

Detecção de Inclinação em Imagens de Documentos

Angélica A. Mascaro ¹
George D. C. Cavalcanti ¹
Carlos A. B. Mello ¹

Resumo:

A digitalização de documentos contribui para a preservação da informação evitando sua perda devido à degradação física do papel. Atualmente, Sistemas de Reconhecimento Automático de Imagens de Documentos são empregados para converter, automaticamente, a informação contida nas imagens em texto editável, de forma rápida e sem a necessidade da presença de um indivíduo. Assim, tornando essa informação pesquisável através, por exemplo, de palavras-chave. A inclinação em documentos é um problema freqüente nesses sistemas e, em geral, é imposta durante a digitalização, quando o papel é posicionado com um ângulo diferente de zero grau sobre o eixo do *scanner*. No caso de documentos manuscritos, a inclinação pode surgir durante a escrita do próprio documento, principalmente quando o escritor não tem uma linha de pauta como guia. A correção da inclinação é essencial para o bom desempenho de sistemas de reconhecimento automático. Este trabalho aborda o problema da detecção de inclinação em documentos impressos e manuscritos, trazendo uma revisão dos principais métodos para detecção de inclinação divulgados na literatura até os dias atuais. As principais técnicas são expostas de forma categorizada e vantagens e limitações de cada método são discutidas.

Abstract:

Scanning documents contributes to preserving information, avoiding loss due to paper physical degradation. Nowadays, Automatic Image Document Recognition Systems are used to convert the information contained in images to editable text, in a fast way and without the presence of an individual. Thus, this information can, for example, be searchable through keywords. Skew is a frequent problem in these systems and is commonly introduced during scanning, when the paper is aligned in an angle different of zero degree upon the scanner axis. In the case of handwritten documents, the skew can appear during writing, especially when the person does not have a baseline to be guided by. The skew correction is essential for the good performance of such automatic recognition systems. This work deals with the problem of skew estimation of printed and handwritten documents and brings a revision of the main methods of skew estimation reported in the literature until nowadays. The main techniques are explained in categories and advantages and limitations of each method are discussed.

¹Centro de Informática, UFPE. {<http://www.cin.ufpe.br/~viisar>}
{aam3,gdcc,cabm}@cin.ufpe.br

1 Introdução

O papel ainda é um dos meios mais comuns para armazenamento e distribuição de informações. No entanto, o papel é frágil e suscetível ao envelhecimento. A digitalização de documentos tem sido amplamente utilizada como forma de preservação da informação em formato eletrônico [1], evitando a perda das informações contidas nesses acervos devido à degradação física do papel. A digitalização também torna possível o acesso a bibliotecas e acervos em meio digital — como a Internet — facilitando a distribuição da informação.

Além da vantagem de não sofrer com degradação causada pelo tempo, as informações passadas para um meio digital também ocupam menos espaço físico. Informações em meio digital podem ser armazenadas em vários tipos de mídias, tais como CDs, DVDs e, também, vários tipos de armazenamento magnéticos, como HDs, evitando a necessidade do grande espaço físico ocupado por livros, documentos, relatórios, etc. Além disso, arquivos em um computador são localizados mais facilmente do que um pequeno documento em uma grande biblioteca ou acervo.

No entanto, na maioria das vezes deseja-se que a informação contida em tais documentos seja pesquisável através, por exemplo, de palavras-chave. Ou seja, deseja-se que seja feita a extração do texto da imagem a qual pode ser feita manualmente através de um indivíduo que transcreve a informação contida nos documentos. No entanto, esse é um processo é lento.

Sistemas de Reconhecimento Automático de Documentos (SRAD) são empregados para converter automaticamente a informação contida nas imagens em texto editável de forma rápida e sem a necessidade de um indivíduo. A digitalização e posterior reconhecimento de documentos também são úteis para acelerar atividades que seriam custosas para serem realizadas manualmente e auxiliam organizações que desejam automatizar algum processo. Sem um SRAD é necessário que um indivíduo observe o texto de um documento e o digite no computador. Por exemplo, o uso de um SRAD acelera o processamento de cheques no banco, sem que seja necessário uma pessoa ler e digitar cada campo do cheque.

A Figura 1 ilustra um esquema com as etapas de um SRAD. São elas:

1. **Captura:** o documento é digitalizado através de um *scanner* ou máquina fotográfica;
2. **Processamento da Imagem:** a imagem capturada é preparada para a fase de classificação. Isto inclui, por exemplo, a conversão da imagem para dois tons — binarização — que converte para preto os pixels correspondentes à tinta do papel e para branco os pixels do plano de fundo. Nesta fase também faz-se a análise de diagramação (*layout*) da imagem, em que se realiza a identificação e a segmentação das regiões de interesse no documento. Nesta etapa ainda estão inclusas a remoção de ruídos através de filtros, a correção da inclinação da imagem, etc.;



Figura 1. As etapas de um sistema de reconhecimento de documentos: captura, processamento da imagem, reconhecimento dos caracteres e geração da saída.

3. **Reconhecimento de Caracteres:** também conhecida como fase de classificação. Nesta fase ocorre a identificação de cada caractere através de um classificador. Esses classificadores são popularmente conhecidos como ferramentas OCR (*Optical Character Recognition*), para o caso de documentos gerados mecanicamente [2];
4. **Geração do Arquivo de Saída:** os caracteres reconhecidos na etapa anterior são agrupados na ordem correta e enviados a um arquivo de saída.

Nesses sistemas, a qualidade da imagem de entrada é crucial para o desempenho final. Há uma variedade de aspectos que interferem no processo, como ruído e inclinação, que danificam a imagem e reduzem o desempenho do sistema.

O ruído pode ocorrer devido à degradação do documento ou devido ao processo de binarização, que pode, por exemplo, converter para preto os pixels do plano de fundo caso o papel esteja sujo ou degradado. Já a inclinação é um problema freqüente e, em geral, é adquirida durante a digitalização, quando o documento é posicionado com um ângulo diferente de zero grau sobre o eixo do *scanner*. A Figura 2(a) mostra um exemplo de documento digitalizado com inclinação.

Para documentos manuscritos, a inclinação aparece não apenas devido à fase de digitalização, mas também pode ser adquirida durante a escrita do documento, especialmente quando o escritor não tem uma linha base como guia. É o caso em que o escritor geralmente não percebe que está escrevendo as linhas do texto com inclinação.

Em geral, podemos observar três tipos de inclinação em um documento:

1. Inclinação global: quando toda a área de texto possui a mesma inclinação (Figura 2(a));
2. Inclinação múltipla: quando certas áreas de texto possuem inclinações distintas de outras áreas de texto na imagem (Figura 2(b));

3. Inclinação não-uniforme dentro de uma linha de texto: quando a orientação “flutua” dentro de uma linha, i.e., uma linha se “entorta” em uma ou em ambas as extremidades, ou a linha contém um formato de “onda” (Figura 2(c)).

