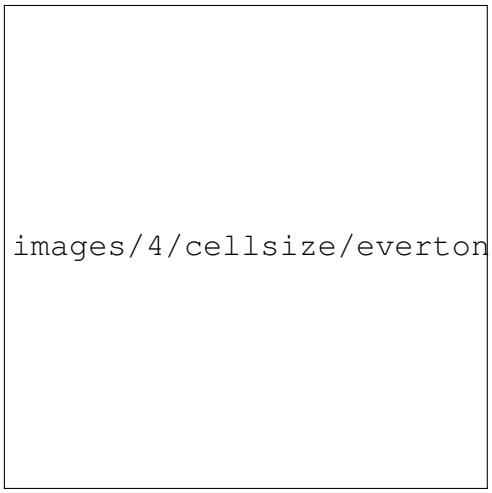
4.2.1 Cálculo do HOG da Imagem

O princípio do descritor de HOG é dividir a imagem em células e estabelecer qual a direção dominante em cada uma dessas sub-partes. Nesta etapa, temos dois parâmetros livres o *cellSize* e o *numBins*. O *cellSize* determina a dimensão da célula na qual a imagem será subdividida. Este parâmetro tem um papel importante no desempenho do método proposto, uma vez que para *cellSizes* muito pequenos o desempenho da HT-HOG se aproxima da HT. Por outro lado, *cellSizes* muito altos não geram informação suficiente para determinar as linhas da imagem. O *numBins* determina o número de orientações do histograma. Este parâmetro pode assumir valores entre 1 e 180, porém, assim como *cellSize*, *numBins* próximo a 180 faz o desempenho ser deteriorado, aproximando-se da HT.

A Figura 4.6 mostra exemplos de HOGs para vários *cellSizes*. É possível perceber que à medida que o *cellSize* aumenta, as informações de contorno da imagem são perdidas. Além disto, determinar o fator de redução de dimensionalidade da imagem. Da mesma forma, a Figura 4.7 ilustra, para o mesmo segmento de imagem, vários *numBins*. É possível perceber que *numBins* grandes granularizam mais o resultado, fornecendo mais assertividade à direção real do gradiente.

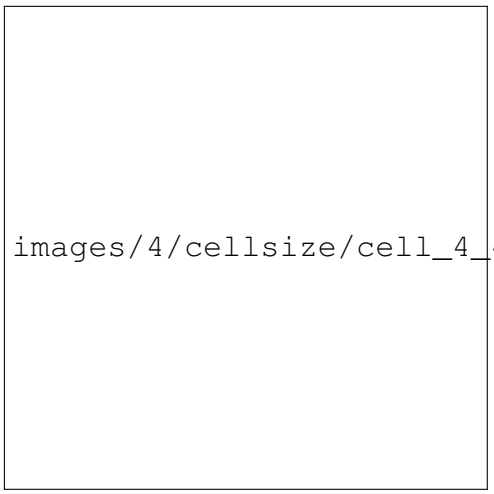
4.2.2 Seleção de HOGs sem Inclinação

Após a geração dos HOGs da imagem, os histogramas com direção dominante verticais e horizontais são filtrados. Após este procedimento, é possível determinar se existem na imagem deformações evidentes. O Algoritmo 2 explica como é realizado este procedimento. Na prática, todos os histogramas são analisados e, apenas os que tem orientação 0º e 90º são selecionados, desde que tenham coeficiente maior que *limiar*. Este parâmetro é escolhido de forma que seja o valor máximo dentre todos os histogramas. Este procedimento tem o objetivo de manter apenas



(a) Imagem original 1944 x 2592.

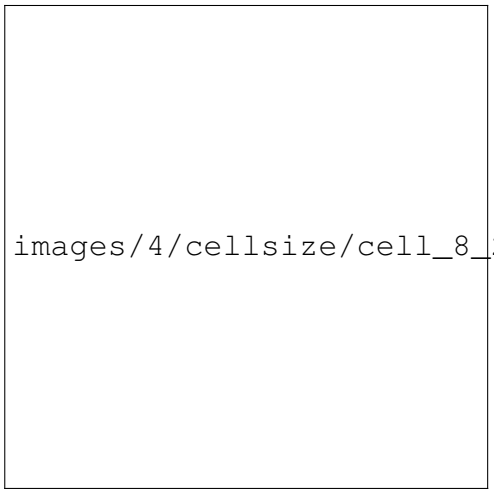
(b) Amostra imagem original



(c) HOG 486 x 648, *cellSize = 4*

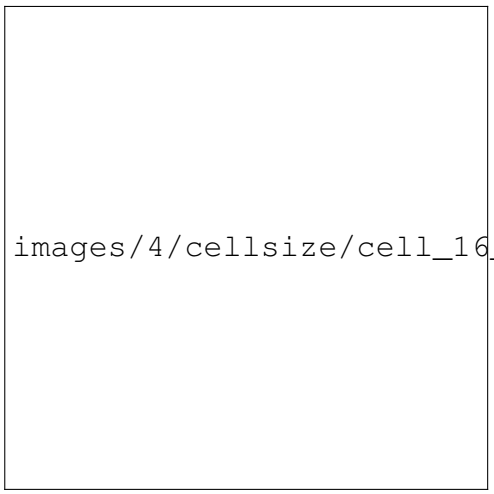
(d) Amostra Imagem *cellSize = 4*

componentes representativos na avaliação dos gradientes importantes. O resultado desta seleção é apresentado na Figura 4.8, onde a Figura 4.8a representa os HOGs originais da imagem e a Figura 4.8b os verticais e horizontais.



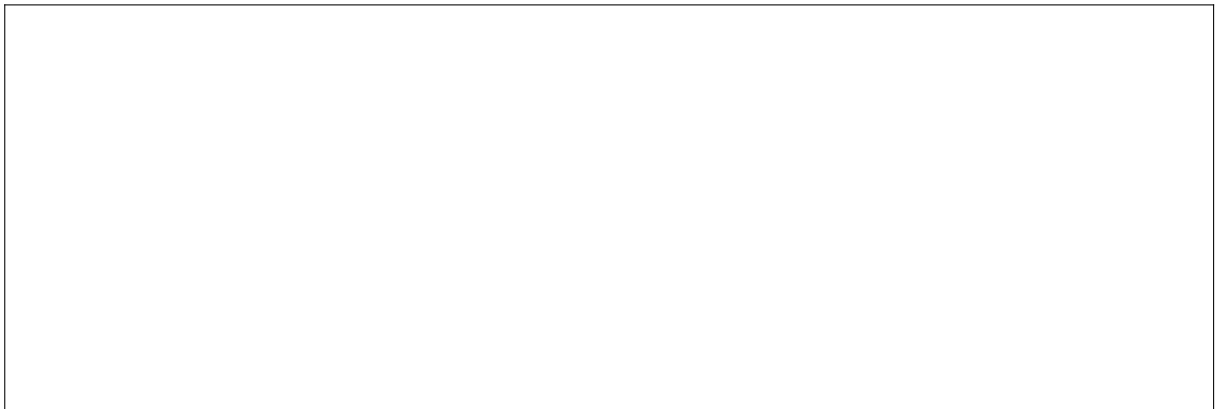
(e) HOG 243 x 324, *cellSize = 8*

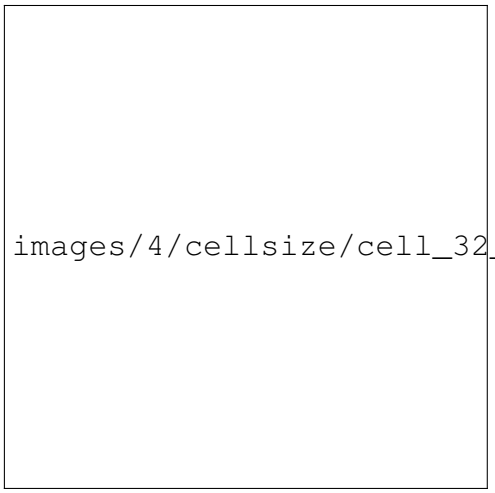
(f) Amostra Imagem *cellSize = 8*



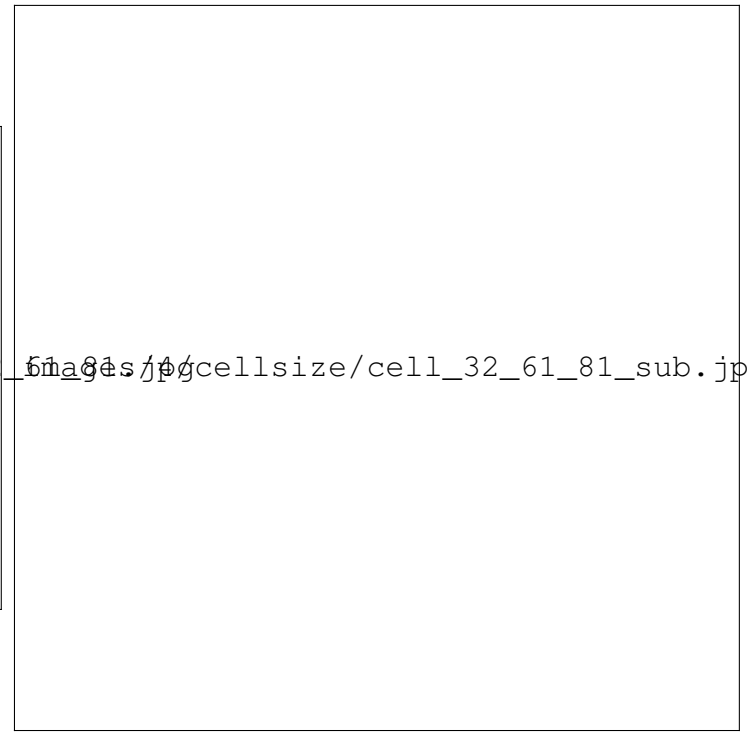
(g) HOG 112 x 162, *cellSize = 16*

(h) Amostra Imagem *cellSize = 16*

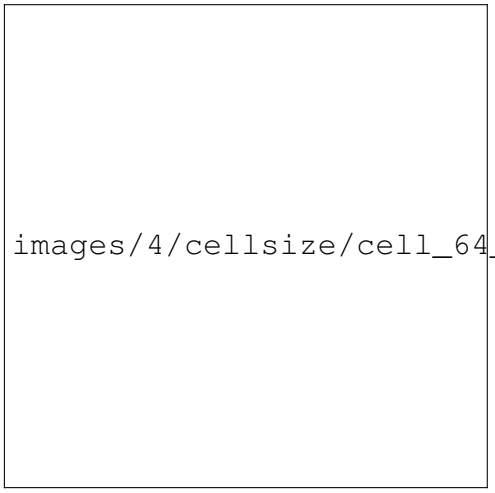




(i) HOG 32 x 61, *cellSize = 32*



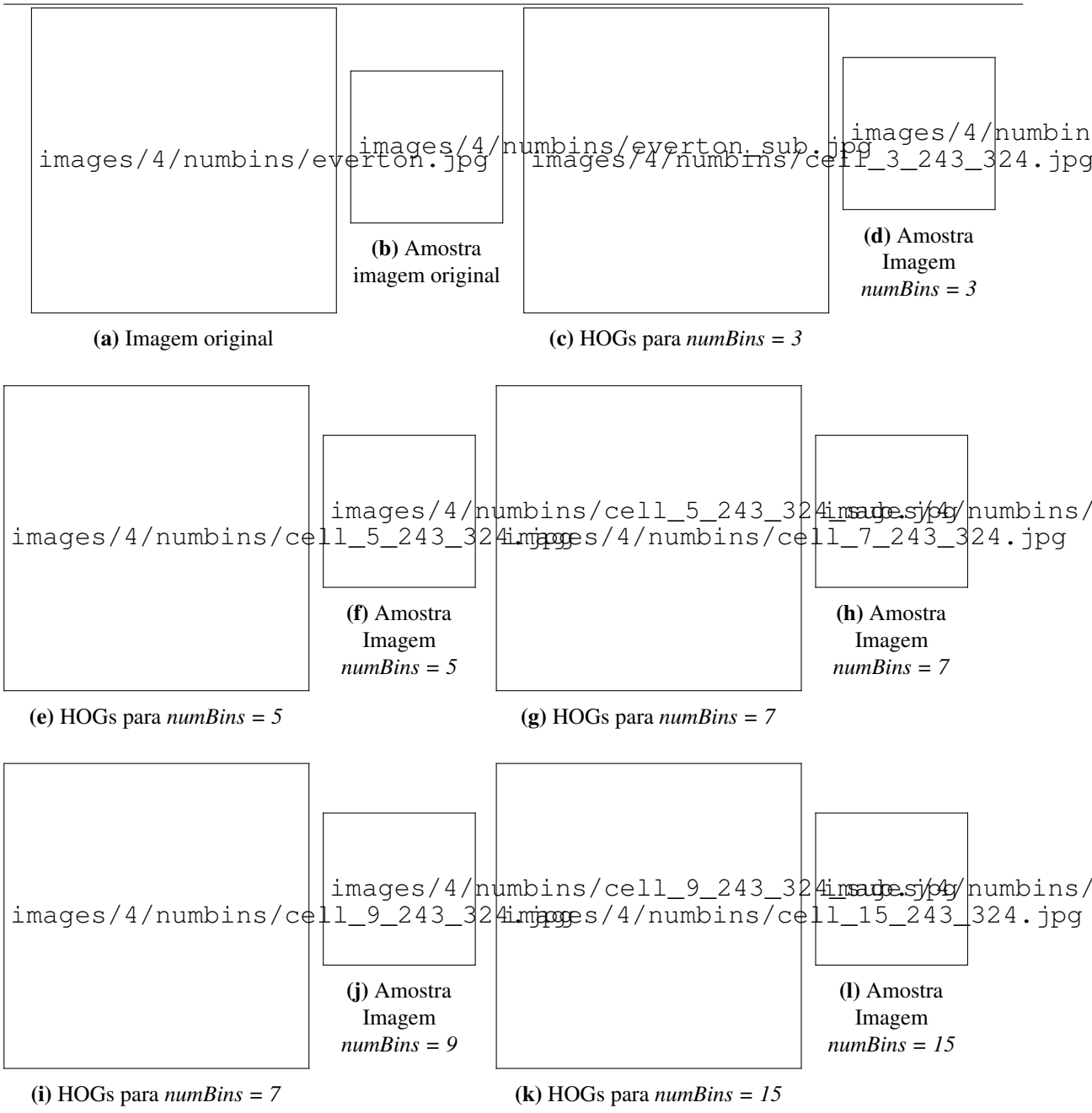
(j) Amostra Imagem *cellSize = 32*



(k) HOG 30 x 41, *cellSize = 64*

(l) Amostra Imagem *cellSize = 64*

Figura 4.6: Aplicação de HOG para vários valores de *cellSize*. Para melhor visualização, as imagens ampliadas foram complementadas.



(a) Imagem original

(b) Amostra imagem original

(c) HOGs para *numBins = 3*

(d) Amostra Imagem *numBins = 3*

(e) HOGs para *numBins = 5*

(f) Amostra Imagem *numBins = 5*

(g) HOGs para *numBins = 7*

(h) Amostra Imagem *numBins = 7*

(i) HOGs para *numBins = 7*

(j) Amostra Imagem *numBins = 9*

(k) HOGs para *numBins = 15*

(l) Amostra Imagem *numBins = 15*

Figura 4.7: Aplicação de HOG para vários valores de *numBins*. Para melhor visualização, as imagens ampliadas foram complementadas.

