

# Algoritmos para alinhamento automático de imagens de documentos

Bruno Sobreira França (217787)

Abril 2024

## 1 Introdução

A análise de documentos digitalizados visa converter a informação contida em imagens de documentos em texto editável. Um problema comum na digitalização é o desalinhamento do documento, que ocorre devido à inclinação do papel durante a digitalização. A correção dessa inclinação é essencial para o funcionamento eficaz de sistemas de reconhecimento ótico de caracteres (OCR). Neste trabalho, são implementados e analisados dois algoritmos para a detecção e correção da inclinação: um baseado na projeção horizontal e outro na transformada de Hough.

## 2 Descrição dos Algoritmos

Este trabalho foca em dois métodos principais para a correção de inclinação de documentos:

### 2.1 Técnica Baseada na Projeção Horizontal

A técnica de projeção horizontal envolve a variação do ângulo testado e a projeção da quantidade de pixels pretos em cada linha de texto. O ângulo que maximiza uma função objetivo, calculada sobre o perfil da projeção horizontal, é escolhido como o ângulo de correção. A função objetivo adotada é a soma dos quadrados das diferenças entre valores em células adjacentes do perfil de projeção.

A implementação desta técnica envolve os seguintes passos:

- **Binarização:** A imagem é binarizada, utilizando um método de limiarização, como o método de Otsu, para separar os pixels de texto (pretos) do fundo (branco).
- **Projeção Horizontal:** Para cada ângulo de rotação dentro de um intervalo específico (por exemplo, de -90 a 90 graus), a imagem é rotacionada e a soma dos pixels pretos em cada linha é calculada, resultando em um perfil de projeção horizontal.
- **Cálculo da Função Objetivo:** A função objetivo, nest caso a soma dos quadrados das diferenças entre valores em células adjacentes do perfil de projeção, é calculada para cada ângulo. Este cálculo avalia o alinhamento do texto;
- **Determinação do Melhor Ângulo:** O ângulo que maximiza a função objetivo é identificado como o ângulo de correção ideal.
- **Rotação da Imagem:** A imagem original é então rotacionada pelo ângulo de correção para alinhar corretamente o texto.

Essa técnica se baseia na premissa de que, quando o texto está corretamente alinhado, as linhas de texto são horizontais e, portanto, a soma dos pixels pretos ao longo das linhas deve apresentar menos variação entre linhas adjacentes.

## 2.2 Técnica Baseada na Transformada de Hough

A técnica baseada na transformada de Hough utiliza a suposição de que os caracteres de texto estão alinhados em linhas. A transformada de Hough converte coordenadas da imagem em curvas nas coordenadas polares, onde pixels pretos alinhados geram picos, permitindo a identificação do ângulo de inclinação.

A implementação da técnica baseada na transformada de Hough em etapas para detectar e corrigir a inclinação de documentos digitalizados:

- **Detecção de Bordas:** Inicialmente, a imagem é processada para detectar bordas usando o algoritmo de detecção de bordas de Canny. Esta etapa é crucial para identificar as linhas de texto nos documentos.
- **Transformada de Hough:** A transformada de Hough é então aplicada às bordas detectadas. Esta etapa converte as coordenadas da imagem em um espaço de parâmetros de linha, onde cada pixel de borda contribui para todas as possíveis linhas que podem passar por ele. Isso é alcançado por meio de uma acumulação em uma matriz de votação.
- **Identificação da Linha Dominante:** Após a transformada de Hough, a linha dominante é identificada na matriz de votação. Isso geralmente é feito encontrando os picos significativos na matriz, que correspondem às linhas dominantes na imagem.
- **Cálculo do Ângulo de Inclinação:** Com a linha dominante identificada, o ângulo de inclinação do documento é calculado a partir dos parâmetros da linha. Este ângulo é então utilizado para corrigir a inclinação da imagem original.

Essa abordagem se baseia na intuição de que as linhas de texto em um documento digitalizado fornecem pistas valiosas sobre sua orientação e inclinação. Ao explorar essas linhas utilizando a transformada de Hough, é possível recuperar informações sobre a inclinação do documento e aplicar correções adequadas para melhorar a legibilidade e a precisão do reconhecimento de caracteres.