O primeiro caso citado ocorre em documentos impressos e manuscritos, enquanto que os dois últimos são mais comuns em documentos manuscritos. O foco desse trabalho está na identificação dos dois primeiros casos.

A correção de inclinação tem um importante papel no pré-processamento da imagem, pois interfere nas etapas subsequentes. Uma pequena inclinação no documento pode interferir na análise da diagramação e conseqüentemente no restante do processo. Ou seja, fases importantes do processo, tais como segmentação e classificação, ficam prejudicadas.

Para correção da inclinação é necessário o uso de um algoritmo para estimar o ângulo da inclinação dos documentos. O desenvolvimento desse tipo de algoritmo ainda é uma área ativa no processamento de imagens e enfrenta desafios, tais como redução do custo computacional e capacidade de lidar com diversos tipos de diagramações e componentes não-textuais na imagem, tais como presença de ruído.

A correção da inclinação para o caso de documentos manuscritos é ainda mais problemática que o caso dos documentos impressos. O texto manuscrito não tem uma aparência uniforme, mudando de pessoa para pessoa. Além disso, como ilustrado na Figura 2, é comum encontrar documentos contendo múltipla inclinação ao longo das linhas de texto, isto é, um ângulo diferente para cada linha no documento ou mais de uma inclinação dentro de uma única linha. Esse problema torna a estimativa da inclinação mais difícil.

Na Seção 2, são apresentados os principais métodos para detecção de inclinação em imagens de documentos. As diversas técnicas são categorizadas e apresenta-se uma discussão sobre vantagens e limitações de cada categoria. São expostos ainda exemplos de trabalhos já divulgados na literatura até os dias atuais.

A Seção 3 apresenta comentários sobre os algoritmos apresentados e considerações finais.

2 Técnicas de Detecção de Inclinação

Para resolver o problema da inclinação em imagens de documentos muito esforço foi empregado para desenvolver técnicas capazes de eliminar a inclinação presente na imagem. Sistemas de reconhecimento automático de texto precisam de algoritmos de pré-processamento para estimar e corrigir a inclinação das imagens a fim de alcançar uma alta taxa de acerto.

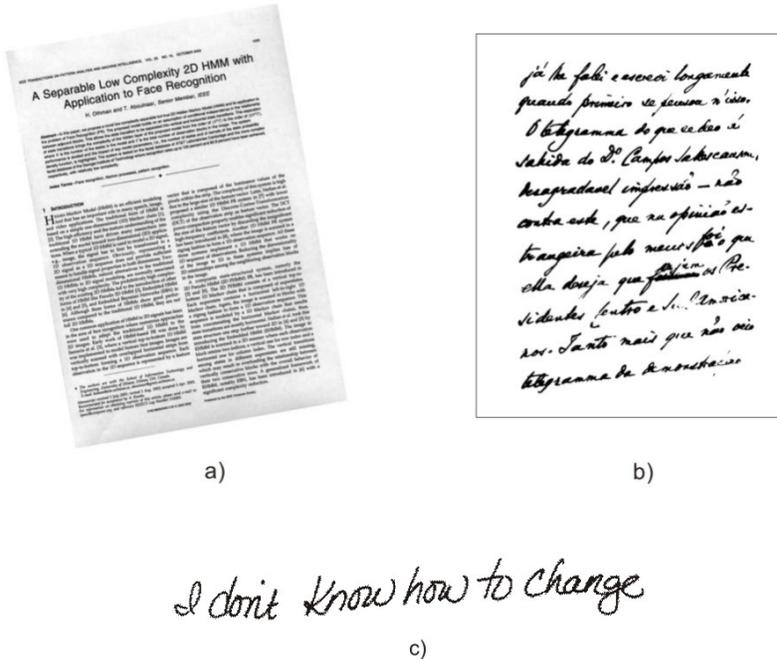


Figura 2. (a) Um documento com inclinação global de 9 graus no sentido anti-horário. (b) Um documento contendo linhas com inclinação diferentes. (c) Porção de texto com inclinação variável dentro da linha.

Grande parte dos algoritmos que trabalham com estimativa de inclinação lidam com documentos contendo uma única inclinação na imagem. Apenas poucos trabalhos lidam com documentos com múltiplas inclinações. Além disso, a detecção e correção de inclinação de documentos manuscritos permanece pouco explorada.

Vários algoritmos foram desenvolvidos para estimar o ângulo de inclinação de documentos impressos digitalizados [3, 4]. Depois de estimar o ângulo de inclinação, um algoritmo de correção deve ser utilizado para ajustar o documento, rotacionando-o [5]. A Figura 2 apresenta um exemplo da correção da inclinação de um documento.

Na Tabela 1, apresenta-se um resumo do estado-da-arte inicialmente construída por Cattoni *et al.* [3], posteriormente complementada por Ávila e Lins [6] e atualizada neste trabalho. Nela, é apresentada uma categorização dos principais algoritmos de detecção de inclinação, além das características de cada técnica.

Os métodos mais comuns são baseados em análise de projeção, transformada de

Hough, agrupamento por vizinhos-mais-próximos (*nearest neighbor clustering*) ou correlação-cruzada (*cross-correlation*). No entanto, técnicas baseadas em outras estratégias também foram desenvolvidas. As seções a seguir expõem as principais técnicas desenvolvidas até o presente momento, comentando suas vantagens e limitações.

2.1 Técnicas Baseadas em Análise de Projeção

Métodos baseados em análise de projeção assumem que uma imagem é formada por regiões de texto organizadas em linhas paralelas. A detecção do ângulo é feita projetando essa região de texto em vários ângulos e decidindo dentre esses qual a real inclinação do documento. Para cada ângulo testado, a quantidade de pixels pretos em cada linha é projetada e uma função objetivo é calculada para esse perfil de projeção. O ângulo escolhido é aquele que otimiza essa função objetivo, comumente chamada de função *premium*.

A Figura 3 ilustra a diferença no perfil de projeção de uma imagem com texto a 0° e com texto a 8° de inclinação. O perfil de projeção possui máximas amplitude e frequência quando o texto está a 0° .

Tabela 1. Classificação dos principais algoritmos para detecção de inclinação.