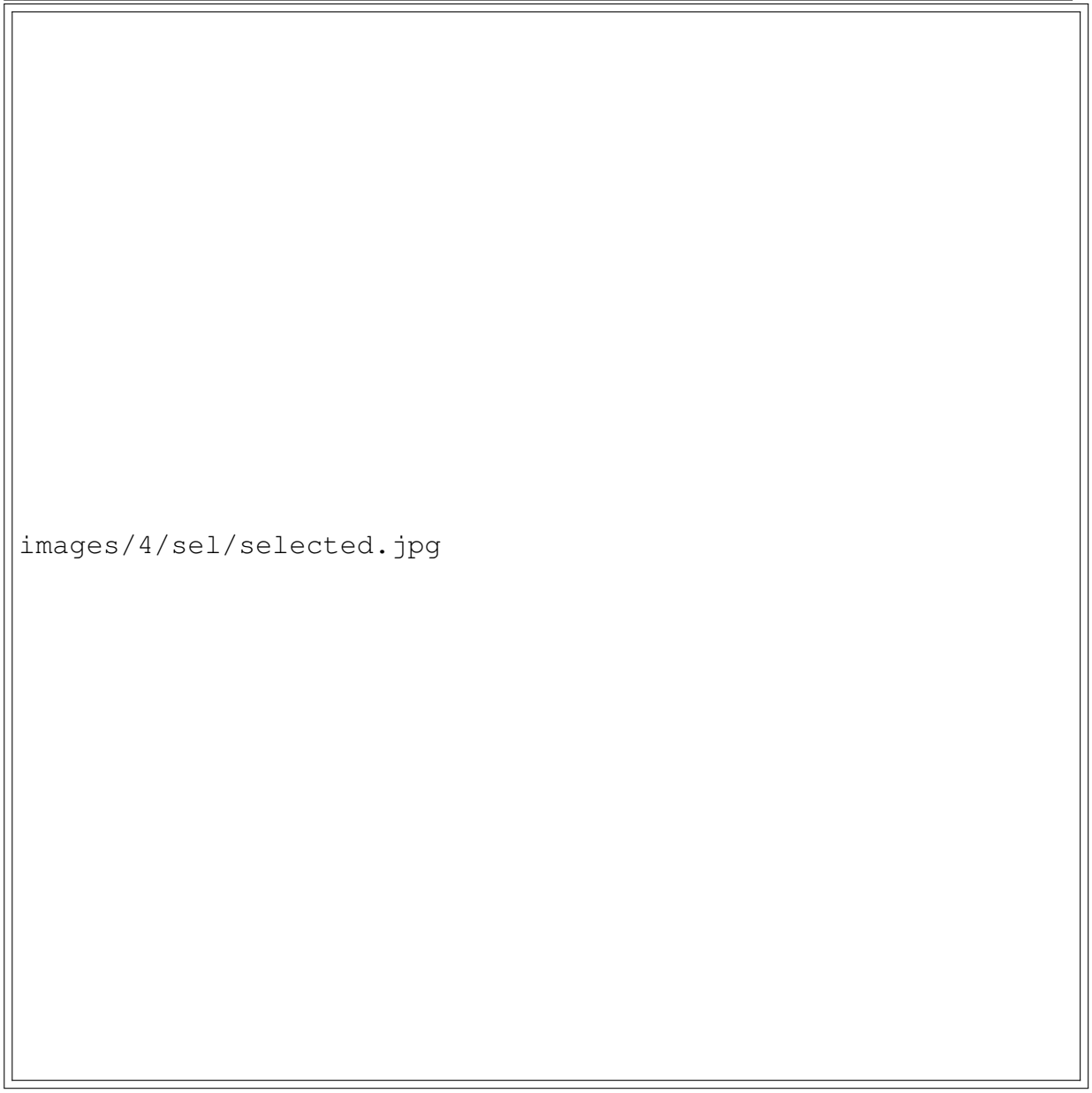
4.2.3 Limpeza de Componentes

Analisando a Figura 4.8b percebe-se que mesmo utilizando a filtragem por *limiar* alguns componentes anômalos persistem na imagem. Por isto, depois da seleção dos histogramas sem inclinação, é necessário remover estes ruídos. A vizinhança de cada histograma é analisada e, caso não existam vizinhos ativos, o histograma é eliminado. O histograma permanece na imagem apenas se existir algum vizinho com uma direção semelhante a dele. Na prática, se a diferença de inclinação entre o HOG e seu vizinho maior que 30º, este HOG é eliminado. Este valor foi fixado pois, permite alguma tolerância em manter HOGs vizinhos sem mesmo coeficiente

Algoritmo 2 Seleção de HOGs sem Inclinação

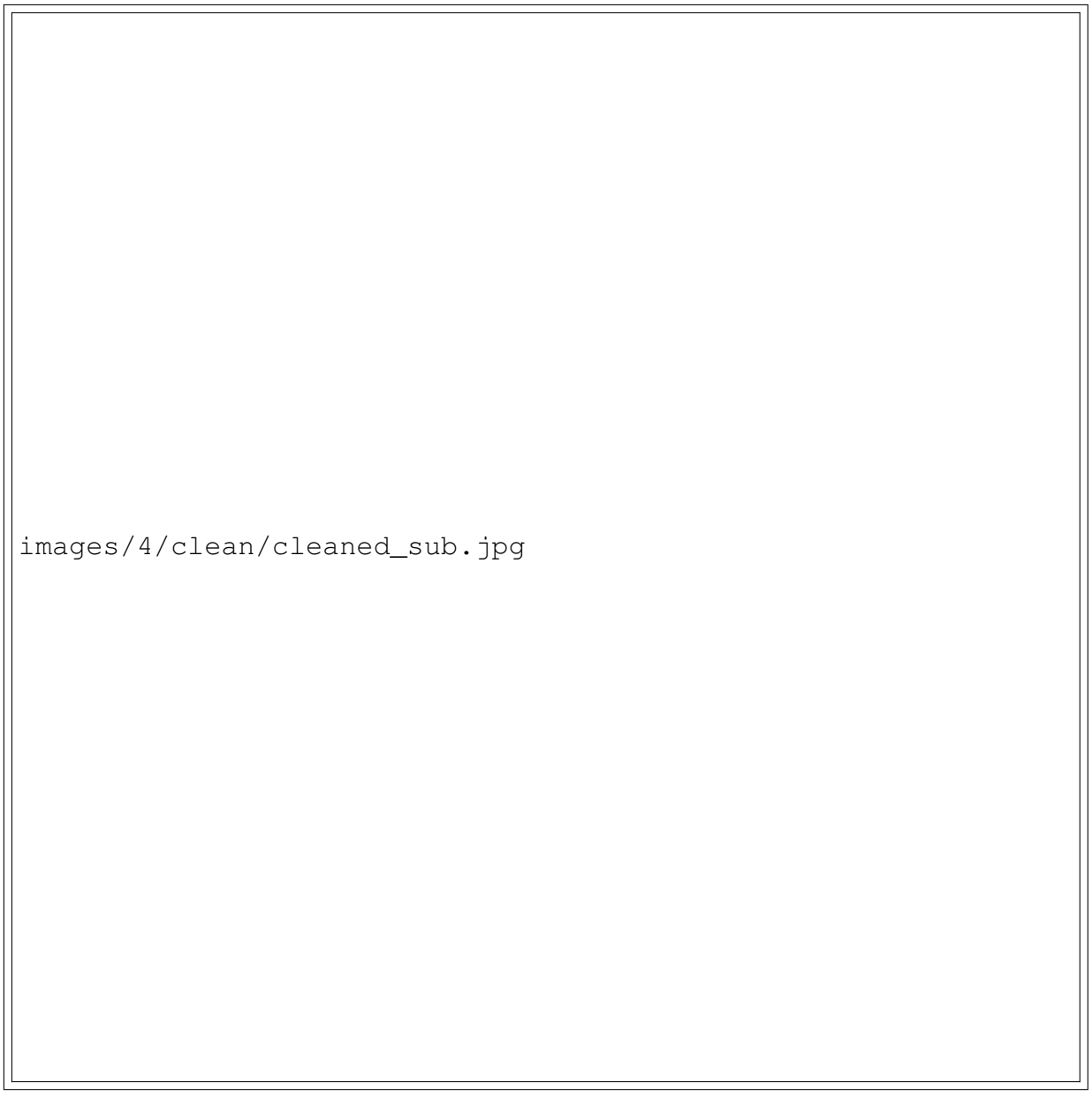
```
1:  procedure HOGSELECTION(H)                          ▷ H são os histogramas da Imagem
2:      cols ← H.cols
3:      rows ← H.rows
4:      for i ← 1, rows do
5:          for j ← 1, cols do                          ▷ Itera sob todas as células
6:              hist ← H(i, j)                          ▷ Seleciona histograma atual
7:              coef ← getMaxCoefficient(hist)          ▷ Seleciona maior coeficiente em hist(i,j)
8:              if coef > limiar then                   ▷ Avalia representatividade de coef
9:                  θ ← getMaxOrientation(hist)         ▷ Seleciona orientação de hist(i,j)
10:                 if θ = 90 ∨ θ = 0 then
11:                     NoInclinationH = H(i, j)        ▷ Constrói H sem Inclinação
12:                 end if
13:             end if
14:         end for
15:     end for
16: end procedure
```

angular, porém, com direção semelhante. A Figura 4.9 mostra o antes e o depois do processo de limpeza.



(b) HOGs 0º e 90º

Figura 4.8: Seleção de HOGs sem Inclinação.



(b) HOGs limpos

Figura 4.9: Limpeza dos componentes anômalos.

4.2.4 Aplicação da HT

O algoritmo da HT utilizado neste passo é diferente da HT original (Seção 2.2) que avalia todos os pixels ativos da imagem. Na HT-HOG, apenas os histogramas remanescentes da limpeza são avaliados. Além disto, cada HOG da imagem tem sua orientação dominante definida. Portanto, não é necessário avaliar todas as linhas (θ, ρ) possíveis, pois, cada histograma representa uma orientação.

Este é um dos principais benefícios do uso dos histogramas da imagem ao invés da imagem inteira. Os HOGs diminuem o espaço de busca da HT com o fator de *cellSizes* e elimina a necessidade de iterar sob todos os pares (θ, ρ) possíveis. O Algoritmo 3 detalha como fica

a HT utilizando HOGs como espaço de busca. Para todos os histogramas remanescentes, sua respectiva linha em $H(\theta,\rho)$ é incrementada de um.

Algoritmo 3 Transformada de Hough utilizando HOGs

```
1:  procedure HOUGHTRANSFORM(Hog)
2:      cols ← Hog.cols
3:      rows ← Hog.rows
4:      colsHough ← 2 * sqrt(cols^2 + rows^2)        ▷ O máximo ρ é a diagonal de Hog
5:      rowsHough ← numBins
6:      H ← CreateImage(rowsHough, colsHough)
7:      for i ← 1, rows do
8:          for j ← 1, cols do
9:              if IsValidHistogram(Hog(i, j)) then
10:                 θ = Hog(i, j).orientation
11:                 ρ = i * cos(θ) + j * sin(θ)
12:                 H(i, ρ) ← H(i, ρ) + 1
13:             end if
14:         end for
15:     end for
16: end procedure
```

O resultado da aplicação deste procedimento nos HOGs ilustrados na Figura 4.10 pode ser visualizado na Figura 4.11a. Como existem apenas componentes horizontais (0°) e verticais (90°) nos HOGs da imagem, apenas estes estão presente no gráfico. Para remover o ruído causado pelo serrilhamento na imagem de dimensões $numHogRows \times numHogCols$, utilizamos a regra na Equação 4.2. O *threshold* é definido pela menor dimensão da imagem (Equação 4.3), resultando nos $H(\theta,\rho)$ mais representativos (Figura 4.11b). O motivo de utilizar 25% da menor dimensão – divisão por 4 – é detalhado na próxima seção. O espaço de Hough gerado nesta fase é denominado $H_o(\theta,\rho)$.

Figura 4.10: Imagem de HOGs sem inclinação

(a)

(b)

Figura 4.11: Domínio de Hough utilizando HOG: (a) Original e (b) Após filtragem.

$$H(\theta,\rho) = \begin{cases} 0 & \text{if } H(\theta,\rho) < threshold \\ H(\theta,\rho) & \text{caso contrário} \end{cases} \tag{4.2}$$

$$threshold = \frac{min(numHogRows, numHogCols)}{4} \tag{4.3}$$

4.2.5 Análise do grau de distorção

O resultado presente na Figura 4.11b evidencia o ponto $H_o(\theta = 90, \rho = 138)$. As fórmulas de transformação cartesiano-polar (Equações 2.7 e 2.8) são utilizadas para obter a função que representa a linha reta. É necessário realizar o ajuste de proporção devido à diminuição de dimensionalidade que a imagem sofre quando os histogramas são determinados. Este ajuste é alcançado pela multiplicação do valor de ρ pelo fator de diminuição *cellSize* (Equação 4.4). A função final obtida é $y = 2208$. A Figura 4.12a mostra o desenho desta linha reta que é a representação de um dos limites do documento presente na Figura 4.12b.

$$\beta = \frac{\rho * cellSize}{sin\theta} \tag{4.4}$$

(a) Representação da reta

(b) Imagem original.

Figura 4.12

Nesse ponto, é possível determinar se na imagem há presença de distorções. Isto é, calculado pelo número de linhas retas encontradas no universo $H_o(\theta,\rho)$. Caso exista um conjunto de linhas retas capazes de formar um retângulo, a imagem é classificada como "sem deformação". Esta classificação é realizada por dois critérios: O primeiro critério é assegurar que existam ao menos dois pares de retas de modo que um par tenha orientação vertical (90° de inclinação) e o outro horizontal (0° de inclinação). Portanto, sendo *nverticals* e *nhorizontals* o número de retas encontradas na vertical e na horizontal, respectivamente, obtemos a primeira condição (Equação 4.5).

$$nverticals > 1 \quad and \quad nhorizontals > 1 \tag{4.5}$$

A segunda condição analisa o aspecto do maior retângulo presente em $H_o(\theta,\rho)$. Primeiro, as retas de maior distância entre si para cada orientação são determinadas, sendo *v1* e *v2* as verticais e *h1* e *h2* horizontais. Após análise nas imagens de documentos utilizadas neste trabalho (Seção 5.1), verificou-se que, na maior parte dos exemplos, o percentual da área do documento ocupa entre 30% a 90% da área total da imagem.

Portanto, um retângulo é considerado válido se o tamanho dos segmentos encontrados é ao menos 25% do tamanho da sua componente correspondente. Esta condição está expressa nas Equações 4.6 e 4.7, onde *rows* é o número de linhas da imagem, *cols* é o número de colunas e *Dist(r1,r2)* é uma função que calcula a distância entre duas retas paralelas. Dessa forma, a utilização deste ponto de corte, evita que retângulos muito pequenos sejam interpretados como documentos.

$$Dist(v1,v2) > \frac{cols}{4} \tag{4.6}$$

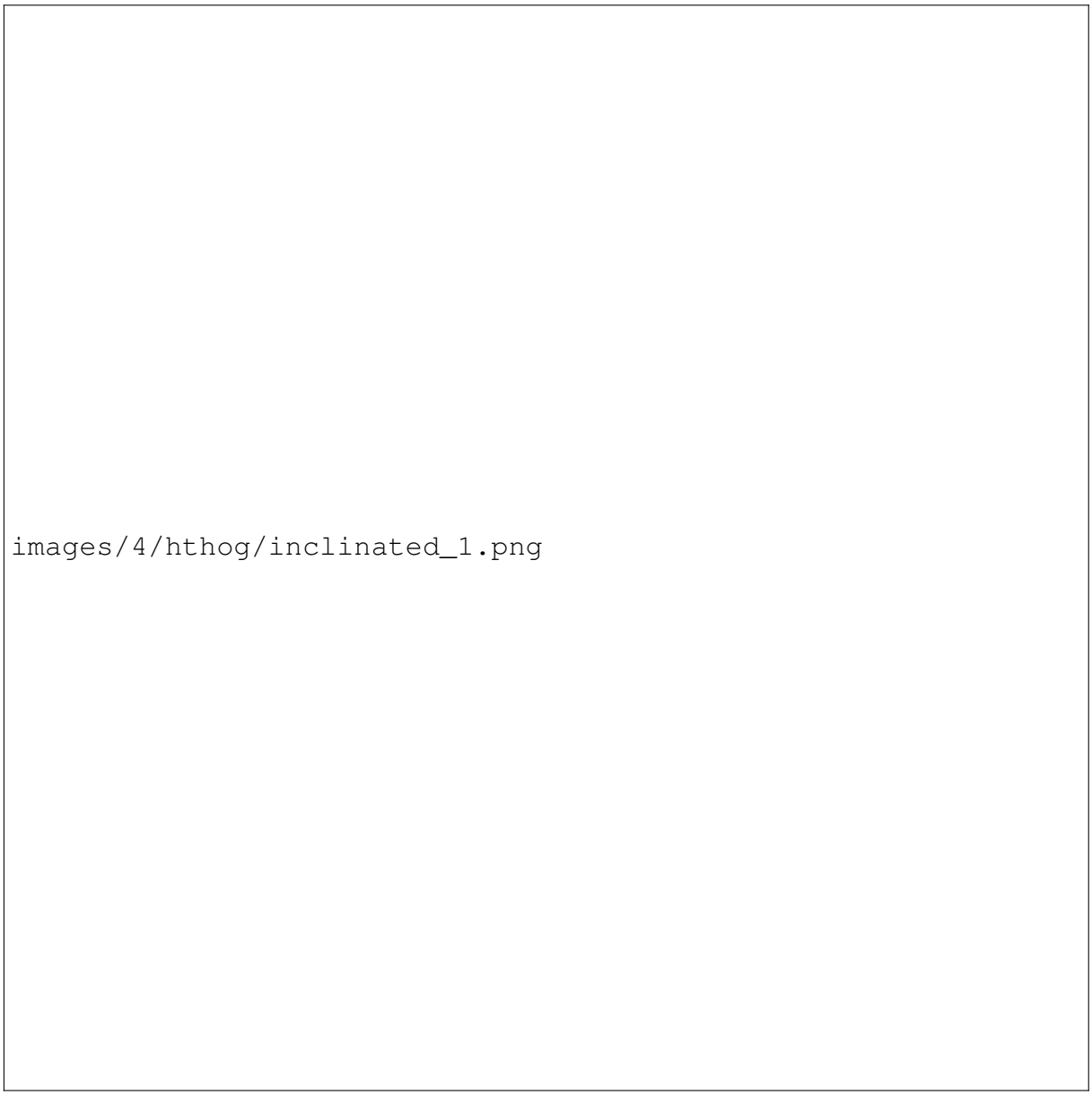
$$Dist(h1,h2) > \frac{rows}{4} \tag{4.7}$$

Quando determinado que não existem distorções no objeto presente na imagem, não existe necessidade da busca por componentes inclinados. Portanto a Seleção das Linhas (Seção 4.3) e, posteriormente, a Aplicação da Correção (Seção 4.4) já podem ser aplicadas na imagem. Por outro lado, quando estas condições não são satisfeitas, é necessário realizar uma análise das componentes inclinadas a fim de determinar quais as linhas que determinam os limites do documento na imagem.

Como mostra o Fluxograma da Figura 4.5, caso a imagem apresente distorção, uma nova análise dos HOGs é realizada, desta vez, considerando apenas os que apresentam inclinação (Seção 4.2.6). Consequentemente, uma nova limpeza de componentes é realizada para aplicação da HT nos HOGs inclinados. Este processo constrói um novo plano $H(\theta,\rho)$ que é utilizado, posteriormente na Seleção das Linhas (Seção 4.3).

