

Universidade Estadual de Campinas

Instituto de Computação

Introdução ao Processamento Digital de Imagem (MC920A)

Professor: Hélio Pedrini

## **Relatório – Trabalho 3**

Gabriel Augusto Bertolino Gomes

RA: 248287

24/05/2024

## 1. Introdução

Ao realizar o procedimento de scanner em uma imagem, essa imagem pode vir ligeiramente desalinhada, com alguma inclinação. Essa inclinação não é desejada e, por isso, deve ser corrigida.

Neste trabalho, serão empregadas duas técnicas para executar tal operação. A primeira delas é a chamada projeção horizontal, que testa diversas rotações possíveis e escolhe a que possui o melhor. Enquanto a segunda técnica utiliza da transformada de Hough, que encontra as retas presentes na imagem e a partir disso, determina a inclinação do texto.

## 2. Materiais e métodos

Os materiais usados para realizar os alinhamentos das imagens foram:

- Python 3.11.6
  - Bibliotecas
    - scikit-image 0.23.2
    - numpy 1.26.1
    - matplotlib 3.8.4
- Visual Studio Code
- Imagens usadas: neg\_4.png, neg\_28.png, pos\_24.png, pos\_41.png, sample1.png, sample2.png, partitura.png

Para além desses itens, abaixo encontra-se a estrutura de pasta como estão organizados o código e os arquivos anexos que são itens primordiais para o funcionamento do programa.

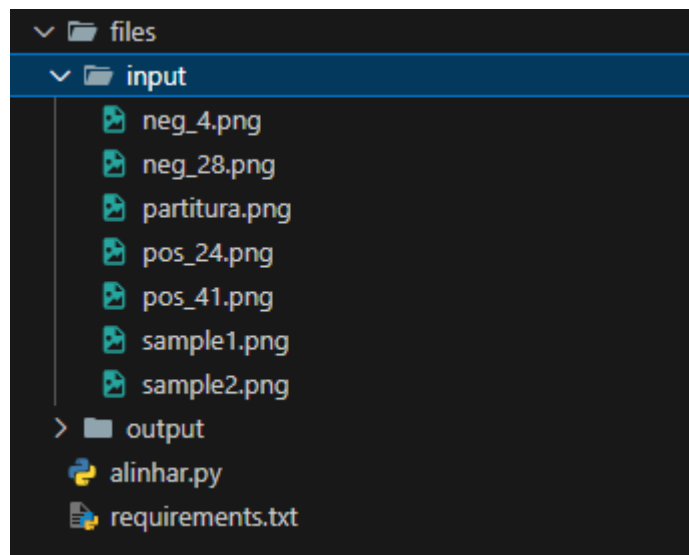


Figura 1 – Estrutura de pasta com os códigos fontes e anexos

Caso seja necessário a instalação de todas as dependências do programa, basta executar no terminal o seguinte comando:

- `pip install -r requirements.txt`

Isso será responsável por instalar todas as bibliotecas que são dependências do programa que realiza o alinhamento do conteúdo das imagens.

### 3. Modo de Uso

Com as dependências instaladas, agora deve-se executar o programa para então executar o alinhamento da imagem. A execução do código `alinhar.py` deve seguir o seguinte formato, abra o terminal para executar o comando:

- `python alinhar.py <imagem_entrada.png> <opção> <imagem_saida.png>`
  - **<imagem\_entrada.png>**: Caminho para imagem de entrada que será usada como base das operações
  - **<opção>**: As possíveis opções são “horizontal” ou “hl” para projeção horizontal, ou “hough” ou “hg” para a transformada de Hough
  - **<imagem\_saida.png>**: Caminho para a imagem de saída com o alinhamento do conteúdo

## 4. Resultados e Discussões

Neste momento, é preciso apresentar as implementações e ideias aplicadas nas duas abordagens escolhidas para o alinhamento de textos em imagens, que são, projeção horizontal e transformada de Hough. Posteriormente, basta analisar os respectivos resultados e, então, realizar a comparação dos dois métodos, identificando pontos positivos e negativos.

### 4.1. Projeção Horizontal

A ideia inicial deste método é rotacionar a imagem em torno de seu centro, realizando projeções horizontais dos pixels rotacionados e assim, identificar qual das projeções realizadas encaixa-se melhor com um texto alinhado.

O processo é dividido em etapas. A primeira delas é definir quais serão os ângulos de varredura. No caso em questão, o ângulo inicial é  $-180^\circ$  e o ângulo final é  $180^\circ$ , com passo de 1 grau.

Com isso definido, agora, para cada ângulo será associado um valor. Esse valor é obtido a partir da função objetivo, que irá quantificar quão boa é aquela projeção.

A função objetivo recebe como parâmetro justamente o ângulo de rotação da imagem. A partir desse valor, rotaciona-se a imagem. Posteriormente, faz-se necessário a projeção horizontal com a nova imagem rodada, essa projeção consiste em criar um vetor com  $r$  posições, em que  $r$  é o número de linhas da imagem. Dessa forma, matematicamente, pode-se definir a projeção horizontal como,

$$P[r] = \sum_{c=0}^{M-1} f(r, c), \forall r = 0, 1, 2, \dots, N-1$$

Considere essa uma imagem com tamanho  $N \times M$  e que o valor do pixel na posição  $x$  e  $y$  na imagem é dada por  $f(x, y)$ .