Método	Autores	Características
Projeção	Postl [7]	Pontos a serem projetados são filtrados por uma "grade".
	Baird [8]	Os pontos médios dos <i>bounding-boxes</i> são projetados.
	Ciardiello <i>et al.</i> [9]	Sub-região contendo alta densidade de pixels é usada na projeção.
	Bagdanov e Kanai [10]	Seleção de pontos para projeção através da análise de seqüências de pixels pretos
	Ishitani [11]	Uso de mapas de transições.
	Li <i>et al.</i> [12]	Uso da transformada Wavelet.
Transformada de Hough	Hinds <i>et al.</i> [13]	Identificação de seqüências de pixels pretos horizontais e verticais.
	Le <i>et al.</i> [14]	Orientação obtida a partir de sub-regiões classificadas como "retrato" ou "paisagem" através de projeção. Inclinação obtida através de transformada de Hough aplicada à última linha de cada componente conectado.
	Min <i>et al.</i> [15]	Redução dos pontos para aplicação da transformada de Hough utilizando <i>divided horizontal histograms</i> .
	Pal e Chaudhuri [16]	Seleção de alguns pontos na parte superior e inferior dos caracteres para aplicação da transformada de Hough para texto em fonte <i>Roman</i> .
	Amin e Fischer [17]	Transformada de Hough é aplicada à última linha pertencente a blocos de texto.
	Singh <i>et al.</i> [5]	Seleção de alguns pontos de contornos dos componentes conectados para aplicação da transformada de Hough.
Agrupamento de vizinhos mais próximos	Pal e Chaudhuri [16]	Avaliação das linhas superior e inferior das linhas de texto.
	Ávila e Lins [6]	Busca dos <i>bounding-boxes</i> dos caracteres e desenha linhas imaginárias que acompanham as linhas de texto para detecção de orientação e inclinação
	Lu <i>et al.</i> [18]	Detecção dos seqüências de pixels brancos entre as linhas do texto. Orientação obtida através de ascendentes e descendentes.
	Smith [19]	Filtragem de componentes de tamanho grande ou pequeno de acordo com uma distribuição das alturas e agrupamento dos componentes restantes em linhas horizontais.
	Shivakumara e Kumar [2]	"Crescimento de fronteiras" detecta palavras. Uso de regressão linear para traçar linhas unindo palavras.
	Saragiotis e Papamarkos [20]	Detecção de mais de um ângulo na imagem.
Correlação-Cruzada	Akiyama e Hagita [21]	Faixas verticais de mesma largura são deslocadas buscando melhor correlação entre suas projeção.
	Yan [22]	Aplicável a imagens em RGB, tons de cinza ou binárias. Uso de correlação acumulada entre faixas verticais selecionadas a uma distância fixa.
	Gatos <i>et al.</i> [23]	Preprocessamento através de suavização horizontal. Estudo da largura ideal das faixas.
	Avanindra [24]	Amostragem de Monte Carlo para selecionar regiões.
Paralelogramos	Chou <i>et al.</i> [25]	Linhas traçadas para construir paralelogramos que se ajustem aos objetos da imagem.
	Mascaro e Cavalcanti [26]	Busca otimizada pelo ângulo. Lida com imagens ruidosas e muitos separadores verticais.
Gradiente	Sauvola e Pietikainen [27]	Análise da direção do gradiente da textura de sub-regiões da imagem.
	Sun e Si [28]	documentos complexos, contendo texto e com intervalo entre-linhas maior que intervalo entre-caracteres
Morfologia	Chen e Haralick [29]	Operações morfológicas de abertura e fechamento para conectar caracteres e palavras.
Slide	Aghajan <i>et al.</i> [30]	Adaptação ao problema da detecção da direção de ondas eletromagnéticas. Detecta múltiplos ângulos.

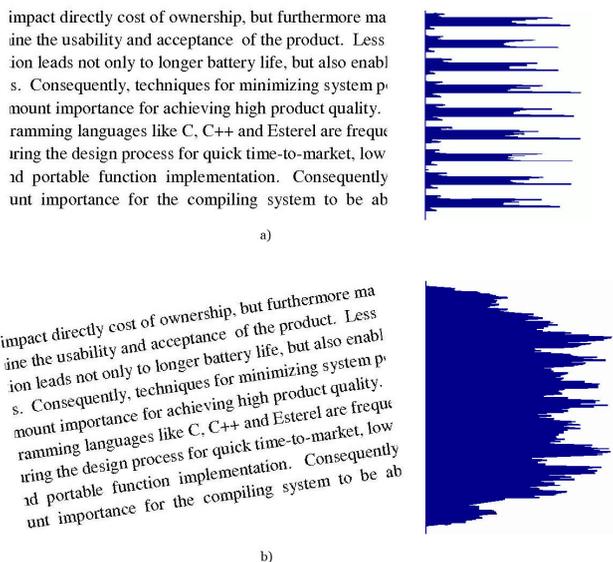


Figura 3. a) Imagem com 0° e b) Imagem com 8° de inclinação no sentido anti-horário e seus respectivos perfis de projeção.

Métodos baseados em projeção assumem que o documento é predominantemente constituído de linhas de texto e geralmente são vulneráveis à presença de outros elementos, como figuras, tabelas ou ruído. Evoluções na técnica tentam eliminar componentes gráficos, mas assumem que o documento possui uma região mínima de linhas paralelas de texto. Originalmente, a análise de projeção possui um grande custo computacional e os trabalhos publicados nessa linha tentam minimizar a quantidade de dados envolvidos na computação do ângulo, ou tentam melhorar a estratégia de busca. O grau de precisão do ângulo pode ser controlado com a mudança do tamanho do intervalo entre os ângulos testados. Para avaliações mais precisas, maior tempo computacional é demandado para que mais ângulos possam ser testados.

O Algoritmo 1 apresenta os principais passos de um algoritmo genérico baseado no método de projeção. O algoritmo recebe como entrada a imagem, soma a quantidade de pixels pretos para cada linha da imagem e retorna o ângulo de inclinação da imagem.

No trabalho de Postl [7], é utilizada outra abordagem para reduzir o volume de dados processados: os pontos a serem projetados são previamente filtrados por uma "grade", e apenas alguns pontos nessa grade são usados para computar o perfil de projeção.

Baird [8] propôs uma técnica para selecionar os pontos a serem projetados: inicial-

Algoritmo 1 Algoritmo de Detecção de Inclinação baseado em Projeção

```

1: PROJ(Imagem)
2: numeroDeLinhasDaImagem ← Size(Imagem.linhas)
3: Imagem ← Not(Imagem)
4: for i = 1 to numeroDeLinhasDaImagem do
5:   Profile(i) ← sum(Imagem.linhas(i))
6: end for
7: for  $\theta = \theta_1$  to  $\theta_2$  do
8:   valor( $\theta$ ) ← FuncaoObjetivo(Profile)
9: end for
10: [maxarg,  $\theta_o$ ] ← max(valor)
11: return  $\theta_o$ 

```

mente a imagem é rotulada em componentes conectados, em que os pixels com conectividade de 8 são agrupados em grupos de componentes. Para cada componente conectado, apenas o ponto médio da parte superior do menor retângulo contendo o componente conectado (*bounding box*) é projetado, como exemplificado na Figura 4. A análise da inclinação continua a partir da projeção desses pontos. A função objetivo a ser otimizada é calculada como a soma dos quadrados dos valores do perfil de projeção. Para acelerar a busca pelo ângulo, Baird propôs um método iterativo: na primeira iteração todo o intervalo é percorrido com uma precisão mais grosseira; a cada iteração posterior a busca é restrita a uma vizinhança do melhor ângulo até então encontrado, e a granularidade da busca é refinada, mas restringe-se o intervalo de busca a no máximo $\pm 15^\circ$. Baird afirma que seu método funciona para uma variedade de *layouts*, incluindo tabelas, múltiplos estilos e tamanhos de fontes.