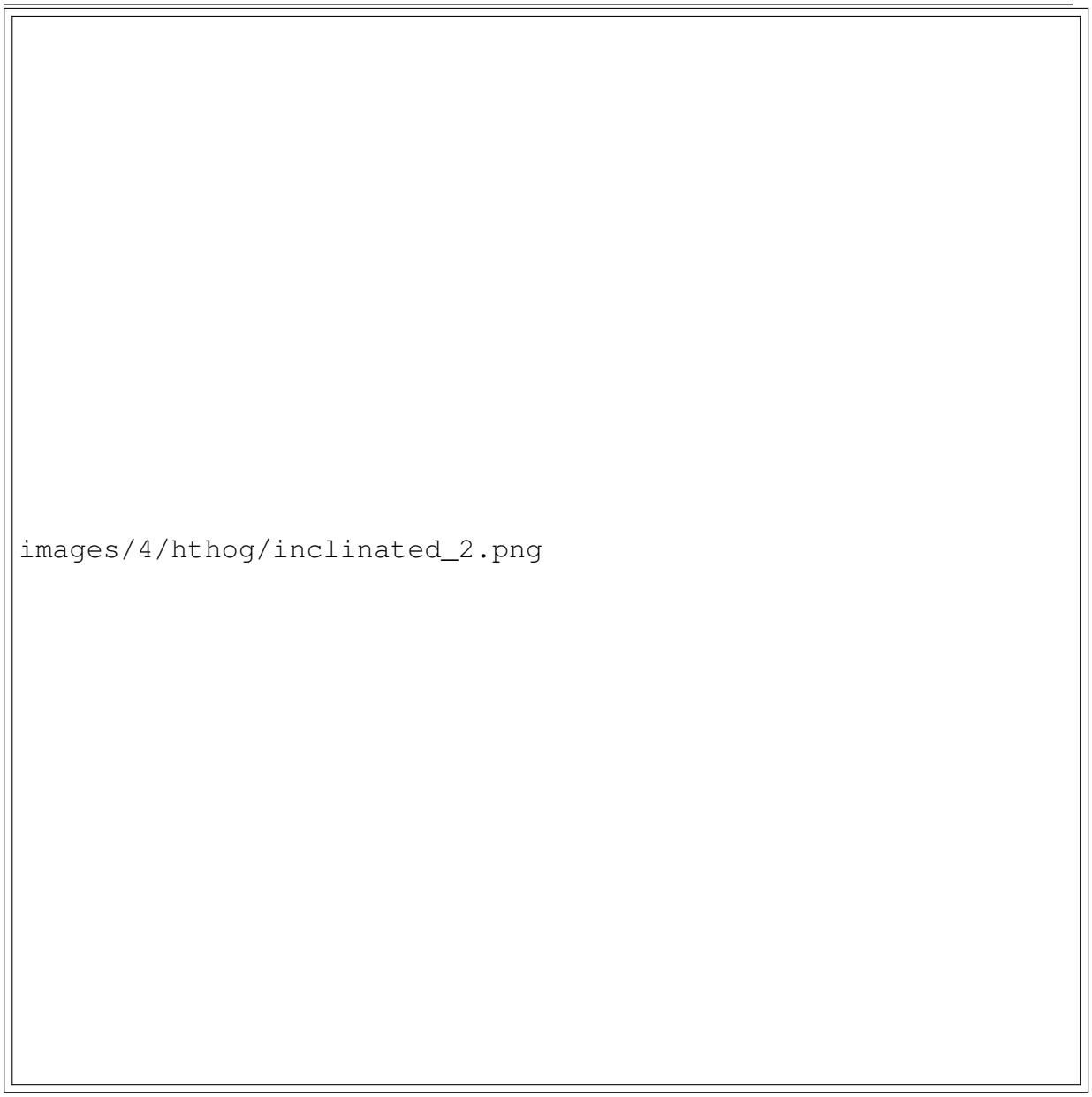
4.2.6 Seleção de HOGs com Inclinação

O processo realizado aqui é basicamente o mesmo mostrado na Seleção de HOGs sem Inclinação (Seção 4.2.2). A diferença é que são considerados apenas histogramas com orientação diferente de 0° e 90°. Portanto, apenas componentes que apresentem alguma inclinação em relação aos eixos horizontal e vertical são mantidos. O procedimento é idêntico ao detalhado no Algoritmo 2, apenas a condição de quais orientações são avaliadas é modificada. O resultado pode ser visto na Figura 4.13. Após este processo, a Limpeza de Componentes (Seção 4.2.3) é aplicada nos HOGs selecionados.



(a) Imagem original

Porém, a fim de diminuir a alta variabilidade presente em algumas regiões da imagem, um processo de normalização é aplicado nos HOGs com inclinação. Este procedimento tem como objetivo reforçar a direção dominante de determinada região. A Figura 4.14 mostra o efeito



(b) HOGs inclinados

Figura 4.13: Seleção de HOGs com Inclinação.

causado pela aplicação da normalização. Para cada HOG todas as orientações na vizinhança em uma janela *nxn* são coletadas. A nova orientação do HOG corrente é dada pela moda destas orientações.

Ao fim da normalização, o algoritmo da HT, detalhado na Seção 4.2.4, é aplicado nos HOGs inclinados. Portanto, para imagens com deformação, são gerados dois espaços de Hough: $H_o(\theta,\rho)$ e $H_i(\theta,\rho)$. O primeiro contém apenas representações de linhas com orientações 0° e 90°, ou seja, sem inclinação em relação aos eixos. Caso identificado que as linhas presentes em $H_o(\theta,\rho)$ não são suficientes para definir um retângulo, $H_i(\theta,\rho)$ é calculado utilizando os HOGs inclinados. Estes dois descritores são utilizados para identificar as linhas que descrevem o documento (Seção 4.3).

(a) HOGs originais **(b)** HOGs normalizados

Figura 4.14: Normalização dos HOGs usando uma janela 7x7

4.3 Seleção das Linhas

Esta seção explica de qual maneira as linhas retas que descrevem os limites do documento na imagem são encontradas. Para isto, pressupõe-se que o documento não contenha dobraduras e, portanto, pode ser descrito pelos cantos de um quadrilátero: C_{tl}, C_{tr}, C_{bl} e C_{br}. A tarefa de achar estes quatro cantos pode ser simplificada em achar as quatro linhas retas que descrevem o quadrilátero: esquerda L_{left}, direita L_{right}, topo L_{top} e base L_{bottom} (Figura 4.15). Para isto, o primeiro passo é realizar a identificação de todas as linhas descritas em H_o e H_i. Lembrar que se a imagem não apresentar distorções (Seção 4.2.5), apenas H_o é utilizado para identificação das linhas.

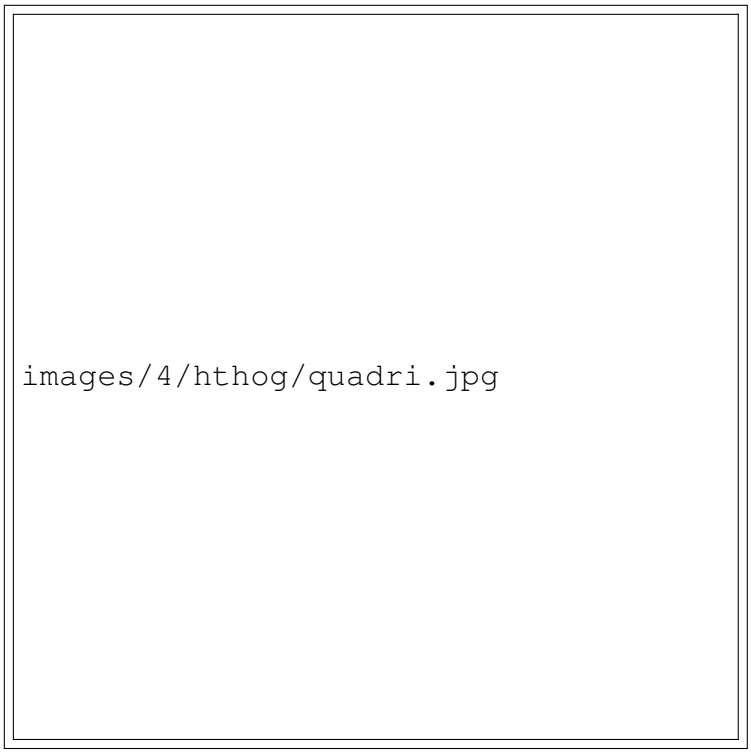


Figura 4.15: Superfície do documento descrita pelas linhas e cantos.

Para cada par (θ, ρ) ativo em H_o e H_i uma linha reta, $x = \alpha y + \beta$, é definida utilizando as Equações 2.7 e 4.4 formando o conjunto de retas L. Este conjunto de linhas retas é dividido em dois subconjuntos: linhas verticais L_v e linhas horizontais L_h. De L_v, é possível identificar as linhas retas à esquerda L_l e à direita do documento L_r, e conseqüentemente, as linhas de topo L_t e base L_b são definidas a partir de L_h. Por fim, de cada subconjunto, é possível definir as linhas que compõem os limites do documento: L_{left}, L_{right},L_{top} e L_{bottom}. O processo de seleção das linhas pode ser dividido em quatro etapas: identificação de L_v e L_h, identificação de L_l, L_r,L_t e L_b, remoção de *outliers* e identificação de L_{left}, L_{right},L_{top} e L_{bottom}.

4.3.1 Identificação de L_v e L_h

Neste procedimento, as diferenças de inclinação entre todas as linhas em L são calculadas. Estes valores são armazenados em D, que é uma matriz n x n, onde n é o número de linhas em L. D é definido de forma que $D(i, j)$ seja a diferença de inclinação entre a *i-ésima* e a *j-ésima* linhas. Desta forma, é possível definir L_v e L_h pelo procedimento apresentado no Algoritmo 4.

Algoritmo 4 Definição de L_v e L_h

```
1:  procedure VERTICALANDHORIZONTALLINESDEFINITION(L, D)
2:      [L1, L2] ← maxDiff(D)        ▷ L1 e L2 são as linhas com maior diferença de inclinação
3:      LM ← max(L1, L2)   ▷ LM é a maior inclinação relativa ao eixo horizontal entre L1 e L2
4:      Lm ← min(L1, L2)   ▷ Lm é a menor inclinação relativa ao eixo horizontal entre L1 e L2
5:      Lv(0) ← LM                  ▷ Lv e Lh são os grupos que são inicializados com Lm e LM
6:      Lh(0) ← Lm
7:      for i ← 1, n do
8:          if Li ≠ LM and Li ≠ Lm then
9:              Dv = CalcDissimilarity(Lv, Li)
10:             Dh = CalcDissimilarity(Lh, Li)
11:             if Dv < Dh then
12:                 Lv.append(Li)
13:             else
14:                 Lh.append(Li)
15:             end if
16:         end if
17:     end for
18: end procedure
```

Este algoritmo agrupa as linhas em L em dois grupos: L_v e L_h. Cada linha L_i é avaliada quanto ao grau de dissimilaridade calculado por *CalcDissimilarity*. A dissimilaridade é a diferença entre a inclinação de L_i e a média das inclinações das linhas em um agrupamento L_x. L_i é atribuído ao grupo que obtiver a menor dissimilaridade.

4.3.2 Identificação de L_l, L_r, L_t e L_b

Neste passo, os agrupamentos vertical e horizontal são subdivididos. As linhas à esquerda (L_l) e à direita (L_r) são provenientes de L_v, enquanto, linhas de topo L_t e base L_b são extraídas de L_h. Este agrupamento é realizado com base nos pontos médios de cada linha. Neste contexto, seja *rows* e *cols* o número de linhas e colunas da imagem, respectivamente, o ponto médio $P_i(x,y)$ de uma linha L_i pode ser definido pela Equação 4.8, se $L_i \in L_v$ ou pela Equação 4.9, se $L_i \in L_h$.

$$P_i(x,y) = \begin{cases} x = \frac{rows}{2} \\ y = \frac{x - L_i.\beta}{L_i.\alpha} \end{cases} \tag{4.8}$$

$$P_i(x,y) = \begin{cases} x = L_i.\alpha * y + L_i.\beta \\ y = \frac{cols}{2} \end{cases} \tag{4.9}$$

Para determinar L_l e L_r, os pontos médio de cada linha em L_v são armazenados em P_v. Depois, a distância euclidiana entre cada um dos pontos contidos em P_v é calculado e armazenado em E_v que é uma matriz *m x m*, onde *m* é o número de pontos em P_v. E_v é definido de forma que $E_v(i,j)$ seja a distância euclidiana entre o *i-ésimo* e a *j-ésimo* pontos. Com estas informações é possível definir L_l e L_r pelo método descrito no Algoritmo 5.

Algoritmo 5 Definição de L_l e L_r

```
1:  procedure SIDELINESDEFINITION(L_v, E_v)
2:      [P_1, P_2] ← maxDist(E_v)          ▷ P_1 e P_2 são os pontos com maior distância entre si
3:      L_left ← getLeft(P_1, P_2)         ▷ L_left contém o ponto mais à esquerda entre P_1 e P_2
4:      L_right ← getRight(P_1, P_2)       ▷ L_right contém o ponto mais à direita entre P_1 e P_2
5:      L_l(0) ← L_left          ▷ L_l e L_r são os grupos que são inicializados com L_left e L_right
6:      L_r(0) ← L_right
7:      for i ← 1, m do
8:          if L_i ≠ L_left and L_i ≠ L_right then
9:              D_l = CalcDist(L_left, L_i)
10:             D_r = CalcDist(L_right, L_i)
11:             if D_l < D_r then
12:                 L_l.append(L_i)
13:             else
14:                 L_r.append(L_i)
15:             end if
16:         end if
17:     end for
18: end procedure
```

CalcDist calcula a distância euclidiana entre os pontos médios de duas linhas. Por este mesmo método descrito pelo Algoritmo 5 é possível determinar L_t e L_b a partir de L_h e E_h. A principal mudança é utilizar *getTop* e *getBottom* ao invés de *getLeft* e *getRight*. Ao término deste processo, os subconjuntos L_l, L_r, L_t e L_b estão definidos.

4.3.3 Remoção de *outliers*

Devido a ruídos causados por *backgrounds* complexos (não uniformes), muitas linhas que não fazem parte da composição do documento são adicionadas aos agrupamentos de linha. Este comportamento pode ser observado na Figura 4.16. A remoção destes *outliers* é realizada baseada na média (μ) e desvio padrão (σ) das inclinações das linhas do agrupamento. Para cada agrupamento L_x, as linhas com inclinação (θ) que não satisfazem a condição descrita pela Equação 4.10 são removidas do agrupamento.

$$\mu - \sigma < \theta < \mu + \sigma \tag{4.10}$$

(a) Amostra da Imagem original.

(b) HOGs inclinados.

(c) Linha fora do padrão em vermelho.

Figura 4.16: Remoção de *outlier*.

4.3.4 Identificação de L_{left}, L_{right},L_{top} e L_{bottom}

Após a remoção dos *outliers* é possível afirmar que as linhas presentes em L_l, L_r, L_t e L_b são as linhas que compõem o documento. Importante lembrar que a finalidade principal deste trabalho é melhorar a legibilidade do texto, portanto a maior área possível – que seja menos suscetível a perda de informações – dentro do universo de linha disponível é procurada. Por isto, para definir a superfície que delimita o objeto (Figura 4.15), é preciso encontrar as linhas mais próximas as margens da imagem.

As quatro margens podem ser descritas por funções do tipo $x = \alpha y + \beta$. Portanto, a margem esquerda (M_l) pode ser descrita por $y = 0$ e a margem direita (M_r) por $y = cols$. Da mesma forma, a linha de topo (M_t) é $x = 0$ e a linha de base (M_b) é $x = rows$. As linhas mais próximas às margens são encontradas a partir dos pontos de interseção entre as linhas de orientações diferentes.

Portanto, para encontrar L_{right} é necessário avaliar todos os pontos de interseção (I_r) entre L_r e as linhas de orientação contrária: L_t e L_b (Figura 4.17). A distância euclidiana entre cada ponto em I_r e M_r é calculado. Seja I_{rmin} o ponto de menor distância à M_r, então, L_{right} (Figura 4.18) é a linha na qual I_{rmin} pertence. Em alguns casos, I_{rmin} pode pertencer a mais de uma linha, como na presença de uma dobradura; neste caso, a primeira ocorrência é considerada. L_{left},L_{top} e L_{bottom} são obtidas pelo mesmo princípio.

4.4 Aplicação da Correção

A partir das quatro linhas que compõem a superfície distorcida do documento, o quadrilátero $C_{tl}C_{tr}C_{bl}C_{br}$ é definido a partir das intersecções dessas linhas (Figura 4.19). Como explicado na Seção 2.3.2, a correção da imagem é o mapeamento dos pixels na superfície distorcida para uma superfície corrigida. Este mapeamento é dado pela Equação 4.11, onde λ é a matriz da transformação, e X e x são as coordenadas original e corrigida, respectivamente.

$$X\lambda = x \tag{4.11}$$

$$\lambda = (X^T X)^{-1} X^T x \tag{4.12}$$

Portanto, λ pode ser calculado a partir da Equação 4.11 rearranjada em 4.12, onde X é definido pelo quadrilátero distorcido $C_{tl}C_{tr}C_{bl}C_{br}$ e x é definido pela superfície planificada $P_{tl}P_{tr}P_{bl}P_{br}$ com $2W$ de largura e $2H$ de altura. Os valores de W e H são definidos pelo maior comprimento encontrado na horizontal e vertical da superfície distorcida, como ilustrado na Figura 4.20a. Essa decisão foi tomada para evitar aumento no tempo de processamento. Uma vez calculada a matriz de transformação λ, o mapeamento $X \rightarrow x$ pode ser feito para todos os *pixels* (Figura 4.20b).