## 3 Implementação

Para a implementação dos algoritmos de correção de inclinação, foram desenvolvidos dois scripts em Python 3.10: `Horizontal_projection.py` e `Hough_transformation.py`. Ambos os scripts utilizam as seguintes bibliotecas:

- **OpenCV:** Para operações de processamento de imagem.
- **NumPy:** Para manipulação de arrays.
- **Matplotlib:** Para visualização do conteúdo gerado.
- **ptesseract:** Para aplicar OCR

Além disso, as seguintes bibliotecas são usadas para tratamento de argumentos de linha de comando e operações de sistema de arquivos: `Getopt`, `Sys` e `Os`.

### 3.1 Alignment\_utils.py

O módulo `alignment_utils.py` fornece funções auxiliares comuns entre os scripts de correção de inclinação, facilitando operações de rotação de imagem, comparação de resultados de OCR e exibição de mensagens de saída informativas. A seguir, detalha-se cada uma dessas funções:

- A função `rotate_image` é responsável por rotacionar uma imagem pelo ângulo especificado sem cortar o conteúdo. Ela calcula as novas dimensões da imagem rotacionada para garantir que o conteúdo não seja perdido após a rotação. Primeiro, obtém as dimensões da imagem original e calcula as novas dimensões usando a trigonometria básica com o ângulo de rotação. Em seguida, gera a matriz de rotação apropriada e aplica a transformação à imagem usando a função `cv.warpAffine` do OpenCV.
- A função `compare_ocr_tesseract` realiza uma comparação entre os resultados de OCR obtidos de duas imagens utilizando o Tesseract. Esta função lê as imagens de entrada e saída, aplica o OCR para extrair o texto e compara os textos extraídos, linha por linha, usando a biblioteca `diffib`. O resultado da comparação é exibido, destacando as diferenças entre os textos OCR das duas imagens. Adicionalmente, a função pode salvar o resultado da comparação em um arquivo especificado pelo usuário.
- A função `print_output_message` gera uma mensagem de saída informativa sobre o processamento realizado. Ela imprime o caminho das imagens de entrada e saída, o melhor ângulo de rotação encontrado e a heurística utilizada para determinar este ângulo. Esta função é útil para fornecer feedback ao usuário sobre o sucesso e os detalhes do processo de correção de inclinação.

### 3.2 Hough\_transformation.py

O script `Hough_transformation.py` implementa a técnica de correção de inclinação baseada na transformada de Hough, que identifica linhas dominantes na imagem para determinar o ângulo de correção adequado. Nesta subseção, é explorado a implementação deste algoritmo passo a passo.

A função `main` gerencia o fluxo do script. Ela analisa os argumentos de linha de comando para determinar os caminhos das imagens de entrada e saída, bem como flags opcionais para visualização, desenho da linha dominante detectada e comparação de OCR. A imagem de entrada é carregada e a sua linha dominante é identificada usando a função `objective_function`. O ângulo de correção é então transformado de radiano para graus. A imagem é rotacionada pelo ângulo obtido e salva no caminho especificado como saída. Se as flags opcionais forem ativadas, uma versão da imagem com as linhas detectadas desenhadas também é salva, as imagens são exibidas usando Matplotlib e um `diff` do OCR da imagem de entrada e saída é feito.

Propriamente a função `objective_function` implementa a detecção da linha principal usando a transformada de Hough. Ela começa detectando as bordas na imagem usando o algoritmo de detecção de bordas de Canny. Em seguida, a transformada de Hough é aplicada às bordas detectadas para identificar as linhas presentes na imagem. Por fim é recuperado a linha que possui a célula com acumulador máximo no espaço de hough.

A fim de visualizar a linha principal detectada na imagem, foi escrito a função `draw_detected_line`. Ela recebe a imagem original e as linhas detectadas como entrada e desenha a linha sobre a imagem. Isso é feito convertendo as coordenadas polares das linhas para coordenadas cartesianas e usando a função `line`, do `opencv`, para desenhar as linhas na imagem.

### 3.3 Horizontal\_projection.py

O script `Horizontal_projection.py` implementa a técnica de projeção horizontal para a correção de inclinação de documentos digitalizados. Esta subseção detalha a implementação passo a passo, explicando as funções e o fluxo do script.

A função `main` gerencia o fluxo do script. Ela analisa os argumentos de linha de comando para determinar os caminhos das imagens de entrada e saída, bem como flags opcionais para visualização, geração de histogramas e comparação de OCR. Inicialmente, a imagem de entrada é carregada, invertida através do `bitwise_not` e binarizada usando a técnica de limiarização de Otsu. Em seguida, o ângulo de correção é determinado usando a função `find_best_rotation_angle`, explanada ao futuramente nesta subseção. A imagem é rotacionada por este ângulo e salva no caminho especificado como saída. Se as flags opcionais forem ativadas, os histogramas da soma dos pixels pretos em cada linha da imagem são gerados, as imagens são exibidas usando Matplotlib e é comparado OCR da imagem de entrada e saída.