Figura 4. Algoritmo de Baird: apenas o ponto médio superior do *bounding box* é projetado.

Ciardiello et al. [9] propuseram filtrar a quantidade de dados a serem processados através da seleção de uma sub-região no documento que contenha alta densidade de pixels pretos por linha e essa região é utilizada na projeção. A função a ser maximizada para descobrir o ângulo real é o desvio médio quadrático do perfil projetado.

A técnica proposta por Li *et al.* [12], usa a transformada Wavelet [31] para decompor

a imagem e encontrar o ângulo de inclinação através de análise de projeção. A influência do filtro base Wavelet no resultado é avaliada variando entre as bases "Daubechies", "Symlets" e "biortogonal". A matriz com os coeficientes das sub-bandas horizontais é rotacionada em vários ângulos e, aquele ângulo que maximiza uma dada função objetivo, é admitido como o ângulo de inclinação do documento.

Ishitani [11] propôs uma técnica baseada em mapas de transições que são projetados horizontalmente. Essa pode ser considerada uma variação da análise de projeção em que, ao invés de projetar a quantidade de pixels pretos em cada linha, a quantidade de transições de preto para branco são projetadas. O trabalho de Ishitani lida com documentos contendo mistura de áreas de texto, fotografias, figuras ou tabelas.

2.2 Técnicas Baseadas na Transformada de Hough

A transformada de Hough [31] é uma técnica bastante utilizada para detecção de linhas em imagens e essa funcionalidade têm sido muito explorada por autores para detecção de inclinação em documentos. Técnicas baseadas na transformada de Hough, admitem que uma característica comum às regiões textuais é o alinhamento dos caracteres (lado-a-lado, formando uma linha) e o paralelismo das linhas de texto.

Seja (x, y) a localização em coordenadas cartesianas de um pixel preto na imagem, a transformada de Hough converte os pares de coordenadas cartesianas (x, y) em curvas nas coordenadas polares (ρ, θ) segundo a equação:

$$\rho = x \cos(\theta) + y \sin(\theta) \tag{1}$$

e um vetor auxiliar é utilizado como acumulador para computar a quantidade de vezes que cada combinação de ρ e θ aparece na imagem (Figura 5). Pixels pretos alinhados (formando linhas) na imagem geram picos no plano de Hough e permitem identificar o ângulo (θ) dessas linhas, ou seja, a inclinação do documento.

A granularidade da busca pelo ângulo depende da resolução do eixo θ , enquanto que a complexidade de tempo e espaço computacionais depende da quantidade de pontos a serem transformados e da resolução de θ .

O Algoritmo 2 apresenta os principais passos de um algoritmo genérico para detecção da inclinação utilizando o método baseado em transformada de Hough.

Métodos baseados na transformada de Hough são bastante populares pela robustez na detecção do ângulo de inclinação de linhas em uma imagem, mas demandam alto tempo

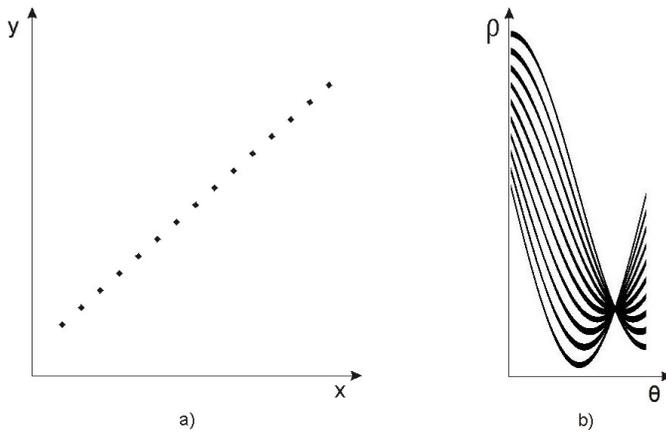


Figura 5. Transformação das coordenadas cartesianas para o plano de Hough.

Algoritmo 2 Algoritmo de Detecção de Inclinação baseado em Transformada de Hough

```

1: HOUGH(Imagem)
2: numeroDeLinhasDaImagem  $\leftarrow$  Size(Imagem.linhas)
3: numeroDeColunasDaImagem  $\leftarrow$  Size(Imagem.colunas)
4: for i = 1 to numeroDeLinhasDaImagem do
5:   for j = 1 to numeroDeColunasDaImagem do
6:     if Imagem(i, j) = preto then
7:       vetAcumulator  $\leftarrow$  HoughTransform(i, j)
8:     end if
9:   end for
10: end for
11: [maxarg,  $\theta_o$ ]  $\leftarrow$  max(vetAcumulator)
12: return  $\theta_o$ 

```

e espaço computacionais para encontrar o pico no plano de Hough. Além disso, a maioria das técnicas desenvolvidas assumem que o documento contém uma região mínima de texto impresso.

Devido a tais problemas, vários trabalhos baseados na transformada de Hough tentam melhorar a eficiência computacional do algoritmo reduzindo a quantidade de pontos a serem transformados para o plano de Hough. Essa redução acontece selecionando subconjuntos representativos da imagem ou restringindo a análise a partes do documento.

Le *et al.* [14] descreveram um algoritmo para identificação da inclinação e também da orientação (retrato ou paisagem) do documento. A detecção da orientação é obtida dividindo a imagem em sub-regiões e cada uma é classificada como contendo componentes textuais

ou não-textuais. Em seguida, essas regiões são classificadas como "retrato" ou "paisagem" de acordo com uma análise de projeção vertical e horizontal. A detecção de inclinação é realizada então, aplicando a transformada de Hough aos pixels pretos da última linha de cada componente conectado. Por utilizar-se de transformação no plano de Hough e análise de componentes conectados, a técnica de Le *et al.* sofre com alto custo computacional.

Amin e Fischer [17] aplicaram transformada de Hough à última linha pertencente a blocos segmentados de texto e agrupam pixels em componentes conectados para reduzir o custo computacional.