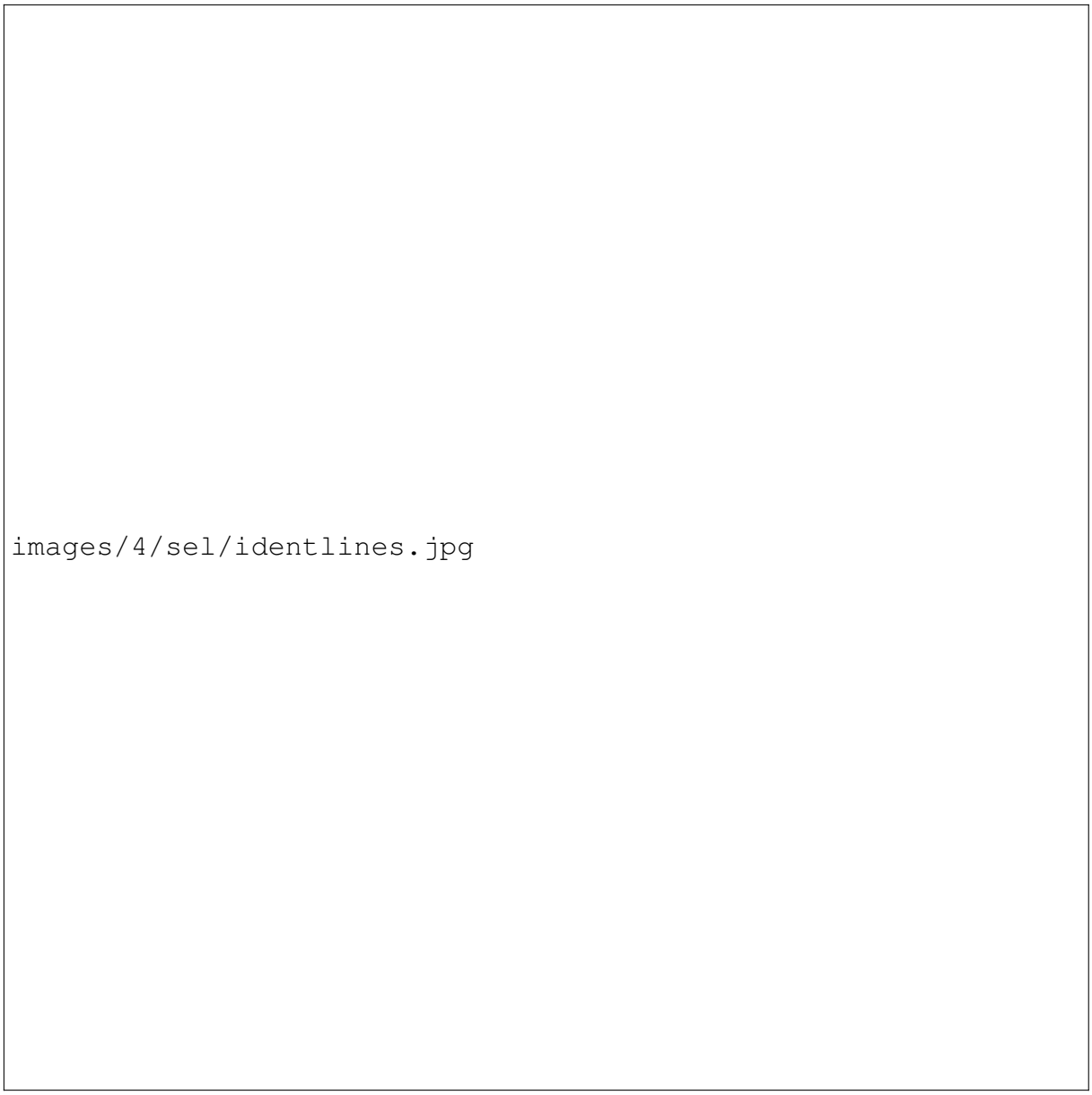
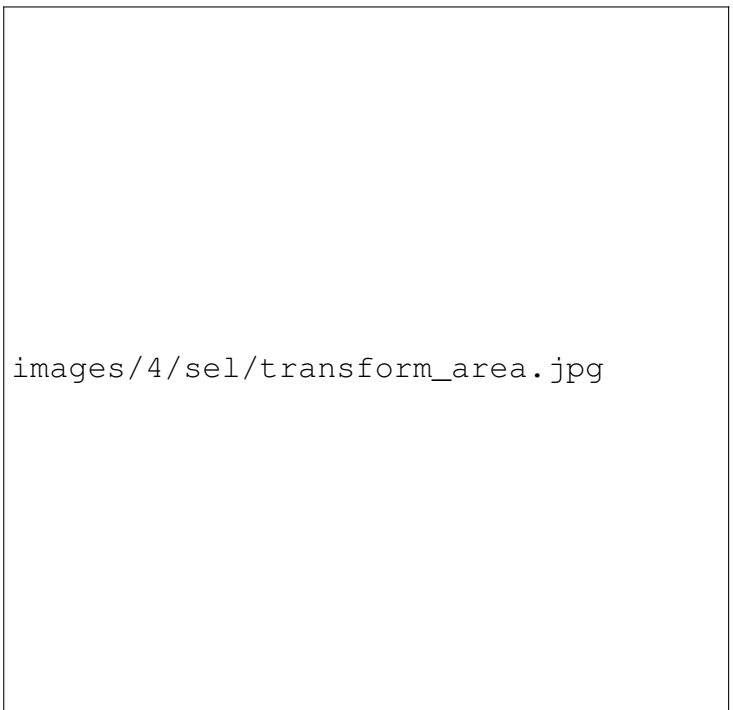


Figura 4.17: Margens, interseções e linhas candidatas.

Porém, este mapeamento não pode ser realizado de maneira "discreta", ou seja, percorrendo i e j tal que $i \rightarrow 1,2,...,nrows$ e $j \rightarrow 1,2,...,ncols$, onde $nrows$ é o número de linhas e $ncols$ é o número de colunas. Isto se deve ao fato de a transformação $X\lambda = x$ não contemplar todos os *pixels* na superfície retificada, causando o surgimento de regiões não mapeadas no resultado final. Para fins práticos, a solução para isto é percorrer a imagem com um valor de incremento menor que 1 (um). Portanto, a superfície deve ser percorrida utilizando a função $f(i) = i \times n$, onde $n < 1$ e $i \rightarrow 1,2,...,nrows$ para índices de linha e $i \rightarrow 1,2,...,ncols$ para índices de coluna. Para este trabalho foi utilizando $n = 0,2$.

Figura 4.18: Escolha de L_{right} a partir de menor distância para M_r.



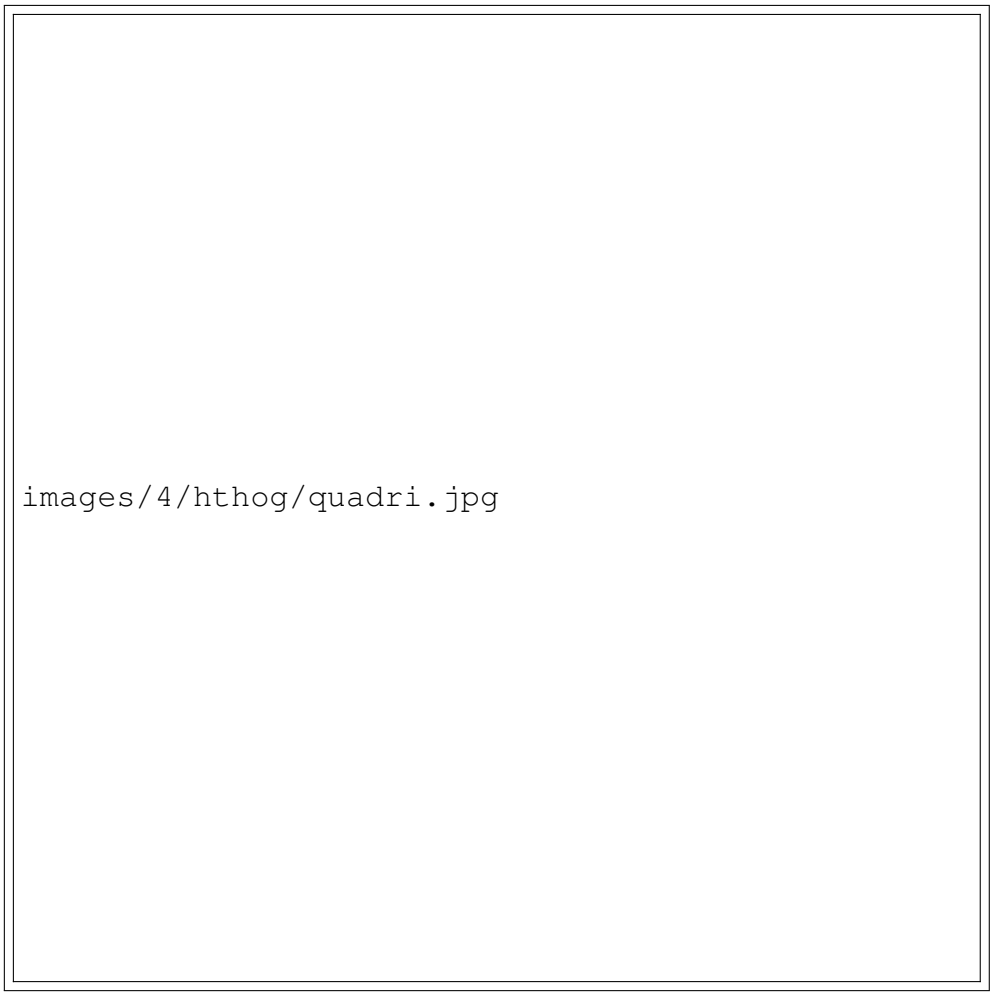


Figura 4.19: Superfície do documento descrita pelas linhas e cantos.

Como a abordagem proposta procura as linhas que definem os documentos para depois definir os cantos, a superfície distorcida é definida mesmo que algum canto esteja fora do domínio da imagem. Os experimentos e resultados da aplicação do algoritmo proposto são abordados no Capítulo 5.

5

Experimentos e Análise de Resultados

Este Capítulo apresenta os experimentos, metodologias e os resultados obtidos para a técnica proposta. As seções seguintes abordam: a base utilizada nos experimentos, o método de escolha dos parâmetros livres, implementação e os critérios de avaliação. Ao fim do capítulo, uma análise à luz dos resultados é realizada.

5.1 Base de Dados

Os Benchmarks utilizados para verificação de técnicas de *dewarping* são fortemente baseados em documentos ricos em texto tais como páginas de livros (**?**). Por isto, para testar a assertividade do método proposto foi necessário capturar uma base de imagens de documentos. As imagens foram capturadas por dispositivos móveis em ambientes sem controle de iluminação e sem a utilização de suporte para câmera.

Modelo	Fabricante	*Megapixels*	N. Imagens
Moto G	Motorola	5	15
Nexus 4	LG	8	3
Lumia 710	Nokia	5	15
Tab 3 T311	Samsung	5	4
Xperia P	Sony	8	3

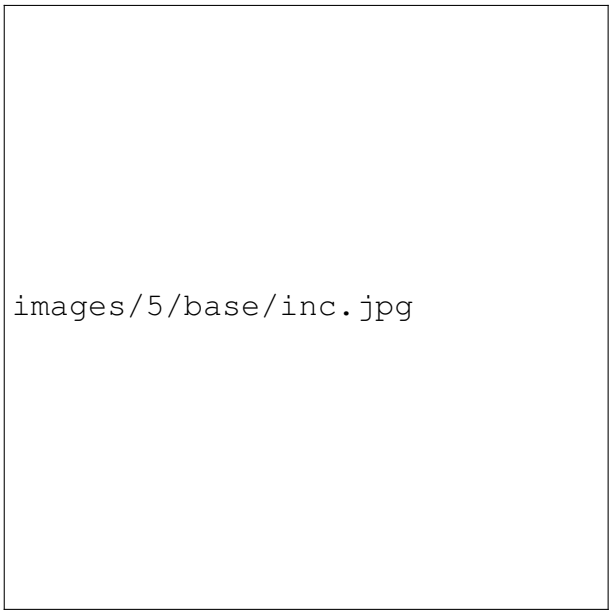
Tabela 5.1: Distribuição da quantidade de imagens pelos modelos de dispositivo.

A Tabela 5.1 mostra os dispositivos utilizados, resolução de câmera e a quantidade de imagens correspondentes. Estes são modelos populares de *smartphones* e *tablets* munidos de câmeras com resolução mediana. No mercado, existem aparelhos móveis com sensores de até 41 *megapixels*, como o Lumia 1020 da *Microsoft*.

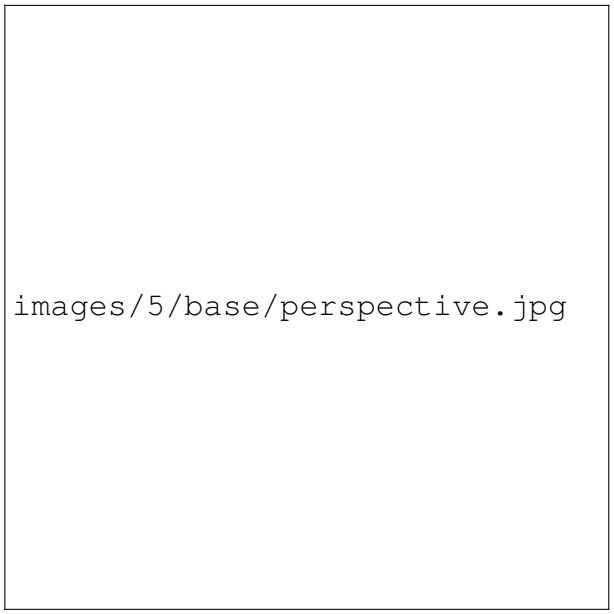
Característica	N. Imagens	Percentual
Inclinação	24	60%
Perspectiva	19	47,5%
Canto Invisível	9	22,5%
Ausência de Deformação	12	30%

Tabela 5.2: Características da Base.

A Tabela 5.2 apresenta algumas características das imagens: 60% das imagens apresentam inclinação. Isto é, o documento está rotacionado em relação aos eixos horizontal e vertical (Figura 5.1a). Além da rotação, a imagem pode apresentar distorção em profundidade causando o efeito de perspectiva (Figura 5.1b), o que ocorre para 47,5% das imagens. Outro efeito colateral de captura de imagens por dispositivos móveis é o de cantos fora do domínio visível da imagem (Figura 5.1c). Para 30% das imagens, nenhuma destas duas deformações ocorrem. Todas as imagens estão no formato *Joint Photographic Experts Group* (JPEG), que é o formato padrão comum de exportação de todos os dispositivos utilizados.

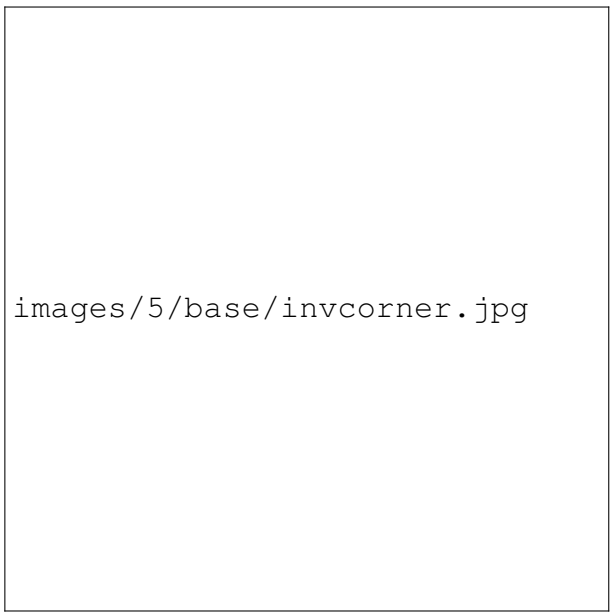


(a) Imagem com inclinação.



(b) Imagem com distorção de perspectiva.

Figura 5.1: Hogs da Amostra



(c) Cantos fora do domínio visível da Imagem.

Figura 5.1: Características das Imagens da Base.

5.2 Metodologia dos Experimentos

O critério para avaliação do método é a distância dos cantos encontrados pelo algoritmo proposto aos cantos reais da imagem. As coordenadas dos cantos reais das imagens, também chamado de *ground truth* da base, foram obtidas através de percepção humana como explicado na Seção 5.2.1. O método do cálculo do erro está detalhado na Seção 5.2.2. Para encontrar a melhor

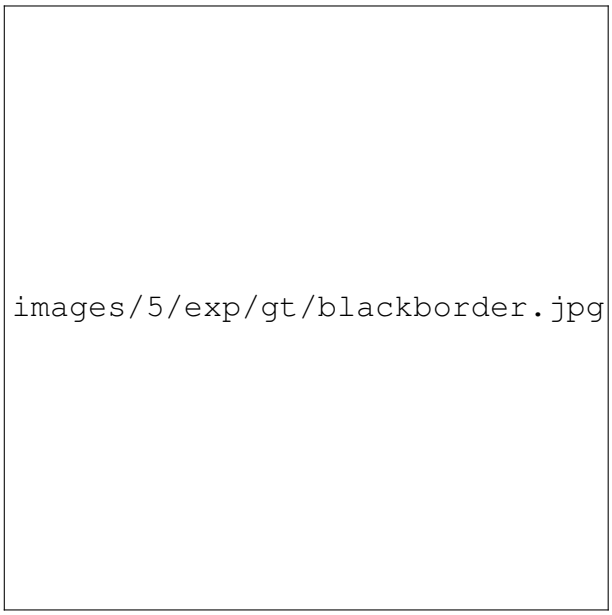
parametrização do algoritmo, várias configurações de *numBins* e *cellSizes* foram avaliadas, como explicado na Seção 5.2.3.

5.2.1 Definição do *Ground Truth*

Para evitar o enviesamento do *ground truth*, quatro pessoas, sem relação com este trabalho, fizeram a identificação dos cantos das imagens manualmente. Apenas informações necessárias para marcação dos cantos serviram como guia:

- Marcar os quatro cantos do documento;
- Caso o canto não esteja no domínio visível, marcar no local provável, fora do domínio da imagem; e
- Caso o canto seja arredondado, marcar no encontro imaginário das margens do documento.

A determinação dos cantos ficou à livre interpretação dos voluntários. Para que fosse possível a marcação de pontos fora do domínio visível, foram adicionados 300 pixels na cor preta à cada margem da imagem (Figura 5.2a). Um exemplo de marcação de borda arredondada está ilustrado na Figura 5.2b. Nestas imagens é possível verificar marcas vermelhas referente à determinação dos cantos pelos voluntários.