Precisamente, a função `find_best_rotation_angle` percorre uma faixa de ângulos de 0 a 180 graus e calcula a função objetivo para cada um deles. Os valores resultantes são armazenados em um array, e o índice do valor máximo é identificado como o ângulo de correção ideal. Se o índice do valor máximo for maior que 90, o ângulo é ajustado para estar na faixa de -90 a 0; caso contrário, estará entre 0 e 89 graus. Isso é aplicado para evitar que as imagens de teste fiquem de cabeça para baixo. Essas janelas foram definidas presumindo que imagens rotacionadas no sentido horário não ultrapassarão os 90 graus, e aquelas rotacionadas no anti-horário não ultrapassarão os -90 graus. Isso não pode ser generalizado, pois, por exemplo, imagens rotacionadas no sentido horário que ultrapassarem os 90 graus serão postas de cabeça para baixo.

A função `objective_function` é responsável por calcular uma métrica que avalia o alinhamento do texto na imagem para um dado ângulo de rotação. A imagem é rotacionada utilizando a função `rotate_image` da biblioteca auxiliar e, em seguida, a soma dos pixels pretos em cada linha (perfil de projeção horizontal) é calculada. A função objetivo é definida como a soma dos quadrados das diferenças entre valores adjacentes desse perfil. Esse valor é usado para determinar quão bem alinhado está o texto para o ângulo testado.

## 4 Análise dos Resultados

As imagens de entrada utilizadas nos testes estão presentes no diretório `input`. Os produtos dos scripts `horizontal_projection.py` e `hough_transformation.py`, analisados nesta seção, encontram-se respectivamente nos diretórios `output/projection` e `output/line`.

Para realizar uma análise do comportamento de um sistema de OCR (Optical Character Recognition), o Tesseract foi aplicado nas imagens de entrada e saída utilizadas na execução dos scripts. Os resultados obtidos serão discutidos a seguir nesta análise, assim como uma avaliação do desempenho dos scripts desenvolvido na tarefa de alinhamento.

### 4.1 Horizontal\_projection.py

Os resultados do programa `horizontal_projection.py` estão presentes na pasta `output/projection/`. Observando os resultados contidos neste diretório, como na Figura 6, gerada a partir da imagem presente na Figura 1, pode-se ver que o ângulo de correção obtido pela técnica de projeção horizontal é eficiente para alinhar corretamente o conteúdo do documento.

Este método mostrou-se particularmente eficaz para documentos com texto bem estruturado e linhas horizontais bem definidas. A técnica de projeção horizontal, ao maximizar a função objetivo

baseada na soma dos quadrados das diferenças entre valores adjacentes do perfil de projeção, permite uma estimativa do ângulo de inclinação. Ao observar a Figura 3, na qual o perfil de projeção da imagem 1 é representado por meio de um histograma, nota-se que a soma dos pixel pretos de cada linha estão muito distribuídos ao longo da imagem. Ao corrigir a inclinação da imagem, observa-se que o perfil de projeção resultante, Figura 7, mostra uma clara definição das linhas de texto.

No entanto, é importante notar que a eficácia deste método pode ser limitada em documentos onde as linhas de texto não são predominantemente horizontais ou onde o texto está distribuído de maneira irregular. Por exemplo, documentos com layouts complexos, gráficos ou tabelas podem não se beneficiar tanto desta abordagem, pois a variação na projeção horizontal pode não refletir fielmente a inclinação geral do texto

## 4.2 Hough\_transformation.py

Os resultados do programa `hough_transformation.py` estão presentes na pasta `output/line/`. Observando os resultados contidos neste diretório, como na Figura 4, gerada a partir da imagem presente na Figura 1, pode-se presumir que o ângulo da linha que possui a célula com o acumulador máximo no espaço de Hough descreve muito bem a inclinação do conteúdo de um documento. Portanto nota-se a possibilidade de utilizar o valor deste ângulo para corrigir a rotação do conteúdo de documentos digitalizados.