Singh *et al.* [5] elaboraram um estudo sobre a complexidade de métodos de detecção de inclinação baseados na transformada de Hough. Eles apresentam uma proposta de algoritmo para acelerar o cálculo da transformada de Hough que seleciona apenas alguns pontos de contornos dos componentes conectados na imagem para aplicação da transformada.

A transformada de Hough é eficiente mesmo em casos de imagens ruidosas. No entanto, a aplicação de etapas prévias utilizando algoritmos para extração de componentes conectados (com a finalidade de reduzir a complexidade computacional da técnica) torna a detecção de inclinação vulnerável a ruído. Para as técnicas descritas acima, os problemas inerentes ao uso da transformada permanecem, como a expectativa de haver texto no documento.

2.3 Técnicas Baseadas em Agrupamento de Vizinhos Mais Próximos

Os métodos baseados em agrupamento de vizinhos-mais-próximos formam uma outra categoria de métodos para detecção de inclinação que também explora os componentes textuais da imagem. Métodos nessa categoria assumem que os caracteres em uma linha encontram-se alinhados e próximos uns aos outros. Técnicas nessa categoria começam o processo de detecção da inclinação através do agrupamento dos pixels pretos na imagem em componentes conectados para identificação dos caracteres e, depois, agrupam esses caracteres em blocos maiores, de acordo com alguma medida de similaridade. Por fim, o ângulo de inclinação é calculado baseando-se em distâncias e outros relacionamentos espaciais entre as linhas de texto.

O Algoritmo 3 apresenta os principais passos de um algoritmo genérico para detecção da inclinação utilizando o método baseado em agrupamento de vizinhos mais próximos.

O algoritmo de Ávila e Lins [6] inicialmente busca os menores retângulos (*blocos*) que envolvem os caracteres e desenha linhas imaginárias que acompanham as linhas de texto desses blocos, como exemplificado na Figura 6. Essas linhas são usadas para detectar inclinação e orientação (retrato/paisagem).

Lu *et al.* [18] propuseram uma técnica para estimar o ângulo de inclinação observando

Algoritmo 3 Algoritmo de Detecção de Inclinação baseado em Agrupamento de Vizinhos mais Próximos

```

1:  $NN(Imagem)$ 
2:  $labels \leftarrow LABEL(Imagem)$ 
3:  $numeroDeGrupos \leftarrow MAX(labels)$ 
4: while tojoin do
5:   for  $i = 1$  to  $numeroDeGrupos$  do
6:      $group \leftarrow groups(i)$ 
7:      $closerGroup \leftarrow EscolheMaisProximo(groups, group)$  {função EscolheMaisProximo() encontra o grupo que é vizinho mais próximo}
8:      $groups \leftarrow Agrupa(groups, closerGroup, group)$  {função Agrupa() agrupa vizinhos próximos}
9:   end for
10: end while
11: for  $i = 1$  to  $numeroDeGrupos$  do
12:    $group \leftarrow groups(i)$ 
13:    $angles(i) \leftarrow encontraLinha(groups)$  {função encontraLinha() calcula a linha base dos caracteres do grupo}
14: end for
15:  $\theta_o \leftarrow calculaAngulo(angles)$  {função calculaAngulo() calcula o ângulo de inclinação}
16: return  $\theta_o$ 

```

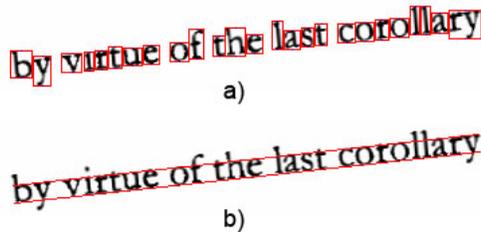


Figura 6. Agrupamento em linhas do algoritmo de Ávila e Lins: (a) detecção dos blocos dos componentes conexos e (b) construção da linha imaginária. Retirado de [6].

seqüências de pixels brancos que percorrem o espaçamento entre-linhas do texto. A orientação do documento é também detectada através da análise de ascendentes e descendentes.

O algoritmo de Shivakumara e Kumar [2] detecta regiões de texto na imagem com o agrupamento dos caracteres na imagem através do chamado “crescimento de bordas” (*boundary-growing*). Nessa técnica os retângulos contendo os caracteres (*bounding boxes*) são expandidos em uma largura calculada a partir da altura média dos caracteres. A interseção dessas expansões agrupam os caracteres em palavras. Linhas imaginárias para encontrar o ângulo de inclinação são traçadas seguindo os caracteres agrupados através de regressão linear.

Saragiotis e Papamarkos [20] utilizaram regras para filtrar os caracteres da imagem

baseando-se no tamanho e em características geométricas, como densidade de pixels pretos, área e proporção entre altura e largura dos componentes conectados. Dessa forma, apenas os caracteres são utilizados no cálculo da inclinação e componentes que não representem texto, como componentes gráficos, devem ser eliminados. Em seguida, os caracteres são agrupados em palavras de acordo com a distância entre eles para então serem agrupados em linhas de texto. Linhas imaginárias são traçadas na base das linhas de texto através de regressão linear. O ângulo de inclinação é calculado a partir dessas linhas. A novidade no trabalho de Saragiotis e Papamarkos é a possibilidade de detectar mais de um ângulo na imagem, um ângulo diferente para cada grupo de palavras.

Beusekom *et al.* [32] encontra inclinação e orientação baseando-se na idéia de que ascendentes ocorrem mais frequentemente que descendentes. A técnica proposta por Beusekom *et al.* busca otimizar o casamento de parâmetros de um modelo geométrico de linhas através de uma função de qualidade que indica o melhor conjunto de parâmetros.

Makridis *et al.* [33] apresentam uma técnica para detecção de inclinação global e local em documentos com coloridos. A imagem é segmentada em camadas de diferentes cores, binarizadas para uma posterior análise dos componentes conectados. Regiões de plano de fundo e figuras são removidos da análise e a inclinação é detectada traçando linhas partindo de cada caractere e agrupando-os em segmentos de linhas de texto.

Técnicas nesta categoria exigem a existência de uma região mínima de linhas de texto na imagem. Além disso, é importante ressaltar que a efetividade desse tipo de técnica é dependente da qualidade da binarização utilizada e do nível de ruído presente na imagem. Imagens ruidosas — como o exemplo da Figura 7 — podem sofrer interferência durante a detecção da inclinação. Já o custo computacional é associado à etapa de detecção dos componentes conectados e não é necessário testar vários ângulos para encontrar a inclinação correta.