Ao final dessa operação, será obtido o vetor de projeção  $P$ . Para determinar sua qualidade será usada a seguinte função, determinada por  $D_\theta$ ,  $\theta$  é o ângulo de rotação.

$$D_\theta = \sum_{i=0}^{N-2} (P[i] - P[i+1])^2$$

Essa fórmula, no contexto de alinhamento de texto, é válida pois, quando a imagem chegar ao alinhamento correto, a soma das diferenças entre cada par de linhas vizinhas ao quadrado será máxima. Uma vez que, com as letras alinhadas existe uma concentração de pontos pretos em determinadas linhas da imagem, enquanto em outras linhas, devido ao formato das letras, essa concentração é menor. Já para o caso em que a imagem não está alinhada, não há essa concentração de pixels pretos em determinadas linhas, já que estão diluídos por todas as linhas. Com isso, a projeção tende a ser mais uniforme. A partir dessa constatação, é possível chegar que a diferença entre a soma das diferenças vizinhas diz muito com relação ao alinhamento da imagem.

Feito isso para todos os ângulos possíveis, agora só é preciso obter o ângulo de resposta  $\theta_r$  que possui o máximo  $D_\theta$  associado.

$$\theta_r = \max D_\theta$$

## 4.2. Transformada de Hough

O objetivo deste método consiste em encontrar os segmentos de retas presentes na imagem. Isto é, achar os pontos colineares. Para o contexto de alinhamento de texto, o problema é dado um texto desalinhado, deve-se identificar a angulação dele e, assim, realizar a correção necessária para alinhar o texto corretamente.

Antes de aplicar a transformada de Hough para encontrar as retas, é preciso previamente executar um operador de detecção de bordas, que irá otimizar a localização e a presença de pontos colineares na imagem. O método de detecção de bordas escolhido foi o Operador de Canny.

Inicialmente, suaviza-se a imagem de entrada com um filtro Gaussiano, já que esse é um filtro passa-baixa. Em seguida, de forma similar à descida do gradiente, calcula-se a magnitude e direção do gradiente com as diferenças finitas para as derivadas parciais. Dessa forma, os pontos de borda podem ser identificados facilmente, uma vez que tais pontos, localmente, têm a magnitude máxima na direção do gradiente.

Tal operador recebe como parâmetro um threshold, dito como sigma  $\sigma$  que tem por objetivo não incluir bordas falsas devido ao possível ruído contido na imagem. Para o contexto de alinhamento, o sigma definido foi  $\sigma = 2$ , haja visto que esse limiar devolve um resultado plenamente satisfatório quanto a eliminação do ruído do mapa de bordas. Como o objetivo é apenas identificar o ângulo de rotação do texto, mesmo que ocorra o corte de algumas bordas existentes, ainda assim é o suficiente para identificar o conteúdo e qual a angulação dele.

Ao final desse processo, obtém-se o mapa de bordas da imagem de entrada. Com isso, é possível agora aplicar o método da transformada de Hough para identificação das linhas na imagem. Tal operação consiste em discretizar um espaço de parâmetros  $E(\rho, \theta)$  em que  $\theta$  é o ângulo da reta na imagem.

Posteriormente a isso, para cada ponto  $(x, y)$ , calcula-se ambos os valores  $\rho$  e  $\theta$  e, assim, incrementa-se a posição  $(\rho, \theta)$  no espaço de parâmetros. Com todas as posições mensuradas, consegue-se a matriz do espaço de parâmetros  $E(\rho, \theta)$  que possui a quantidade de ocorrências das retas que correspondem a  $\rho$  e  $\theta$ .

Podem existir várias retas dentro da imagem, dessa forma, é preciso agora detectar quais são os picos relevantes, que serão as dadas retas na imagem. De acordo com a interface do método, é possível definir um limiar de intensidade dos parâmetros da reta. Ou seja, fazer um corte com relação ao número de ocorrências de uma dada posição na matriz  $E(\rho, \theta)$ .

A lógica para definir um threshold eficiente quanto ao corte realizado é dizer que os valores de ocorrências dos parâmetros deve ser pelo menos 30% do número de colunas da imagem, isto é,  $0.3M$  em que  $M$  é o número de colunas. Isso é suficiente para adaptar-se ao tamanho da imagem de entrada, bem como, selecionar primordialmente as retas que correspondem ao texto. Para além disso, se o limiar fosse mais alto, muitas vezes não é capaz de identificar nenhuma reta e assim gerar um erro.

Feito isso, o método devolve a lista de ângulo das retas que representam os picos, ou seja, as retas presentes na imagem. Com essa lista, seleciona-se a mediana desse vetor de ângulos como resposta do ângulo de rotação da imagem. Tal opção estatística é escolhida, pois evita a escolha de um ângulo que corresponde a um possível outlier, ou mesmo, alinhar de acordo com algum outro elemento da imagem que não é o texto.