Este resultado ocorre devido ao fato de que, em um documento, o conteúdo geralmente está estruturado de maneira alinhada, sendo frequentemente puramente textual. Intui-se, então, que uma grande quantidade de pixels pretos que formam a imagem estão alinhados. Esses pixels, no espaço de Hough, descrevem linhas em coordenadas polares. Essas linhas possivelmente podem ditar a inclinação da imagem. Na aplicação destes conceitos, observou-se nos experimentos a possibilidade de utilizar a linha dominante do espaço de Hough, isto é, aquela que possui a célula com o acumulador máximo nesse domínio, para descrever o ângulo de inclinação do conteúdo textual nas imagens de teste. Um exemplo de uma linha dominante detectada nos experimentos está presente na Figura 5. O ângulo dessa linha foi utilizado para corrigir a rotação da imagem presente na Figura 1 e gerar a imagem presente na Figura 4. Nota-se, então, que esse método pode ser de fato utilizado para corrigir a inclinação do conteúdo de texto em imagens.

## 4.3 Tesseract

Para aplicar o Tesseract nas imagens, a função `compare_ocr_tesseract` foi implementada no arquivo `alignment_utils.py`, utilizando a biblioteca Python `pytesseract`, um wrapper para o projeto Tesseract. Ambos os algoritmos `Hough_transformation.py` e `Horizontal_projection.py` utilizam essa função para comparar o OCR obtido pelas imagens de entrada e de saída.

Ao aplicar o Tesseract nas imagens de entrada, notou-se que, no conjunto de imagens de teste, ele foi capaz de reconhecer os caracteres apenas em imagens com inclinação entre -4 e 4 graus. Em imagens deste tipo, fora desse intervalo, ele não apresentou resultados, por exemplo ao aplicá-lo na imagem 1, com ângulo de -14 graus não possível recuperar nenhum caractere, como pode ser observado no listing ???. Mesmo dentro desse intervalo de inclinação, o conteúdo detectado não foi igual àquele gerado após a aplicação de ambos os métodos de rotação, resultados esses que podem ser vistos nos arquivos, `output/line/neg_4_rotated_hl_ocr.txt` e `output/project/neg_4_rotated_hp_ocr.txt`

Interessante notar que, para a imagem da Figura 1, o alinhamento fornecido pelo algoritmo `Hough_transformation.py` não permitiu ao Tesseract recuperar muitos caracteres presentes no documento, enquanto o `Horizontal_projection.py` teve maior êxito nessa tarefa. Isso provavelmente ocorre porque o método de Hough fornece um ângulo levemente diferente do método de projeção ho-

rizional. Mesmo que seja imperceptível ao comparar visualmente as Figuras 4 e 6, essa leve diferença de inclinação é prejudicial para o Tesseract, como pode ser observar nas listagens 2 e 3.

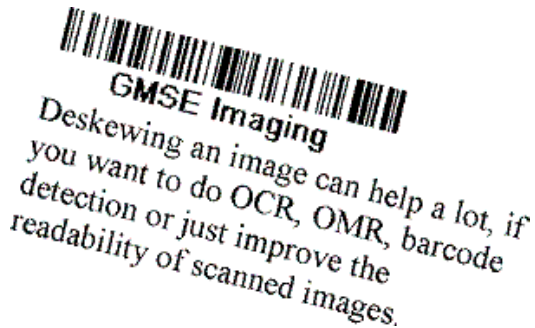


Figura 1: Imagem input/sample1.png, cujo o conteúdo possui uma inclinação de -14 graus

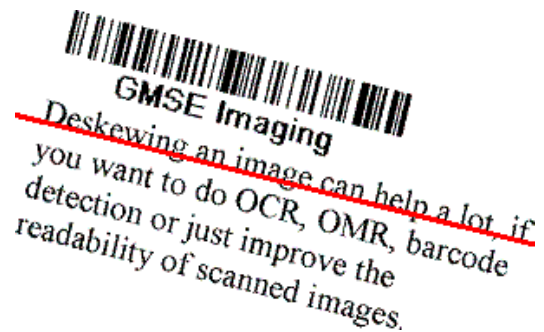


Figura 2: Imagem input/sample1\_with\_lines.png com linha horizontal dominante desenhada sobre o texto inclinado