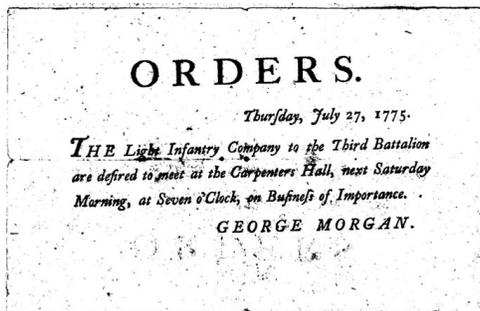


Figura 7. Exemplo de imagem ruidosa que pode interferir na detecção de inclinação.

2.4 Técnicas Baseadas em Correlação-Cruzada

As técnicas baseadas em correlação partem do princípio que o texto sem inclinação possui uma estrutura horizontal homogênea. No geral, as técnicas nessa categoria fazem a divisão da imagem em faixas verticais e calculam alguma medida de correlação entre faixas adjacentes para encontrar o ângulo de inclinação do documento.

O Algoritmo 4 apresenta os principais passos de um algoritmo genérico para detecção da inclinação utilizando o método baseado em correlação-cruzada.

Algoritmo 4 Algoritmo de Detecção de Inclinação baseado em Correlação-Cruzada

```

1: CC(Imagem, larguraFaixa)
2: numeroDeFaixas ← Imagem.cols/larguraFaixa
3: faixas(i) ← DivideImagem(Imagem, larguraFaixa) {função DivideImagem() faz divisão horizontal em faixas}
4: for i = 1 to numeroDeFaixas - 1 do
5:   faixa ← faixas(i)
6:   for d = 1 to dxmax do
7:     faixadx ← desloca(faixa, dx) {função desloca() faz deslocamento vertical da faixa}
8:     corl ← CORRELATION(faixadx, faixa(i + 1))
9:     correlations(i, d) ← corl
10:  end for
11: end for
12:  $\theta_o$  ← decideAngle(correlations)
13: return  $\theta_o$ 

```

Na técnica descrita por Akiyama e Hagita [21], a imagem é dividida em diversas faixas verticais de mesma largura e um perfil de projeção horizontal é computado para essas faixas. Calcula-se a correlação entre projeções adjacentes e, cada faixa é deslocada verticalmente até que a melhor correlação entre as faixas seja encontrada. Na Figura 8, F_1 , F_2 e F_3 são as faixas verticais do documento, L é o deslocamento que atinge maior correlação. O ângulo de inclinação θ é calculado como o arco-tangente da razão entre o deslocamento médio e a largura da faixa.

Gatos *et al.* [23] propuseram a extração da correlação das faixas verticais extraídas de uma imagem pré-processada por um filtro de suavização (*smoothing*) horizontal. A largura das faixas foi determinada experimentalmente sendo igualmente espaçadas e para cada faixa uma “assinatura” é gerada, atribuindo valor 1 para linhas que contenham pelo menos 1 pixel preto. A seguir, a correlação é extraída de um par de assinaturas.

No trabalho desenvolvido por Avanindra [24], ao invés da correlação ser calculada para a imagem inteira, ela é calculada sobre pequenas regiões selecionadas aleatoriamente, sem a necessidade de uma segmentação prévia para distinguir as regiões textuais de figuras. Avanindra utilizou a amostragem de Monte Carlo para selecionar regiões sobre as quais devem se realizar o cálculo da correlação.

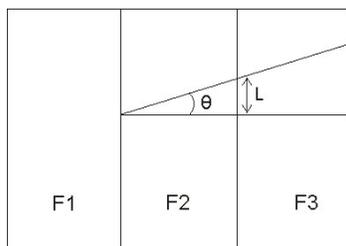


Figura 8. Cálculo do ângulo de inclinação θ a partir do deslocamento que resulta em maior correlação entre as faixas.

Técnicas baseadas em correlação-cruzada geralmente exigem a existência de uma região mínima de texto para o bom funcionamento da técnica, que garantem a homogeneidade horizontal da imagem e são sensíveis a diagramações muito complexas.

2.5 Técnicas Baseadas em Paralelogramos

A técnica baseada em paralelogramos foi recentemente introduzida por Chou *et al.* [25] e baseia-se na idéia de que um documento é composto por uma combinação de objetos retangulares, tais como linhas de texto, figuras, tabelas, etc.

A idéia do algoritmo de detecção de inclinação proposto por Chou *et al.* é construir paralelogramos em vários ângulos e decidir qual dentre eles melhor se adapta aos objetos na imagem. A Figura 9 apresenta exemplos da construção de paralelogramos. Pode ser observado que, quando o ângulo de inclinação está próximo ao ângulo real, os paralelogramos se adaptam melhor aos componentes da imagem e a região branca de plano de fundo é maior.

O algoritmo de Chou *et al.*, realiza uma busca iterativa por ângulos entre -15° e $+15^\circ$: a cada iteração a busca é refinada a uma vizinhança próxima do ângulo encontrado na iteração anterior. O ângulo de inclinação escolhido a cada iteração é aquele que maximiza o tamanho da região branca de plano de fundo. O tamanho dessa região branca é calculado contabilizando o número de *seções* brancas — segmentos de retas utilizados para desenhar os paralelogramos.

A vantagem da técnica dos paralelogramos é a capacidade de lidar com diversos tipos de diagramações, sendo independente também do idioma existente no texto, pois não se baseia na análise minuciosa de características dos caracteres. Além disso, não exige a existência de uma área mínima de texto plano, podendo encontrar a inclinação de documentos contendo pouco (ou nenhum) texto.

O Algoritmo 5 apresenta os principais passos de um algoritmo genérico para detecção

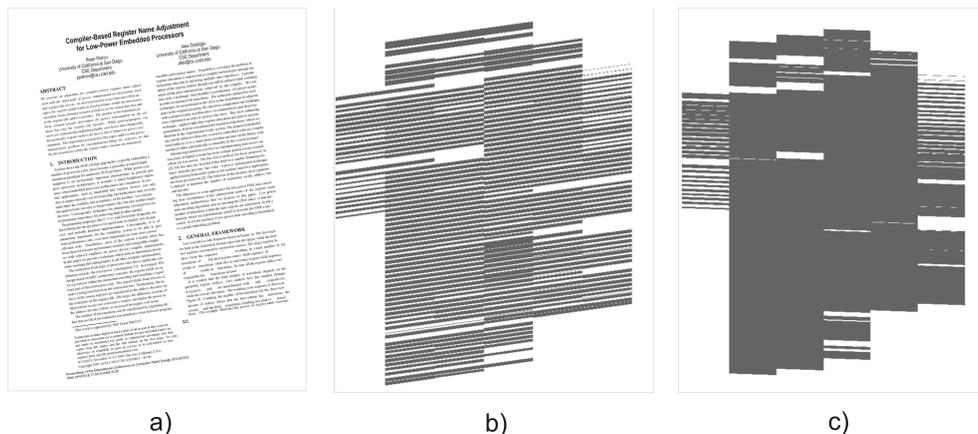


Figura 9. (a) Uma imagem com $+8^\circ$ de ângulo de inclinação. (b) Paralelogramos construídos a $+8^\circ$. (c) Paralelogramos construídos a -2° .

da inclinação utilizando o método dos paralelogramos.