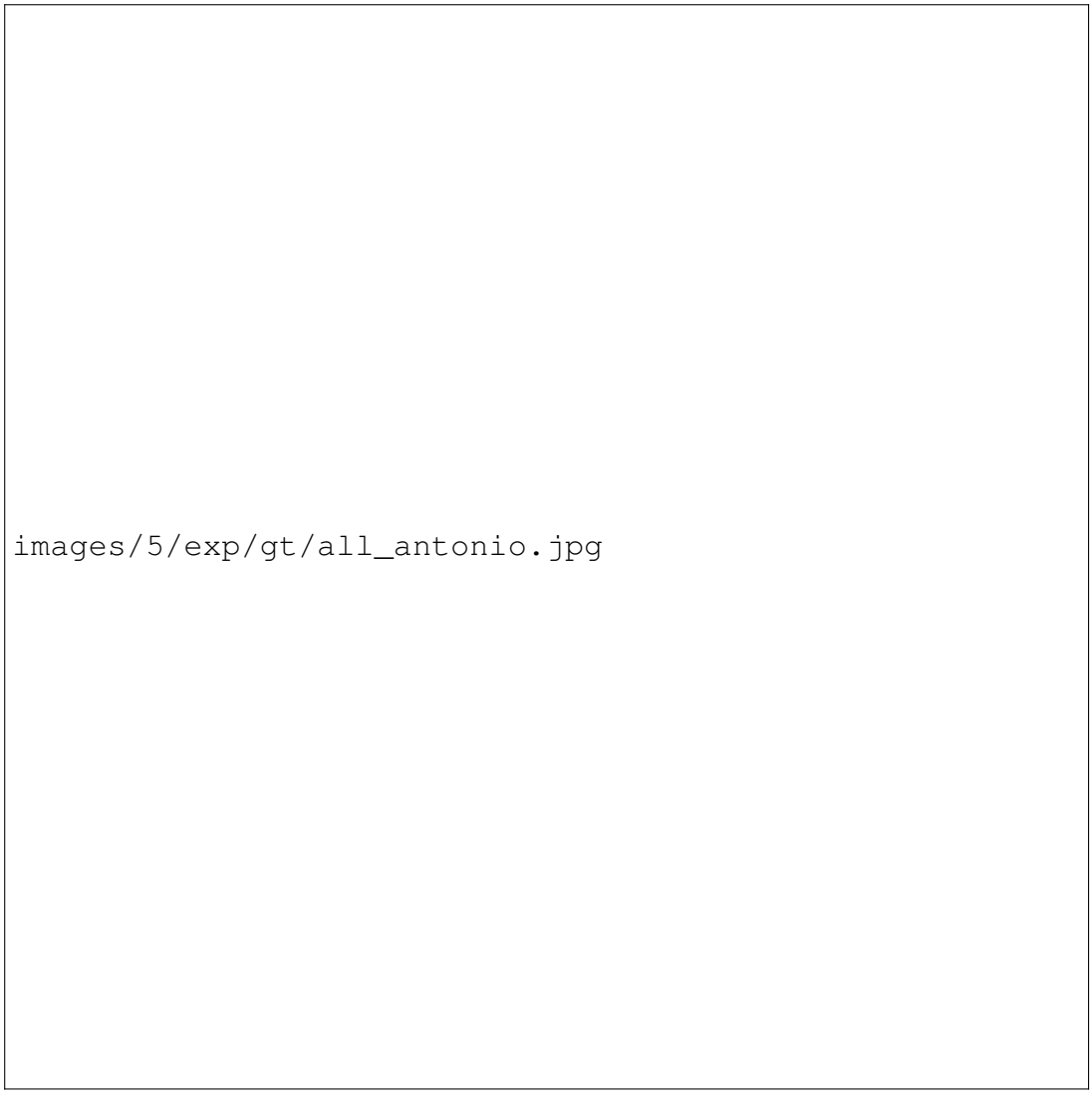
(a) Margens em preto para auxiliar marcação de cantos fora do domínio da imagem.

(b) Marcação de canto arredondado.

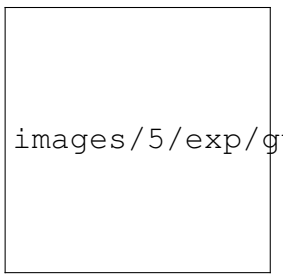
Figura 5.2: Exemplo de Imagens do *ground truth*

A Figura 5.3 mostra a semelhança na marcação dos pontos pelos voluntários em uma imagem. Os pontos nas Figuras 5.3b, 5.3c, 5.3d e 5.3e estão representados em quatro cores

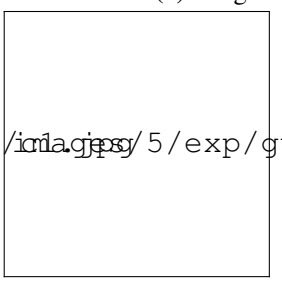
diferentes – uma cor por indivíduo. Em geral, a média da marcação dos pontos foi bem próxima o desvio padrão absoluto para os quatro cantos foi 39 *pixels*. Após a determinação dos quatro cantos de cada imagem, uma rotina guarda suas coordenadas correspondentes: C'_{tl}, C'_{tr}, C'_{bl} e C'_{br}.



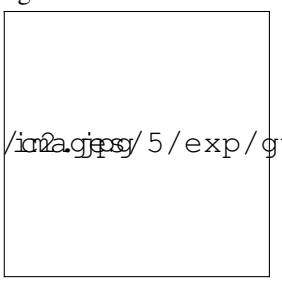
(a) Imagem original.



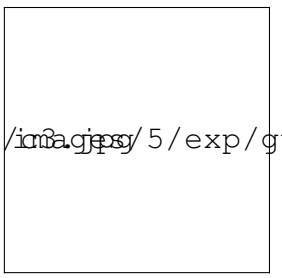
(b) C'_{tl}



(c) C'_{bl}



(d) C'_{br}



(e) C'_{tr}

Figura 5.3: Marcação de C'_{tl}, C'_{tr}, C'_{bl} e C'_{br} pelos quatro voluntários.

5.2.2 Cálculo do Erro ($E_{\%}$)

O erro ($E_{\%}$) é dado pela distância euclidiana entre os cantos no *ground truth* (C_i') e os cantos encontrados pelo método proposto (C_i). Sejam (x', y') e (x, y) as coordenadas de C_i' e C_i, respectivamente, a distância é calculada pela fórmula da Equação 5.1. O erro percentual $E_{\%}$ é encontrado a partir da distância normalizada local. Isto é, a diferença é dividida pelo tamanho da dimensão correspondente, onde *rows* é o número de linhas e *cols* é o número de colunas da imagem (Equação 5.2).

$$E_a = \sqrt{(x' - x)^2 + (y' - y)^2} \tag{5.1}$$

$$E_{\%} = \sqrt{\left(\frac{x' - x}{rows}\right)^2 + \left(\frac{y' - y}{cols}\right)^2} \times \frac{1}{\sqrt{2}} \times 100 \tag{5.2}$$

A distância normalizada é necessária porque as dimensões da imagem têm valores diferentes ($cols \neq rows$). Por este motivo, a diferença das componentes na vertical e na horizontal não tem mesmo peso. A divisão das diferenças pelo tamanho da dimensão correspondente, desfaz esta incongruência. Para garantir que o valor de $E_{\%}$ esteja contido no intervalo $[0\%, 100\%]$ é necessário uma divisão por $\sqrt{2}$ e multiplicação por 100.

5.2.3 Parametrização do Método Proposto

O método proposto possui dois parâmetros livres: *numBins* e *cellSize*. O *numBins* define o número de orientações que um HOG pode assumir, enquanto *cellSize* determina o tamanho da janela na qual a imagem é dividida. O objetivo é encontrar valores ótimos destes parâmetros de forma que minimizem $E_{\%}$. Para isto, é necessário testar diversas configurações destes parâmetros.

O *numBins* é um parâmetro sensível, pois, a cada incremento, uma nova configuração de orientações é determinada. Por isto, uma função linear foi definida para determinar sua variação (Equação 5.3). Por outro lado, *CellSize* é medido em *pixels*, portanto, é necessário uma função de variação que forneça uma maior diferença entre os valores que uma função linear, no caso, uma exponencial (Equação 5.4).

$$numBins(n) = 2n \tag{5.3}$$

$$cellSize(m) = 2^m \tag{5.4}$$

5.3 Implementações

Todos os algoritmos deste trabalho foram implementados no *software* MATLAB (**?**). Este ferramenta provê uma suite de desenvolvimento em processamento de imagens e reconhecimento

de padrões bastante abrangente. Soma-se a isto, a facilidade de uso e a vasta documentação *online*. Por estes motivos, várias soluções científicas são disponibilizadas em MATLAB, facilitando melhorias e integração com outras aplicações.

Para calcular os HOGs das imagens, este trabalho utilizou a biblioteca de código aberto VLFeat (**?**). Esta biblioteca implementa vários algoritmos de visão computacional e extração de características. A interface do descritor de HOG é bastante flexível, o que permite a simulação de vários cenários, como os mostrados na Seção 5.2.3. Além disto, é possível criar versões renderizadas dos HOGs para melhor compreensão de sua manipulação. Esta *feature* permitiu o uso das várias imagem dos HOGs neste trabalho, fornecendo uma abstração visual dos descritores.

5.4 Resultados e Análise

Nesta seção, os resultados obtidos pelo método proposto são apresentados, descritos e analisados. Uma análise geral dos resultados para todas as configurações testadas está presente na Seção 5.4.1. A partir dos resultados mais significativos, uma análise qualitativa é realizada na Seção 5.4.2. Os resultados para reconhecimento do texto nas imagens originais e retificadas estão na Seção 5.4.3.

5.4.1 Parametrizações

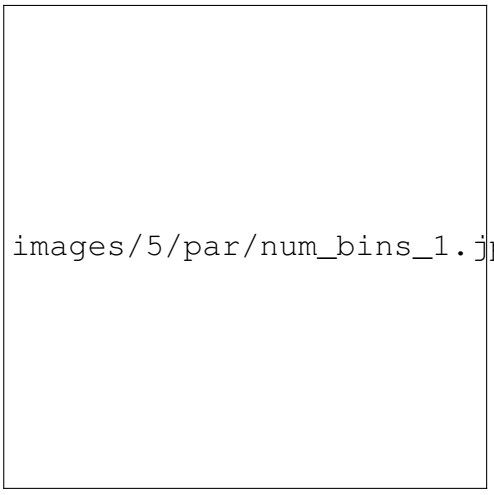
A Tabela 5.3 traz os valores dos erros ($E_{\%}$) para $numBins = 2n, n \longrightarrow 3,4,\ldots,15$ e $cellSize = 2^m, m \longrightarrow 3,4,5$. Estes *ranges* foram escolhidos de forma a contemplar o espaço de busca coerentes com a proposta. Valores muito altos podem produzir muita informação, removendo a função de "extração de característica"do descritor e adicionando uma alta variabilidade de informação, por outro lado, valores pequenos podem não produzir informação suficiente. A coluna $L_{\%}$ é o percentual de imagens que não obtiveram resultado representativo, ou seja, não foram encontrados valores suficientes para determinar os quatro cantos.

Analisando a Tabela 5.3, é possível dividi-la em duas partes, considerando os valores de $L_{\%}$ e $E_{\%}$. A primeira metade, com *numBins* menores, apresenta uma quantidade expressiva de imagens não aproveitadas. Isto se deve a distribuição dos valores no histograma, que é dividido em *numBins* partes. Neste cenário, o uso do *limiar* máximo não atinge o seu objetivo de filtrar componentes indesejados, criando uma quantidade muito grande de ruído. É possível constatar isto na Figura 5.4b.

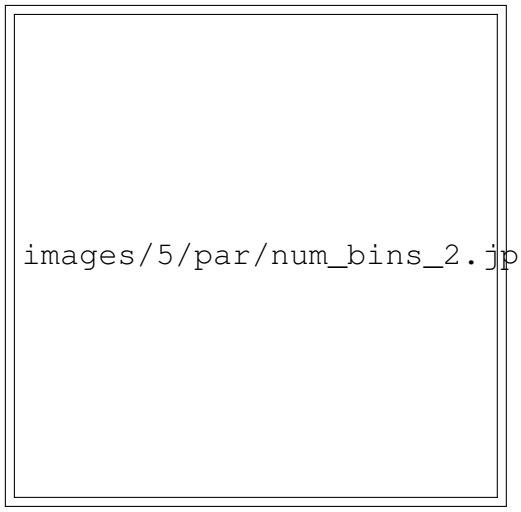
Da mesma forma, valores altos de *numBins* apresentam $L_{\%}$ expressivos, porém com variação mais suave. Neste caso, os HOGs resultantes da filtragem não obtiveram uma boa normalização. A maior variabilidade entre as direções presentes deteriora a busca pela orientação correta, processo detalhado na Seção 4.2.6. A Figura 5.4c ilustra isto.

	cellSize					
		8		16		32	
		$E_{\%}$	$L_{\%}$	$E_{\%}$	$L_{\%}$	$E_{\%}$	$L_{\%}$
numBins	4	-	100,00	-	100,00	24,72	95,00
	6	21,08	90,00	35,85	82,50	26,19	95,00
	8	25,43	80,00	22,26	82,50	21,36	90,00
	10	25,53	77,50	23,85	82,50	25,55	82,50
	12	30,00	70,00	31,68	80,00	27,18	77,50
	14	26,08	70,00	23,85	72,50	18,30	65,00
	16	15,90	67,50	14,97	55,00	11,00	45,00
	18	9,12	17,50	4,08	0,00	6,88	42,50
	20	8,65	17,50	4,93	7,50	7,09	45,00
	22	8,98	22,50	5,16	7,50	8,44	45,00
	24	12,17	32,50	5,83	10,00	8,71	40,00
	26	9,41	25,00	6,36	20,00	8,30	40,00
	28	10,24	30,00	7,17	22,50	7,87	40,00
	30	13,48	30,00	7,95	22,50	10,49	37,50

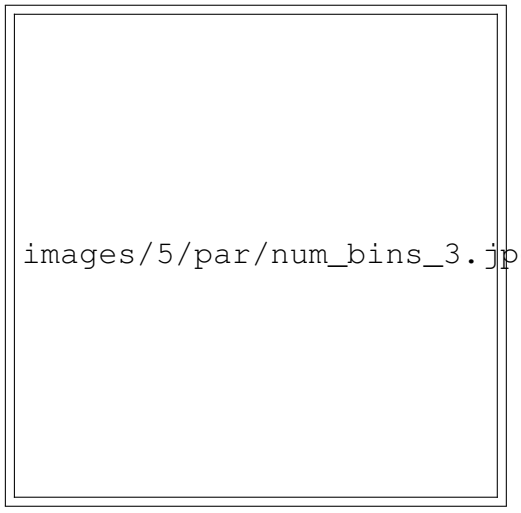
Tabela 5.3: $E_{\%}$ e $L_{\%}$ para vários valores de *cellSize* e *numBins*



(a) Amostra original.



(b) HOGs da imagem para *numBins* = 4. Alta densidade de histogramas não filtrados.



(c) HOGs da imagem para *numBins* = 30. Grande variação nas bordas do documento.

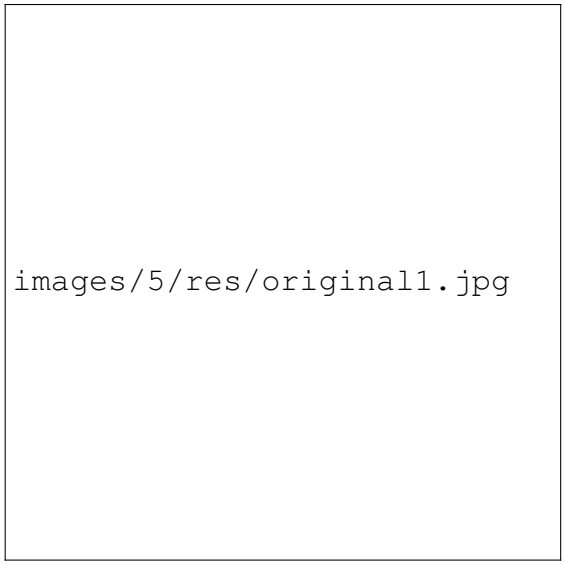
Figura 5.4: Amostra de imagens para *numBins* diferentes.

Os melhores resultados para $L_{\%}$ e $E_{\%}$ foram encontrados para valores intermediários da Tabela 5.3. Especificamente, a configuração $numBins = 18$ e $cellSize = 16$ obteve $E_{\%} = 4,08\%$ e $L_{\%} = 0\%$. Os resultados da retificação para esta parametrização são analisados na Seção 5.4.2.

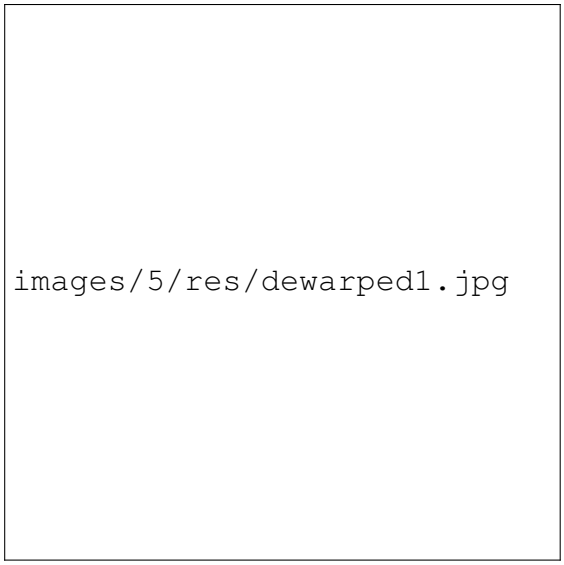
5.4.2 Retificação da Imagem

Nesta seção, uma análise qualitativa do resultado do *dewarping* é realizada. As imagens utilizadas nesta seção foram obtidas pela melhor configuração encontrada, onde $numBins = 18$ e $cellSize = 16$. É importante ressaltar que o objetivo deste trabalho é corrigir a imagem a fim de melhorar a legibilidade dos documentos. Por isto, eventuais discrepâncias entre os limites dos documentos originais e encontrados são considerados aceitáveis, desde que o conteúdo textual seja mantido. Uma análise quantitativa deste critério é feita na Seção 5.4.3.