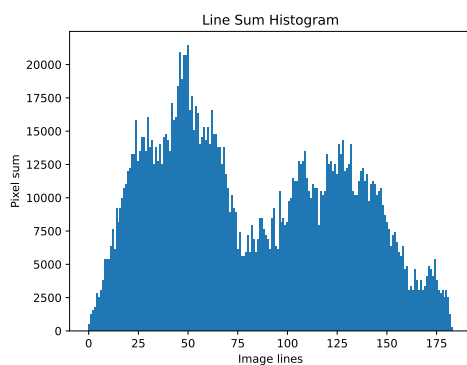


Figura 3: Histograma do perfil de projeção horizontal da imagem sample1.png. A soma dos pixels pretos por linha é distribuída ao longo da imagem inclinada

```
Empty page!!
Empty page!!
```

Listing 1: Saída do Tesseract OCR para a imagem da figura 1

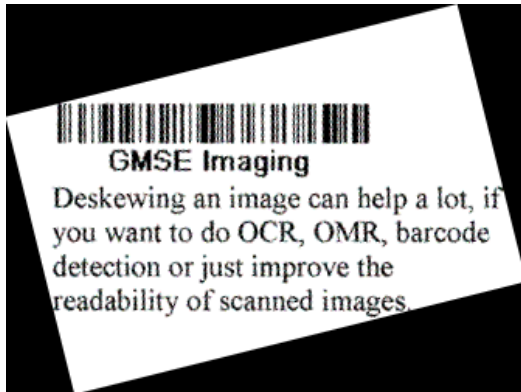


Figura 4: Imagem output/projection/sample1\_rotated\_hl.png, gerada a partir da imagem da figura 1 rotacionada pelo método da transformada de hough

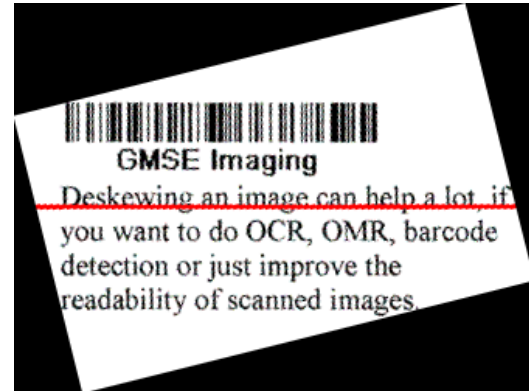


Figura 5: Imagem output/projection/sample1\_rotated\_hl\_with\_lines.png cujo é a a imagem da figura 1 rotacionada, com linha dominante desenhada sobre o texto alinhado corretamente

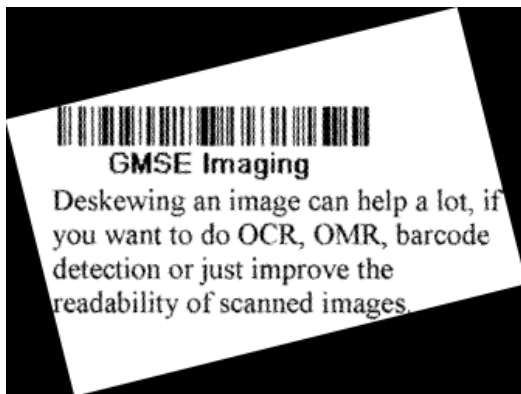


Figura 6: Imagem output/projection/sample1\_rotated\_hp.png com o conteúdo do texto alinhado horizontalmente após correção de inclinação através do método da projeção horizontal

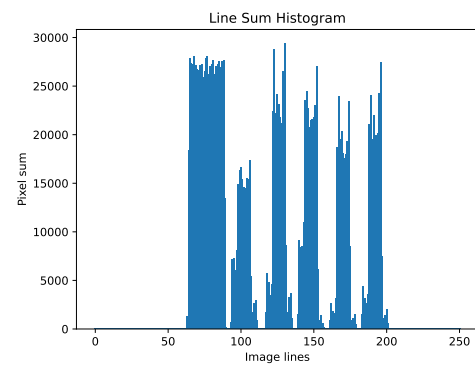


Figura 7: Histograma do perfil de projeção horizontal da imagem rotacionada sample1\_rotated\_hp.png, figura 6. A soma dos pixels pretos por linha mostra uma clara definição das linhas de texto

```
/Estimating resolution as 215  
Detected 18 diacritics  
Ea  
  
E Imaging
```

Listing 2: Saída do Tesseract OCR para a imagem rotacionada sample1\_rotated\_hl.png, figura 4

```
Estimating resolution as 215  
AUNQUE  
GMSE Imaging  
wing an image can help a lot, if  
nt to do OCR, OMR, barcode  
improve the  
  
ability of scanned i -_
```

Listing 3: Saída do Tesseract OCR para a imagem rotacionada sample1\_rotated\_hp.png, figura 6