Algoritmo 5 Algoritmo de Detecção de Inclinação baseado em Paralelogramos

```

1: PARALELOGRAMOS(Imagem)
2: for  $\theta = \theta_1$  to  $\theta_2$  do
3:   parallelograms  $\leftarrow$  DesenhaParalelogramos(Imagem,  $\theta$ ) {função DesenhaParalelogramos() plota
   linhas a um dado ângulo }
4:   vector(i)  $\leftarrow$  calculaBranco(parallelograms) {função calculaBranco() mede o tamanho da região
   branca da imagem de paralelogramos}
5: end for
6:  $\theta_o \leftarrow \max(\mathit{vector})$ 
7: return  $\theta_o$ 

```

Mascaro e Cavalcanti [26] e Mascaro *et al.* [34] apresentam uma evolução do algoritmo original proposto por Chou *et al.* calculando o tamanho da região branca utilizando a área: quantidade total de pixels brancos. Essa modificação permitiu reduzir o espaço de busca pelo ângulo, acelerando a computação em até 50% do tempo necessário para imagens digitalizadas. Mascaro *et al.* ainda propõem uma abordagem para reduzir a sensibilidade da técnica a componentes não-textuais, como separadores verticais e ruído.

Uma desvantagem da técnica, é a rigidez na subdivisão da imagem em *slabs* de tamanho previamente fixado. Formas de determinar um tamanho de *slab* automaticamente ainda não foram exploradas. Isso pode levar à vulnerabilidade do algoritmo para documentos com diversas colunas que não possuem alinhamento horizontal entre elas.

2.6 Outros Métodos para Detecção de Inclinação

Outros métodos são ainda divulgados na literatura sem que pertençam a uma das classes listadas anteriormente. Sauvola e Pietikainen [27] propuseram uma técnica para detecção de inclinação tratando a imagem do documento como regiões de textura e analisando a direção do gradiente de sub-regiões da imagem. A análise proposta por Sauvola e Pietikainen assume que a imagem possui característica homogênea na horizontal, geralmente presente em regiões de texto, podendo sofrer com múltiplas colunas, com grande quantidade de regiões não textuais e diagramações mais complexas.

Chen e Haralick [29] apresentaram um algoritmo que utiliza operações morfológicas de abertura e fechamento [31]. Uma operação recursiva de fechamento com elementos estruturantes de 2×2 ou 3×3 pixels é executada com a finalidade de conectar caracteres, palavras e outros componentes. A robustez da técnica de Chen e Haralick depende da escolha de um elemento estruturante adequado a diferentes tamanhos de fontes, podendo ser vulnerável a algumas diagramações.

Aghajan *et al.* [30] propuseram uma interpretação do problema de inclinação como um problema da determinação da direção de ondas eletromagnéticas. O algoritmo de Aghajan é capaz de detectar múltiplos ângulos de inclinação na imagem, sendo robusto à presença de fórmulas ou diagramas no documento.

O segundo método proposto por Postl em [35] calcula a transformada de Fourier bi-dimensional [31] da imagem do documento e analisa o espectro de potência para calcular o ângulo de inclinação do documento.

2.7 Técnicas para Detecção de Inclinação em Documentos Manuscritos

Embora a detecção de inclinação de documentos impressos tenha sido vastamente explorada, a detecção de inclinação de documentos manuscritos continua pouco investigada.

Chin *et al.* [36] investigaram algumas técnicas, tais como a transformada de Hough e análise de projeção de histogramas, para determinar qual método é mais adequado para lidar com trechos de texto manuscrito. O autor concluiu que a transformada de Hough é o método mais eficaz para aplicação em documentos manuscritos, mas que técnicas mais rápidas podem ser empregadas na detecção da inclinação de manuscritos.

Kapoor *et al.* [37] utilizaram a transformada de Radon confrontando-a com a transformada de Hough para análise da inclinação em palavras manuscritas de vários escritores diferentes. Analisaram que a transformada de Hough detecta mais picos no plano de Hough, no entanto, é mais custosa em termos de tempo computacional. Além disso, a transformada de Radon apresentou melhor desempenho sobre imagens com ruído.

O trabalho de Kavallieratou *et al.* [38] apresenta um método baseado no perfil de projeção horizontal e distribuição de *Wigner-Ville* do documento. O objetivo é lidar com documentos impressos e manuscritos simultaneamente e detectar um ângulo global na imagem.

Roy *et al.* [39] propuseram uma abordagem para correção da inclinação utilizando uma prévia segmentação do texto manuscrito. Roy *et al.* avaliam algumas características, tais como algumas distâncias medidas dentro de um único caractere e distância entre os caracteres e utiliza palavras *Bangla* como um estudo de caso de caracteres cursivos.

3 Considerações Finais

Com a alta demanda pela digitalização e reconhecimento de documentos, cresce a necessidade por melhores taxas de reconhecimento dos Sistemas de Reconhecimento Automático de Documentos. A inclinação é um problema que interfere em todo o processo de reconhecimento, sendo essencial para melhorar tais taxas e, por isso, muito estudo tem sido empregado no desenvolvimento de algoritmos capazes de detectar e corrigir essa inclinação.

O problema de detecção de inclinação há muito começou a ser explorado, mas continua como área de pesquisa ativa na busca por métodos com baixa complexidade computacional e robustez no processamento de imagens de documentos com quaisquer tipo de diagramação. As técnicas previamente divulgadas na literatura apresentam problemas de alto custo de memória ou tempo computacional ou, ainda, possuem alguma restrição quanto ao tipo de documento de entrada. Algumas técnicas requerem a presença de uma mínima região de linhas de texto para que o algoritmo funcione corretamente e são sensíveis à presença de ruído e outras componentes não-textuais, como figuras, tabelas e elementos manuscritos.

Embora a detecção de inclinação em documentos impressos esteja bastante evoluída, poucos trabalhos foram divulgados com o objetivo de detectar vários ângulos em uma única imagem de documento. Além disso, desenvolveram-se poucos trabalhos para detecção de inclinação em documentos manuscritos, devido à dificuldade natural do problema, dada a alta variabilidade do texto.

Referências

- [1] C. Mello, A. Sanchez, A. Oliveira, and A. Lopes. An efficient gray-level thresholding algorithm for historic document images. *Journal of Cultural Heritage*, 9(2):109–116, 2008.
- [2] P. Shivakumara and G. Hemantha Kumar. A novel boundary growing approach for accurate skew estimation of binary document images. *Pattern Recognition Letters*, 27(7):791–801, 2006.