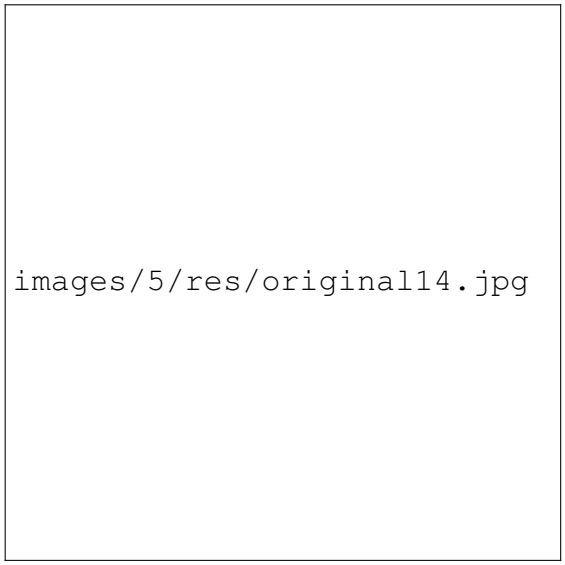
Na base de dados utilizada, 47,5% das imagens sofrem de distorção de perspectiva. Por exemplo, as imagens de documentos presentes nas Figuras 5.5g, 5.5i e 5.5k sofrem de uma distorção moderada, enquanto as Figuras 5.5a, 5.5e e 5.5c apresentam uma profundidade mais evidente. Em ambos os casos, o conteúdo textual é preservado após a aplicação da correção. O documento da Figura 5.5k, por sua vez, apresenta uma dobradura dificultando a determinação do quadrilátero ótimo (Seção 4.3). Porém, as informações textuais do documento não são perdidas.



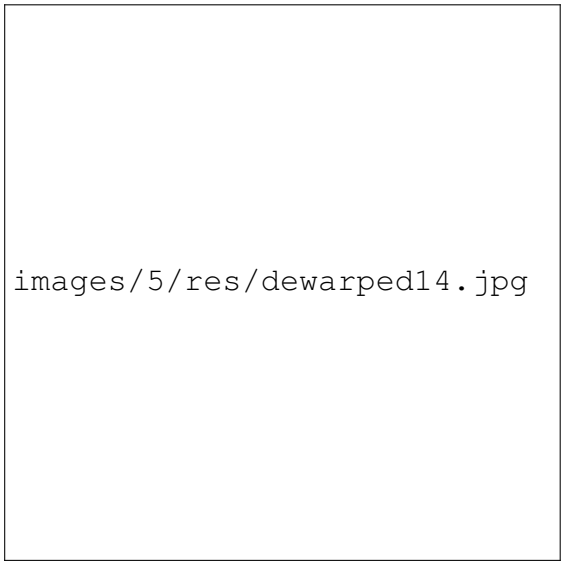
(a) Imagem Original.



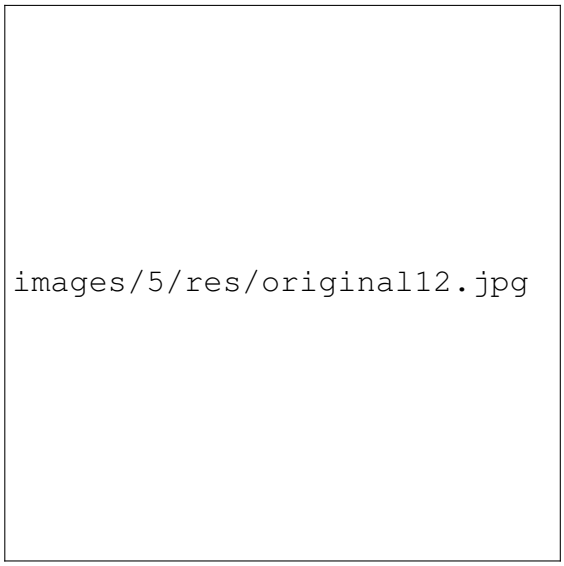
(b) Imagem Corrigida.



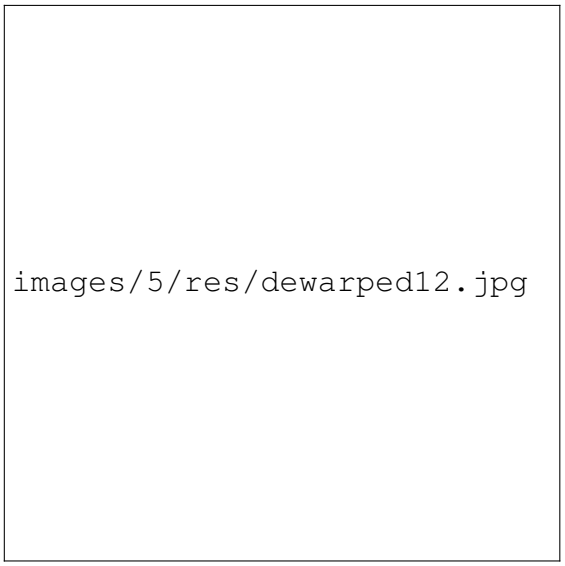
(c) Imagem Original.



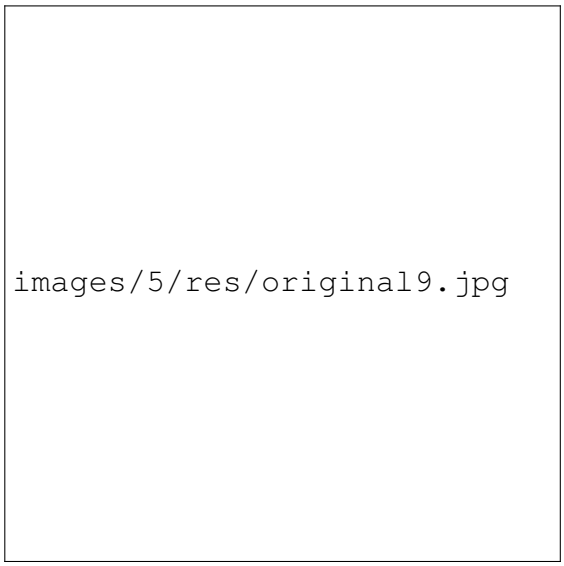
(d) Imagem Corrigida.



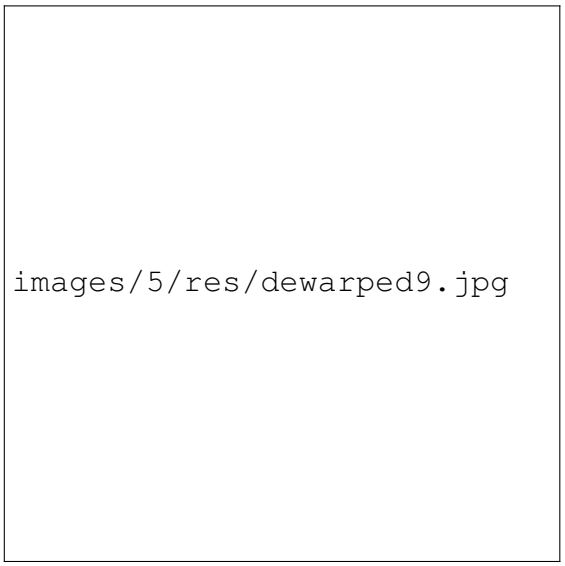
(e) Imagem Original.



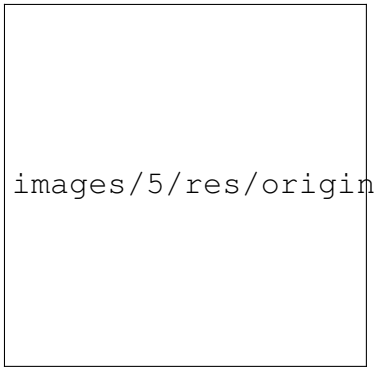
(f) Imagem Corrigida.



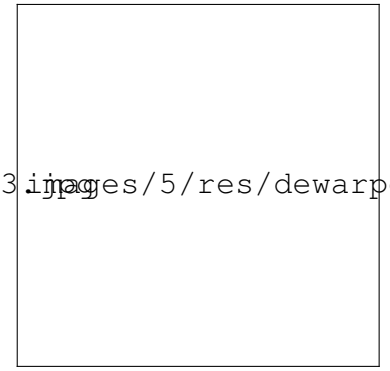
(g) Imagem Original.



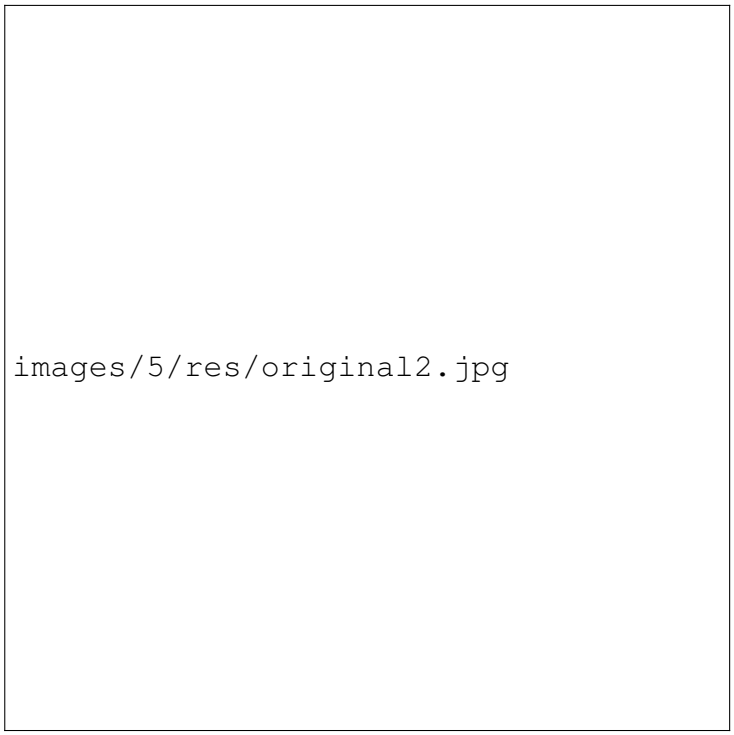
(h) Imagem Corrigida.



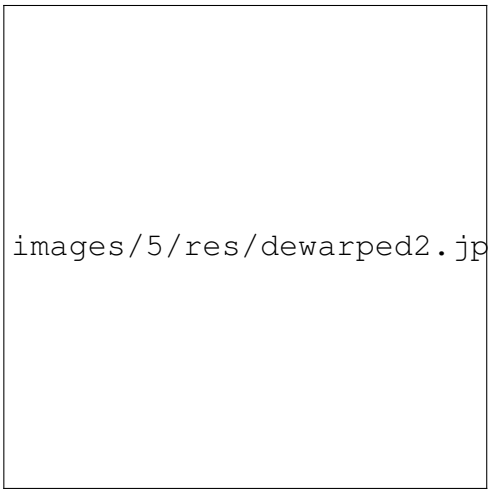
(i) Imagem Original.



(j) Imagem Corrigida.



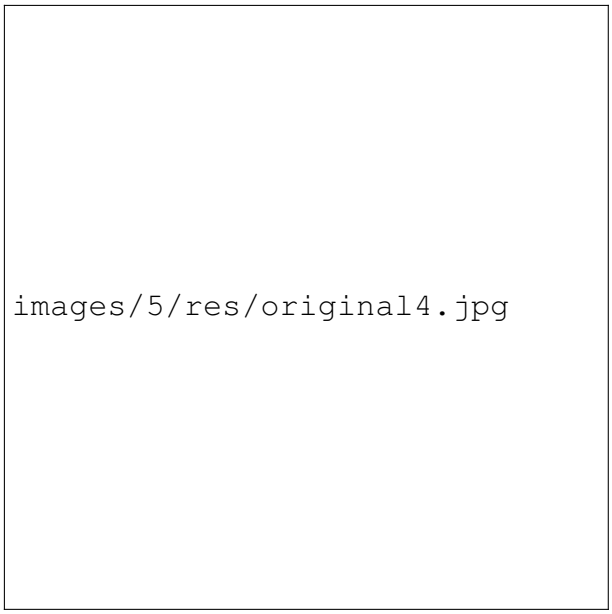
(k) Imagem Original.



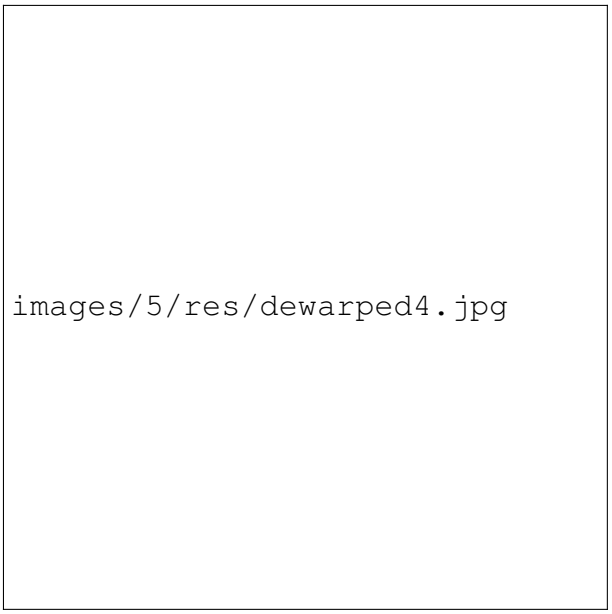
(l) Imagem Corrigida.

Figura 5.5: Resultado de retificação para imagens com diferentes graus de distorção de perspectiva

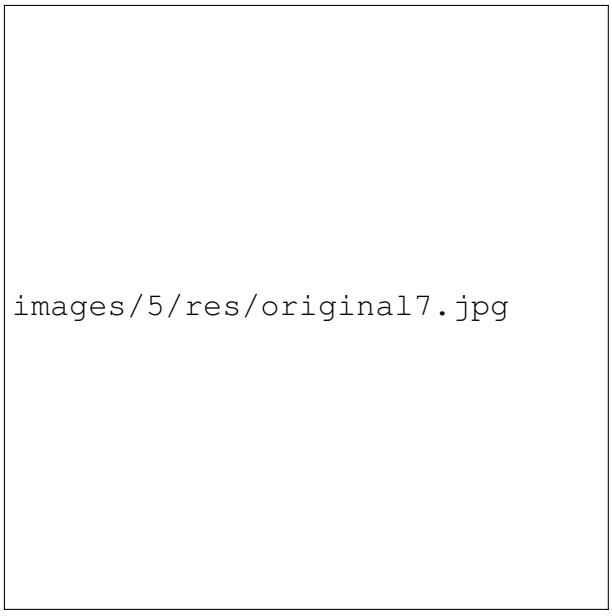
As imagens de documentos da Figura 5.6 apresentam embassamento. Este fenômeno acontece em decorrência da qualidade do dispositivo de captura ou mesmo pela perda de foco durante o processo de aquisição. Apesar disto, a retificação da imagem não é prejudicada (Figuras 5.6b, 5.6f e 5.6d). Outra característica importante é a robustez do método em relação à razão de aspecto dos documentos. O documento presente na Figura 5.6a detém orientação e um aspecto mais alongado quando comparado com o documento da Figura 5.6c.



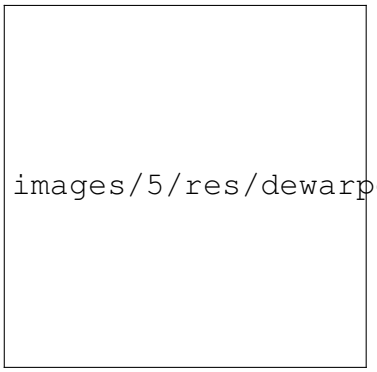
(a) Imagem Original.



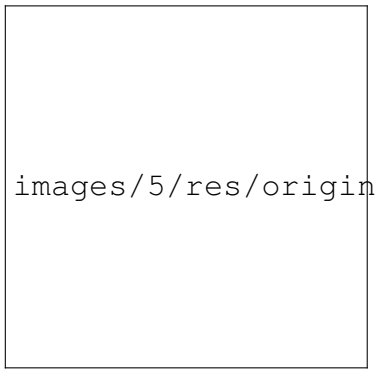
(b) Imagem Corrigida.



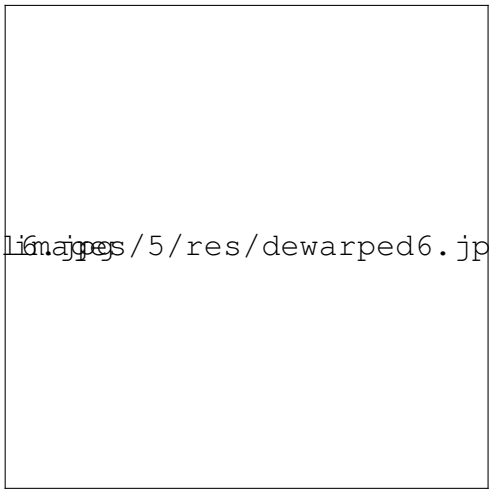
(c) Imagem Original.



(d) Imagem Corrigida.



(e) Imagem Original.



(f) Imagem Corrigida.

Figura 5.6: Resultado de retificação para imagens borradas.

A Figura 5.7 traz exemplos de documentos com cantos fora do domínio visível da imagem. Como explicado na Seção 4.3, os cantos do documento são determinados pelas intersecções das linhas que descrevem os limites do documento. Esta abordagem torna o método proposto independente da presença dos quatro cantos do documento na imagem.