- [3] R. Cattoni, T. Coianiz, S. Messelodi, and C. Modena. Geometric layout analysis techniques for document image understanding: a review. *ITC-IRST Technical Report*, 9703(09), 1998.
- [4] J. Hull. Document image skew detection: Survey and annotated bibliography. *Document Analysis Systems II*, pages 40–64, 1998.
- [5] C. Singh, N. Bhatia, and A. Kaur. Hough transform based fast skew detection and accurate skew correction methods. *Pattern Recognition*, 41(12):3528–3546, 2008.
- [6] B. Ávila and R. Lins. A fast orientation and skew detection algorithm for monochromatic document images. In *ACM symposium on Document Engineering*, pages 118–126, 2005.
- [7] W. Postl. Detection of linear oblique structures and skew scans in digitized documents. In *International Conference on Pattern Recognition*, pages 687–689, 1986.
- [8] H. Baird. The skew angle of printed documents. In *Proceedings of Society of Photographic Scientists and Engineers*, volume 940, pages 21–24, 1987.
- [9] G. Ciardiello, G. Scafuro, M. Degrandi, M. Spada, and M. Roccotelli. An experimental system for office document handling and text recognition. In *International Conference on Pattern Recognition*, pages 739–743, 1988.
- [10] J. Kanai and A. Bagdanov. Projection profile based skew estimation algorithm for JBIG compressed images. *International Journal on Document Analysis and Recognition*, 1(1):43–51, 1998.
- [11] Y. Ishitani. Document skew detection based on local region complexity. In *International Conference on Document Analysis and Recognition*, pages 49–52, 1993.
- [12] S. Li, Q. Shen, and J. Sun. Skew detection using wavelet decomposition and projection profile analysis. *Pattern Recognition Letters*, 28(5):555–562, 2007.
- [13] S. Hinds, J. Fisher, and D. D’Amato. A document skew detection method using run-length encoding and the Hough transform. In *International Conference on Pattern Recognition*, volume 1, pages 464–468, 1990.
- [14] D.S. Le, G.R. Thoma, and H. Wechsler. Automated page orientation and skew angle detection for binary document images. *Pattern Recognition*, 27(10):1325–1344, 1994.
- [15] Y. Min, S. Cho, and Y. Lee. A data reduction method for efficient document skew estimation based on Hough transformation. In *International Conference of Pattern Recognition*, volume 96, pages 732–736, 1996.

- [16] U. Pal and B. Chaudhuri. An improved document skew angle estimation technique. *Pattern Recognition Letters*, 17(8):899–904, 1996.
- [17] A. Amin and S. Fisher. A Document Skew Detection Method Using the Hough Transform. *Pattern Analysis and Applications*, 3(3):243–253, 2000.
- [18] S. Lu, J. Wang, and C. Tan. Fast and Accurate Detection of Document Skew and Orientation. In *International Conference on Document Analysis and Recognition*, volume 2, pages 684–688, 2007.
- [19] R. Smith. A simple and efficient skew detection algorithm via text row accumulation. In *International Conference on Document Analysis and Recognition*, volume 2, pages 1145–1148, 1995.
- [20] P. Saragiotis and N. Papamarkos. Local skew correction in documents. *International Journal of Pattern Recognition and Artificial Intelligence*, 22(4):691–710, 2008.
- [21] T. Akiyama and N. Hagita. Automated entry system for printed documents. *Pattern Recognition*, 23(11):1141–1154, 1990.
- [22] H. Yan. Skew correction of document images using interline cross-correlation. *CVGIP: Graphical models and image processing*, 55(6):538–543, 1993.
- [23] B. Gatos, N. Papamarkos, and C. Chamzas. Skew detection and text line position determination in digitized documents. *Pattern Recognition*, 30(9):1505–1519, 1997.
- [24] S. Avanimra. Robust detection of skew in document images. *IEEE Transactions on Image Processing*, 6(2):344–349, 1997.
- [25] C. Chou, S. Chu, and F. Chang. Estimation of skew angles for scanned documents based on piecewise covering by parallelograms. *Pattern Recognition*, 40(2):443–455, 2007.
- [26] Angélica A. Mascaro and George D. C. Cavalcanti. Estimating the skew angle of scanned document through background area information. In *Brazilian Symposium on Computer Graphics and Image Processing*, pages 87–94, 2008.
- [27] J. Sauvola and M. Pietikainen. Skew angle detection using texture direction analysis. In *Scandinavian Conference on Image Analysis*, pages 1099–1106, 1995.
- [28] C. Sun and D. Si. Skew and slant correction for document images using gradient direction. In *Document Analysis and Recognition*, volume 1, pages 1871–1879, 1997.
- [29] S. Chen and R. Haralick. An automatic algorithm for text skew estimation in document images using recursive morphological transforms. In *International Conference on Image Processing*, pages 13–16, 1994.

- [30] B. Khalaj H. Aghajan and T. Kailath. Estimation of skew angle in text-image analysis by SLIDE: Subspace-based line detection. *Machine Vision and Applications*, 7(4):267–276, 1994.
- [31] R. Gonzalez and R. Woods. *Processamento Digital de Imagens*. Pearson, 3a edição edition, 2010.
- [32] J. van Beusekom, F. Shafait, and T.M. Breuel. Combined orientation and skew detection using geometric text-line modeling. *International Journal on Document Analysis and Recognition*, 13(2):79–92, 2010.
- [33] M. Makridis, N. Nikolaou, and N. Papamarkos. An adaptive technique for global and local skew correction in color documents. *Expert Systems with Applications*, 2010.
- [34] Angélica A. Mascaro, George D. C. Cavalcanti, and Carlos A. B. Mello. Fast and robust skew estimation of scanned documents through background area information. *Pattern Recognition Letters*, 31(11):1403–1411, 2010.
- [35] W. Postl. Method for automatic correction of character skew in the acquisition of a text original in the form of digital scan results, february 2 1988. US Patent 4,723,297.
- [36] W. Chin, A. Harvey, and A. Jennings. Skew detection in handwritten scripts. In *Conference on Speech and Image Technologies for Computing and Telecommunications*, pages 319–322, 1997.
- [37] D. Bagai R. Kapoor and T. Kamal. A new algorithm for skew detection and correction. *Pattern Recognition Letters*, 25(11):1215–1229, 2004.
- [38] N. Fakotakis E. Kavallieratou and G. Kokkinakis. Skew angle estimation for printed and handwritten documents using the Wigner–Ville distribution. *Image and Vision Computing*, 20(11):813–824, 2002.
- [39] S. Parui A. Roy, T. Bhowmik and U. Roy. A Novel Approach to Skew Detection and Character Segmentation for Handwritten Bangla Words. In *Digital Image Computing: Techniques and Applications*, pages 30–30, 2005.