O novo método também foi experimentado em imagens com *backgrounds* complexos, como mostra a Figura 5.8. Nota-se que, mesmo quando a retificação não é perfeita (Figura 5.8b), os componentes textuais se mantém na imagem resultante. O resultado ilustrado pela Figura 5.8d reforça a eficácia da técnica proposta mesmo quando o documento está sobre um *background* com uma textura complexa.

(a) Imagem Original.

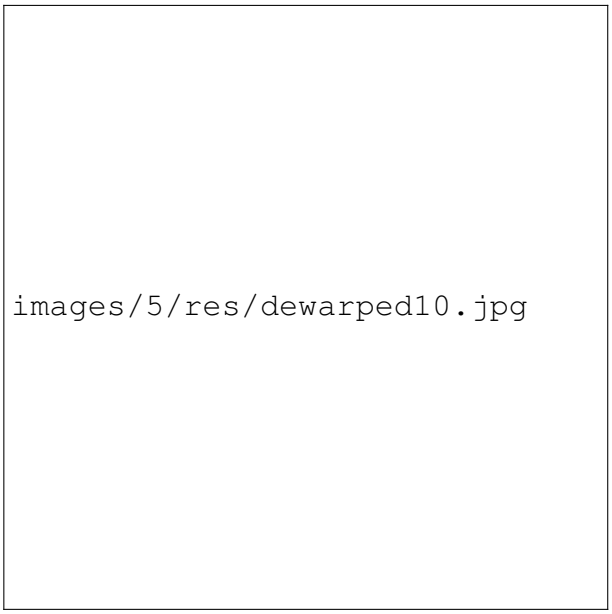
(b) Imagem Corrigida.

(c) Imagem Original.

(d) Imagem Corrigida.

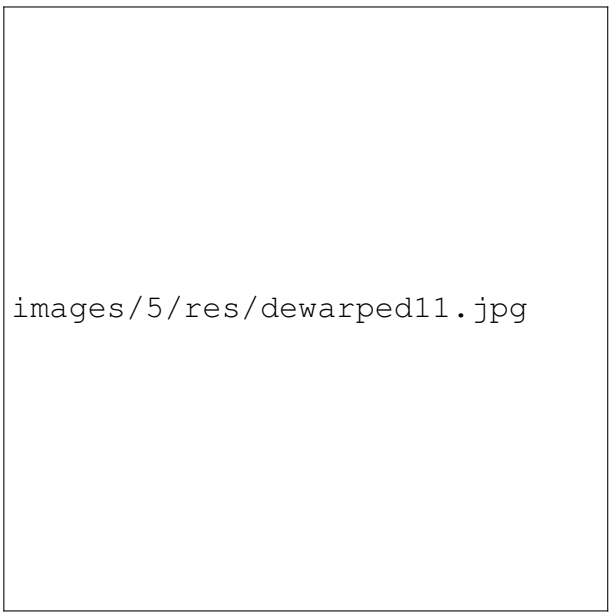
Figura 5.7: Resultado de retificação documentos com cantos fora do domínio visível.

(a) Imagem Original.



(b) Imagem Corrigida.

(c) Imagem Original.

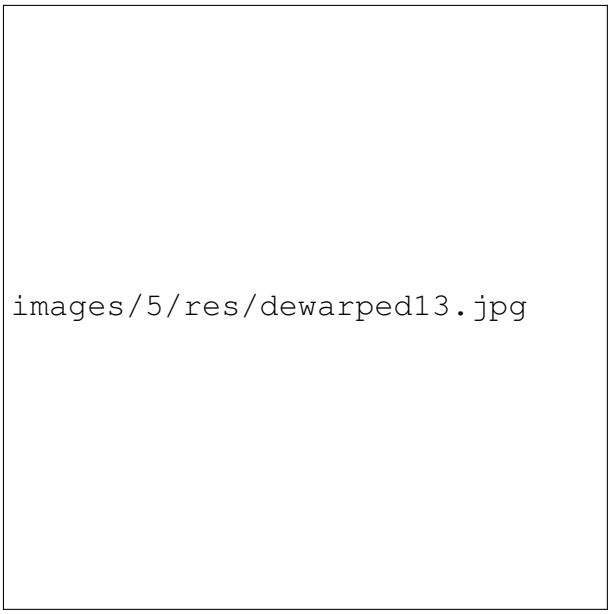


(d) Imagem Corrigida.

Figura 5.8: Resultado de retificação de imagens com *backgrounds* complexos.

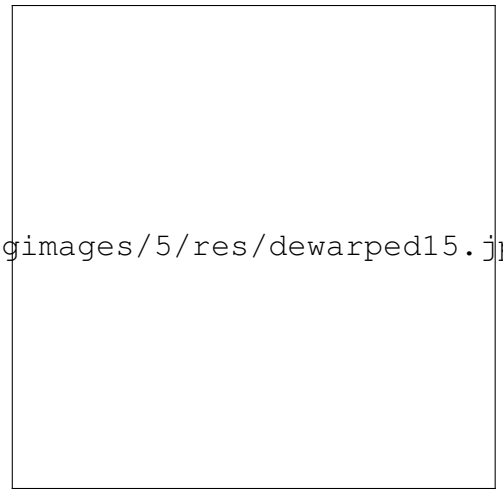
Até então, a maioria das imagens apresentadas nesta Seção obtiveram resultados bastante satisfatórios quanto a assertividade do *dewarping*. Porém, em alguns casos para a base utilizada, a retificação não é completamente alcançada. A Figura 5.9b ilustra o resultado em decorrência do mapeamento errado dos limites do documento original (Figura 5.9a). Neste caso, um elemento do *background* – bloco escuro no canto superior esquerdo – gerou linhas consideradas pertencentes à superfície distorcida. De maneira semelhante, na Figura 5.9f, a linha de base do documento original (Figura 5.9e) foi confundida por um componente do *background* não filtrado. Uma possível solução para estes casos é utilizar informações de cor para diferenciar os limites

(a) Imagem Original.



(b) Imagem Corrigida.

(c) Imagem Original.

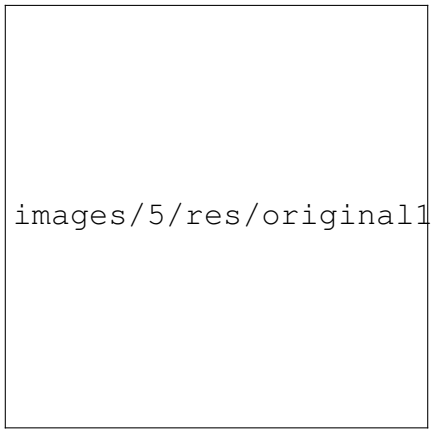


(d) Imagem Corrigida.

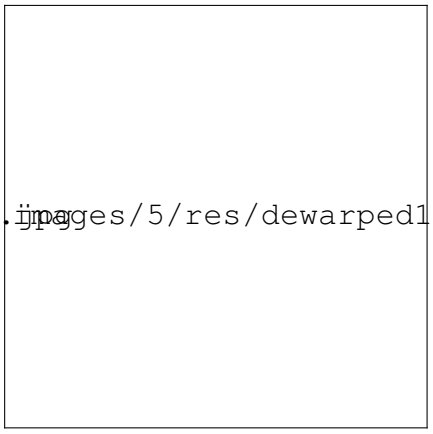
pertencentes ao documento.

O método proposto não conseguiu mapear a mudança de direção do contorno esquerdo do documento na Figura 5.9c, causando um deformidade no resultado final (Figura 5.9d). O erro neste caso é decorrente de uma dobradura presente no documento. Como o método proposto pressupõe que a superfície distorcida é descrita por um quadrilátero, este tipo de distorção não tem cobertura.

A base utilizada neste trabalho contém 30% de imagens que não apresentam nenhum tipo de deformação. Alguns resultados para estas imagens estão presentes na Figura 5.10. As imagens originais (Figuras 5.10a, 5.10c e 5.10e) são classificadas pelo seu grau de distorção, como explicado na Seção 4.3. Este procedimento tem um grau de tolerância a pequenas deformidades, e caso, a imagem esteja dentro da tolerância, a transformação é realizada apenas nas linhas sem inclinação.



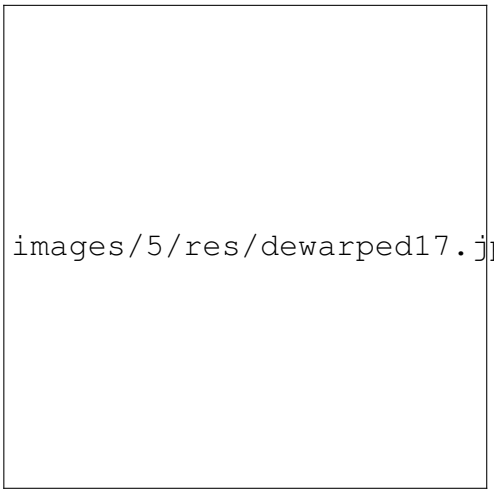
(e) Imagem Original.



(f) Imagem Corrigida.

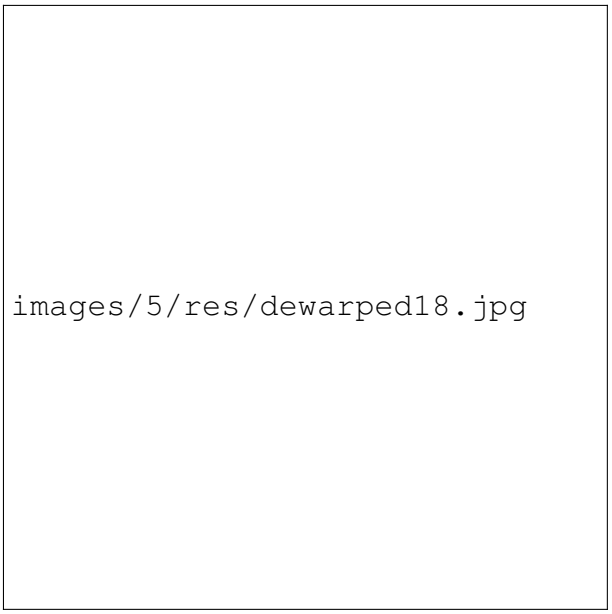
Figura 5.9: Imagens com retificação com erro evidente.

(a) Imagem Original.

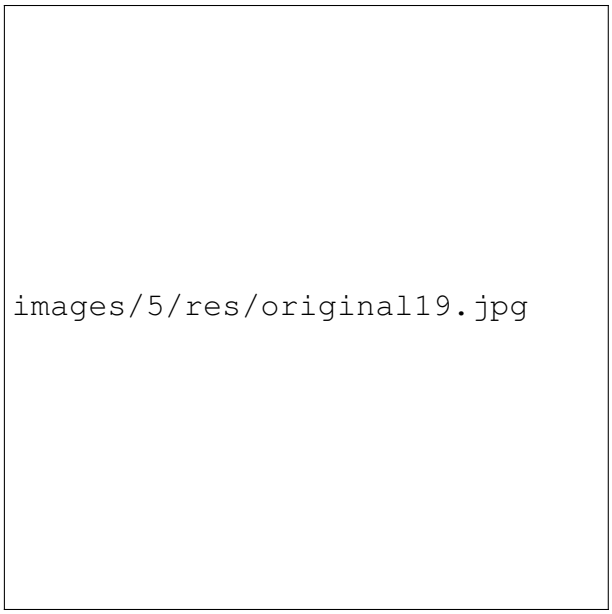


(b) Imagem Corrigida.

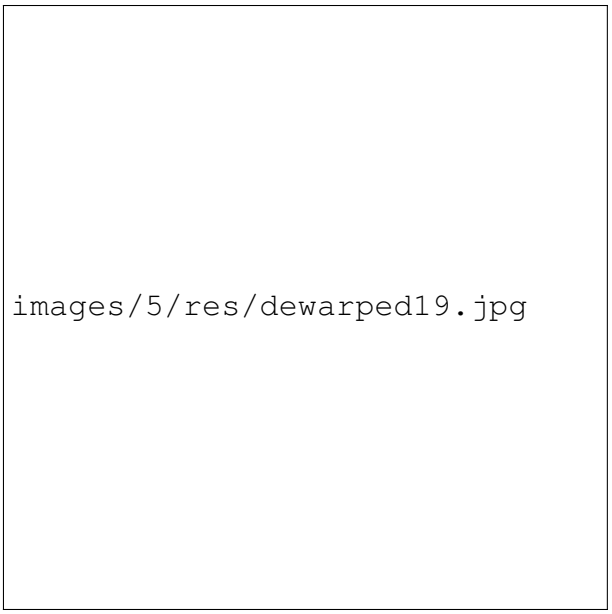
(c) Imagem Original.



(d) Imagem Corrigida.



(e) Imagem Original.



(f) Imagem Corrigida.

Figura 5.10: Resultado da aplicação da transformação em Imagens sem distorção evidente.

Um aspecto importante avaliado é o tempo de processamento do método proposto. Na Tabela 5.4 estão contidos os tempo médio e relativo discriminados por fase. O tempo de correção é muito maior que o tempo das demais fases, correspondendo a mais que 98% do tempo total. A aplicação do HT-HOG e a seleção de linhas duram por volta de 3 segundos, o que corresponde a menos de 1% do tempo total. O ajuste do contraste tem tempo desprezível se comparado com as demais fases.

Fase	Tempo Médio (s)	%
Ajuste Constraste	0,12	0,03
HT-HOG	3,01	0,78
Seleção de Linhas	3,44	0,89
Correção	377,44	98,28

Tabela 5.4: Tempo médio de cada fase do método proposto.

5.4.3 Reconhecimento

Nesta Seção, uma análise quantitativa das imagens corrigidas é realizada utilizando resultados de sistemas de OCRs como medida de desempenho. Os sistemas de OCR utilizados neste trabalho foram o *Transym Optical Character Recognition* (TOCR) (**?**) e o FineReader (**?**). Os resultados do reconhecimento para cada imagem original e corrigida são coletados e os acertos são contabilizados. A partir deste resultado, é possível testar a hipótese da melhora da legibilidade da imagem corrigida em relação a imagem original.

A medida utilizada para comparar o desempenho é o número de acertos perfeitos por palavra, também chamado de "*match* perfeito". Portanto, se algum caractere da palavra estiver suprimido ou trocado, esta palavra não é contabilizada como acerto. A Tabela 5.5 sintetiza as regras utilizadas para inferir se o resultado do OCR é contabilizado ou não.

Ground Truth	**Resultado OCR**	***Match* perfeito**	**Motivo**
República	República	Sim	Caracteres iguais
Brasil	brasil	Sim	Desconsidera Capitalização
Filiação	Filiacao	Sim	Desconsidera Caracteres especiais
Detran	Detram	Não	Caractere Incorreto
Número	Nmero	Não	Caractere Faltante

Tabela 5.5: Regras de contabilização de acertos do OCR.

Depois, define-se D_i pela Equação 5.5, onde HD_i e HO_i são os números de acertos perfeitos para a *i-ésima* imagem corrigida e original, respectivamente. Os valores dos $D_i s$ para os dois sistemas de OCR utilizados e para todas as imagens estão no Apêndice A. Portanto, D_i mede a diferença de acertos entre as versões corrigida e original, $D_i > 0$ reforça a hipótese de que o novo método é eficaz, enquanto que, $D_i < 0$ indica piora da legibilidade da imagem corrigida. Por sua vez, $D_i = 0$ indica que não houve diferença no reconhecimento pela aplicação do algoritmo proposto.

$$D_i = HD_i - HO_i \tag{5.5}$$

A Tabela 5.6 contém o resultado geral da análise da taxa de reconhecimento dos OCRs. Desta forma, pode-se afirmar que os resultados dos dois OCRs foram consistentes. A taxa de melhoria variou entre 40% (TOCR) e 42,5% (FineReader) das imagens, enquanto a taxa de piora

no desempenho ficou entre 15% (TOCR) e 17,5% (FineReader). O número de imagens que se mantiveram com mesmo número de acertos também foi próximo para ambos os OCRs.

	$D_i > 0$ (%)	$D_i < 0$ (%)	$D_i = 0$ (%)
FineReader	42,5	17,5	40
TOCR	40	15	45

Tabela 5.6: Taxas dos acertos perfeitos para os dois OCRs utilizados.

A partir destes números, dois testes de hipótese foram realizados para os resultados do FineReader e do TOCR. Como a aderência a normal foi rejeitada pelo teste *Kolmogorov-Smirnov* (KS) para as duas distribuições, foi utilizado o teste de *Wilcoxon*. Ambos os testes tinham como hipótese nula que o desempenho do OCR na imagem corrigida era inferior ou igual ($\overline{HD_i} \leq \overline{HO_i}$) ao da imagem original. As duas hipóteses foram rejeitadas com 95% de confiança para 39 graus de liberdade (Tabela 5.7).

		p-value	Resultado
FineReader	$H_0 : \overline{HD_i} \leq \overline{HO_i}$ $H_1 : \overline{HD_i} > \overline{HO_i}$	0,0069	Rejeitada
TOCR	$H_0 : \overline{HD_i} \leq \overline{HO_i}$ $H_1 : \overline{HD_i} > \overline{HO_i}$	0,0184	Rejeitada

Tabela 5.7: Testes de hipótese *t-student* realizados.

Os resultados anteriores são calculados para 100% da base. Porém, 30% das imagens não apresentam qualquer distorção (Tabela 5.1). Por isto, as taxas para os 70% de imagens com distorção foram calculadas (Tabela 5.8). Os resultados melhoraram para ambos OCRs: 46,43% (TOCR) e 50% (FineReader). Da mesma forma, o resultado do teste de hipótese foi ratificado para este este novo conjunto de dados, como mostra a Tabela 5.9.

	$D_i > 0$ (%)	$D_i < 0$ (%)	$D_i = 0$ (%)
FineReader	50	10,71	39,29
TOCR	46,43	7,14	46,43

Tabela 5.8: Taxas dos acertos perfeitos para os dois OCRs utilizados, considerando apenas imagens distorcidas.

		p-value	Resultado
FineReader	$H_0 : \overline{HD_i} \leq \overline{HO_i}$ $H_1 : \overline{HD_i} > \overline{HO_i}$	0,091	Rejeitada
TOCR	$H_0 : \overline{HD_i} \leq \overline{HO_i}$ $H_1 : \overline{HD_i} > \overline{HO_i}$	0,066	Rejeitada

Tabela 5.9: Testes de hipótese *t-student* realizados, considerando apenas imagens distorcidas.

5.4.4 Comparação com Stamatopoulos *et al.*

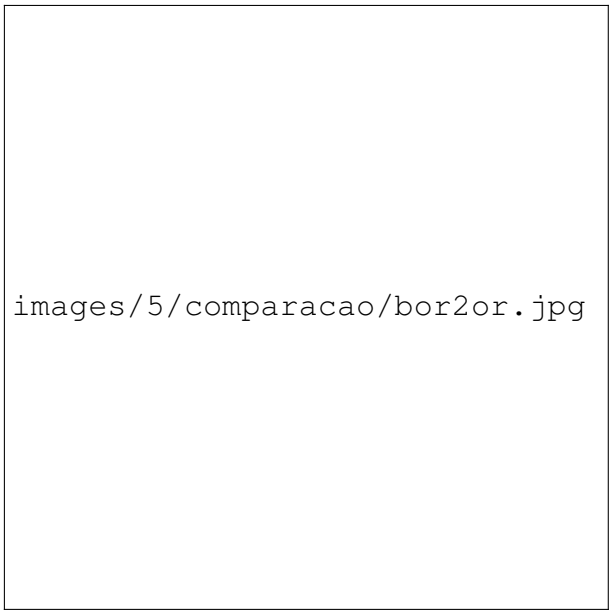
Para comparação de resultados, a técnica do estado da arte escolhida foi a de Stamatopoulos *et. al* que, em seus experimentos, alcançou 93,82% para taxa de reconhecimento de caracteres (**?**). Além disso, é a mais recente e, em comparação com as outras duas abordadas neste trabalho, as publicações serem mais ricas em detalhes, facilitando implementação. Stamatopoulos *et. al* aplicaram seu método a uma base própria com 100 imagens em 200 dpi composta basicamente por páginas de livros. Esta base não está disponível para *download*.

Como mostrado na Seção 3.2.3, este método utiliza uma limiarização adaptativa para preservação e melhoraria do texto (**?**). Uma das desvantagens deste método é a necessidade de fornecimento das dimensões médias dos caracteres ao algoritmo, para esta simulação foi utilizado 30 *pixels* para altura e largura. Alguns resultados da aplicação desta técnica na base de documentos utilizada neste trabalho estão na Figura 5.11.

O pré-processamento não obteve resultado satisfatório para imagens com borramento (Figuras 5.11a e 5.11c). Praticamente, toda informação contida nestes documentos foi perdida (Figuras 5.11b e 5.11d). Na Figura 5.11e, o pré-processamento não conseguiu separar o texto do background texturizado (Figura 5.11f), perdendo inclusive a maioria das informações textuais.

(a) Imagem Original.

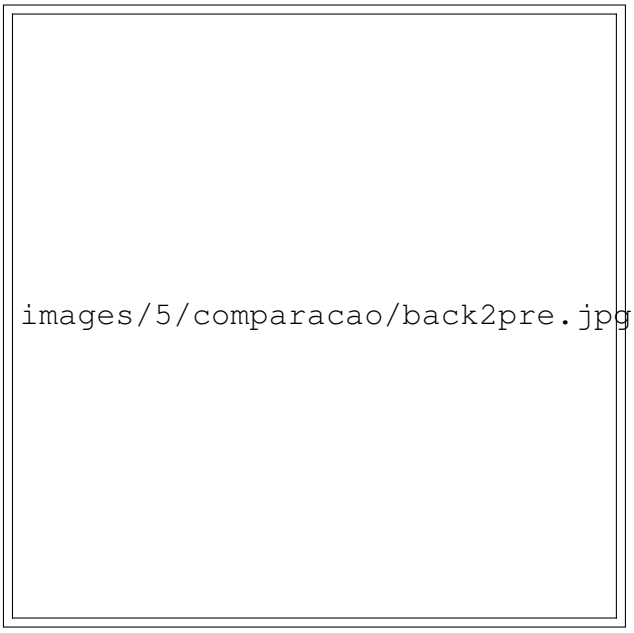
(b) Imagem Limiarizada.



(c) Imagem Original.

(d) Imagem Limiarizada.

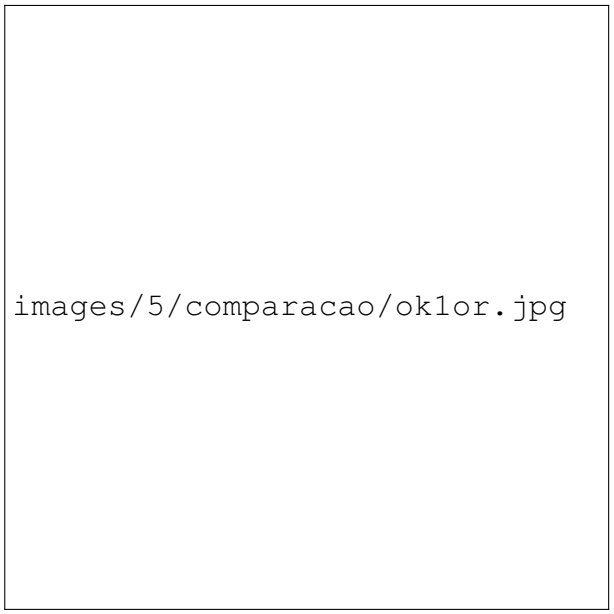
(e) Imagem Original.



(f) Imagem Limiarizada.

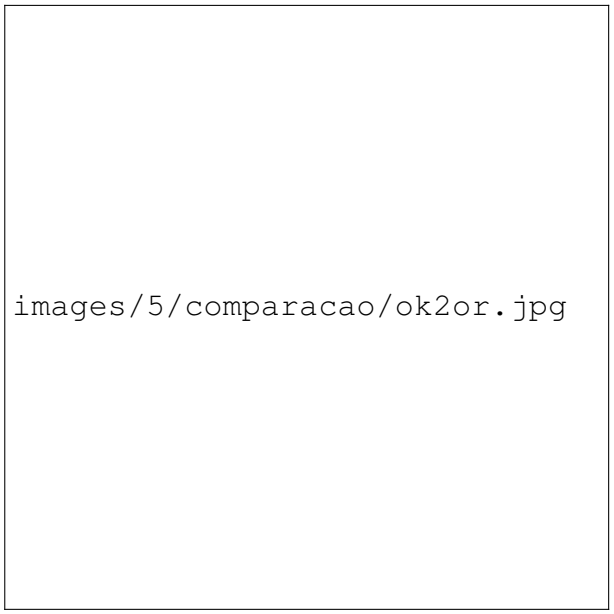
Figura 5.11: Resultado da aplicação do método de Limiarização.

Para imagens com contraste maior entre background e o texto, a limiarização apresentou resultados melhores, como mostra a Figura 5.12. Porém, as imagens resultantes apresentam ruídos causados, principalmente, pelas bordas remanescentes dos documentos. Um exemplo representativo deste comportamento é a imagem da Figura 5.12h, onde inclusive as bordas internas do documento permaneceram preservadas.



(a) Imagem Original.

(b) Imagem Limiarizada.



(c) Imagem Original.

(d) Imagem Limiarizada.

(e) Imagem Original.

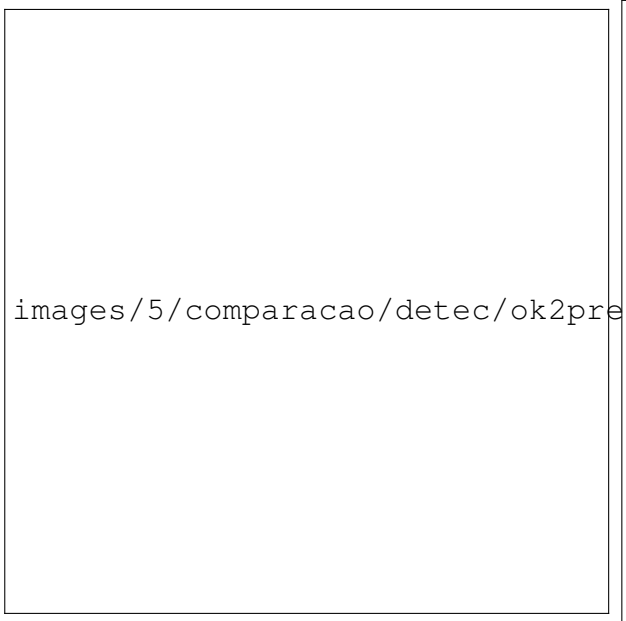
(f) Imagem Limiarizada.

(g) Imagem Original.

(h) Imagem Limiarizada.

Figura 5.12: Resultado da aplicação do método de Limiarização.

Após o pré-processamento da imagem, Stamatopulos *et. al* realizam a detecção de palavras, e posteriormente, o agrupamento em linhas. Esta abordagem pressupõe que o documento presente na imagem seja rico em informações textuais, as letras tenham um tamanho uniforme e o texto em formato justificado. Um exemplo de resultado para esta abordagem está ilustrado na Figura 5.13, onde o resultado da segmentação das linhas está colorizado para melhor visualização.



(a) Imagem Limiarizada.

(b) Linhas Segmentadas.

(c) Determinação da Superfície distorcida.

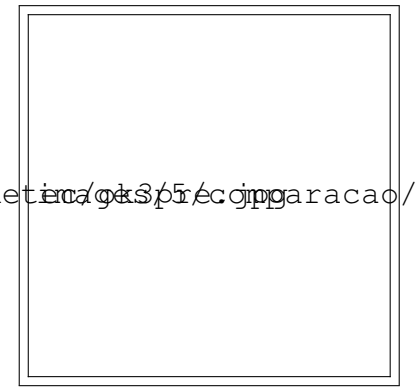
Figura 5.13: Resultado do agrupamento de linhas.

A princípio, ao analisar a Figura 5.13b, a segmentação está correta, pois existe uma separação lógica entre as linhas. Porém, o algoritmo de Stamatopoulos *et. al* se baseia nas linhas imaginárias estimadas a partir das linhas segmentadas. Isto funciona quando o texto é justificado, como mostra a Figura 5.13c. Mas como há um alto nível de dispersão na disposição do texto nas imagens utilizadas, a proposta de Stamatopoulos *et. al*, não consegue aferir adequadamente a superfície distorcida.

A Figura 5.14 ilustra o resultado da segmentação para uma imagem com orientação invertida. Além de apresentar o mesmo problema da figura anterior, neste exemplo, percebe-se

que o agrupamento das linhas é irregular devido ao espaçamento heterogêneo entre as letras. Em detalhe (Figura 5.14c), é possível perceber que há caracteres que não foram agrupados em nenhuma linha.

(a) Imagem Limiarizada.



(b) Linhas Segmentadas.

(c) Detalhe da Segmentação.

Figura 5.14: Resultado do agrupamento de linhas.

O fato de não ser possível calcular a superfície distorcida a partir das linhas encontradas, impede a correção da imagem. Portanto, a abordagem de Stamatopulos *et. al* não foi capaz de convergir para nenhuma das imagens da base utilizada nesta pesquisa.

5.4.5 Teste em Imagens Ricas em Texto

A base de imagens utilizada por Stamatopoulos *et al.* não está disponível para *download*. Por este motivo, algumas imagens apresentando distorção de perspectiva, inclinação e forte presença de texto foram capturadas e experimentadas no método proposto. O resultado deste teste está na Figura 5.15.

O método proposto conseguiu,, de maneira satisfatória, retificar as distorções presentes nestas imagens (Figuras 5.15a e 5.15c). A primeira imagem apresenta distorção de perspectiva e inclinação, e sua correção (Figura 5.15c) mantém o conteúdo textual do documento. Por sua vez, a segunda imagem sofre apenas com distorção de profundidade que também apresentou sucesso na sua correção (Figura 5.15d).

(a)

(b)

(c)

(d)

Figura 5.15: Imagens Ricas em Texto.

6

Conclusão e Trabalhos Futuros

A pesquisa realizada e apresentada neste trabalho permitiu a concepção de uma nova abordagem para retificação de imagens digitalizadas por câmeras. Esta nova abordagem é uma composição entre a Transformada de Hough e o Descritor de Histogramas de Gradientes Orientados.

A principal motivação deste trabalho partiu, em primeiro lugar, da necessidade de captura de documentos em ambientes restritos onde o uso de *scanners* de mesa são proibitivos. Neste cenário, dispositivos portáteis – como celulares ou câmeras – tem sido uma alternativa viável. Porém, o uso destes dispositivos causam uma série de distorções indesejadas que podem impedir o uso destas imagens para diversos fins, como reconhecimento automático de texto. Além do mais, as técnicas previamente divulgadas para retificação de imagens com estas distorções são estreitamente dependentes de uma alta densidade de texto na imagem do documento.

Portanto, a proposta deste trabalho foi desenvolver uma técnica capaz de corrigir as distorções causadas por capturas desta natureza sem dependência de conteúdo textual. Desta forma, as principais contribuições deste trabalho são abordadas na seção a seguir.

6.1 Contribuições

A principal contribuição deste trabalho foi desenvolver um método eficaz para retificação de imagens de documentos capturadas por dispositivos móveis utilizando a Transformada de Hough diretamente aos Histogramas de Gradientes Orientados. Esta composição forneceu um novo viés para interpretação das componentes da imagem, permitindo uma melhor análise e identificação das bordas dos documentos. O êxito da utilização desta composição neste trabalho sugere sua experimentação em outros problemas de segmentação.

Além disto, foi realizado um estudo sobre o impacto da variação dos parâmetros livres *cellSize* e *numBins* na geração dos HOGs. A pressuposição de que valores mais altos de *cellSize* e *numBins* iriam melhorar a assertividade da extração das bordas do documento não foi ratificada. Apesar de gerar mais informação, valores altos destes parâmetros aumentam a quantidade de ruído e deteriora a capacidade de extrair informação do espaço de Hough.

Ainda mais, o método inclui uma maneira eficaz de avaliar o grau de distorção da imagem do documento. Para isto, uma API para manipulação de HOGs foi desenvolvida, permitindo realizar, também, seleção de histogramas com uma ângulo de inclinação específico, remoção de componentes anômalos e uma técnica para normalização dos histogramas.

O método proposto corrigiu de maneira satisfatória as imagens com distorção de perspectiva e inclinação. Além disto, embora não fosse um objetivo deste trabalho, funcionou adequadamente para imagens desfocadas e com orientação de texto invertidas. Para base de documentos pessoais coletadas para avaliar a eficácia da técnica proposta, o menor erro obtido foi de 4,08%.

Esta base foi construída especialmente para este trabalho e é composta por 40 exemplos de imagens capturadas por câmeras de celulares e tablets. As imagens da base apresentam distorções de inclinação e perspectiva ou mesmo distorção alguma. Além disto, o *ground truth* foi adquirido utilizando a percepção visual de quatro voluntários para cada uma das 40 imagens.

A técnica proposta em **?**) foi implementada em MATLAB e experimentada na base de documentos pessoais. Por ser fortemente dependente do formato e conteúdo do texto, não obteve resultados satisfatórios.

6.2 Trabalhos Futuros

Trabalhos futuros decorrentes desta pesquisa incluem:

- Adicionar suporte a documentos dobrados. O método foi concebido para buscar quadriláteros, enquanto, um documento com dobradura pode conter mais de quatro cantos, inclusive cantos convexos;
- Tratar documentos com distorção de encurvamento, como por exemplo em fotos de livros espessos;
- Utilizar a informação de tamanho da linha fornecida pela Transformada de Hough para refinar o método;
- Utilizar informações de cor, quando houver, para remoção de *outliers*; e
- Disponibilizar uma base de documentos pessoais para fomentar *benchmarking* para estes tipos de documentos;
- Investigar impacto da mudança de resolução das imagens na assertividade do método para possibilitar calibragem automática, tornando o método adaptativo.

,

Apêndice

A

Tabela da Diferença de Acertos de Classificação Entre as Imagens Originais e Corrigidas

Na Tabela A.1 encontram-se os valores das diferenças entre os acertos perfeitos das imagens corrigida e original para dois OCRs: TOCR e FineReader. Células com valores iguais a zero sugerem que a aplicação do algoritmo de correção não surtiu efeito na legibilidade, valores positivos sugerem que o novo algoritmo melhorou a legibilidade, enquanto valores negativos indicam a degradação da legibilidade pelos OCRs. Estes valores foram utilizados no teste hipótese, como detalhado na seção 5.4.3.

Imagem	D_i FineReader	D_i TOCR
i=1	0	0
i=2	5	-5
i=3	28	6
i=4	-1	1
i=5	0	0
i=6	0	0
i=7	36	0
i=8	0	0
i=9	13	-6
i=10	6	8
i=11	0	2
i=12	0	0
i=13	-3	1
i=14	-4	1
i=15	3	7
i=16	0	-2
i=17	1	0

i=18	12	-2
i=19	0	-3
i=20	2	6
i=21	2	0
i=22	6	0
i=23	9	6
i=24	13	9
i=25	5	1
i=26	0	0
i=27	8	0
i=28	3	0
i=29	0	7
i=30	21	5
i=31	-5	0
i=32	0	0
i=33	0	0
i=34	0	3
i=35	0	0
i=36	-5	0
i=37	0	1
i=38	-3	-3
i=39	-2	0
i=40	0	5

Tabela A.1: Diferenças (D_i) entre os acertos perfeitos das imagens originais e corrigidas.

Referências