Além dessas escolhas, é preciso fazer uma pequena correção em relação ao ângulo de resposta  $\theta_r$ , que é transladar o ângulo em  $-90^\circ$ , já que a resposta pelo método de Hough vem com o ângulo transladado em  $90^\circ$ .

$$\theta_r = \text{mediana}(\text{angulos}) - 90$$

#### 4.3. Ajuste do Ângulo e Rotação com o Ângulo de Resposta

Devido à definição de como achar o alinhamento correto da imagem, é possível encontrar duas respostas válidas como resultado, uma delas é a imagem alinhada corretamente e outra é a imagem alinhada de cabeça para baixo. Ambos os casos podem ser vistos abaixo, e claramente cada uma das duas está com seu ângulo de rotação correto, porém com orientações diferentes.

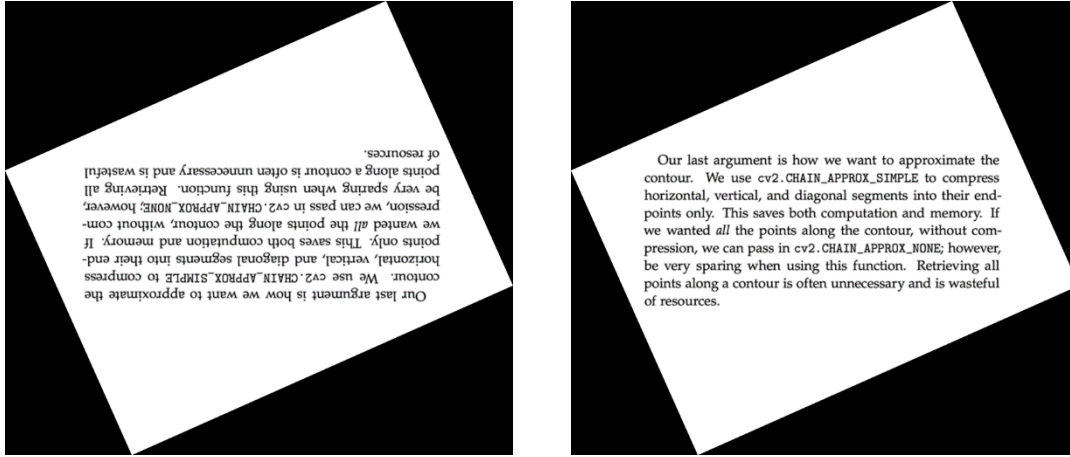


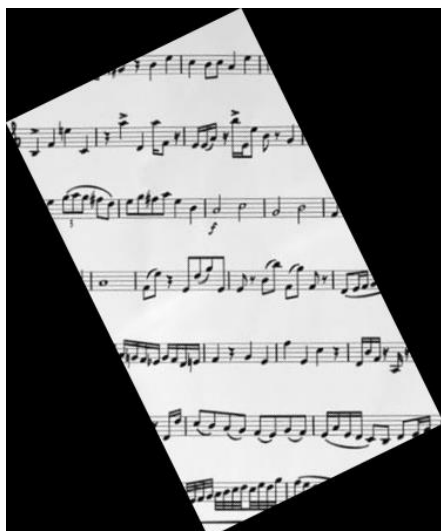
Figura 2 – Imagem rodada de cabeça para baixo e imagem correta

Os possíveis ângulos de rotação estão no intervalo fechado  $[-180^\circ, 180^\circ]$ . Para evitar que ela fique de cabeça para baixo com relação à imagem de entrada, é possível verificar que manter o ângulo de rotação entre  $[-90^\circ, 90^\circ]$  garante com que o conteúdo da imagem não rode completamente, isto é, não fique de ponta cabeça. Porém, antes de tudo é preciso garantir que o resultado de ambos os métodos anteriormente explicado não seja perdido. Para isso, é preciso computar o ângulo de resposta novo que mantém a rotação de resposta obtida.

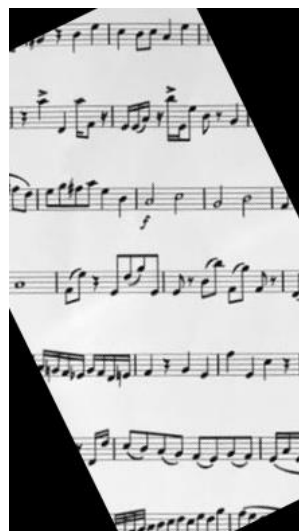
$$\theta_{r_{novo}} = \begin{cases} \theta_r & \text{se } \theta_r \in [-90^\circ, 90^\circ] \\ \theta_r + 180^\circ & \text{se } \theta_r < -90^\circ \\ \theta_r - 180^\circ & \text{se } \theta_r > 90^\circ \end{cases}$$

Isso pode ser entendido da seguinte maneira, caso o ângulo de resposta original dos métodos  $\theta_r$  extrapole o intervalo  $[-90^\circ, 90^\circ]$  rotacione translade esse ângulo em  $180^\circ$ , no sentido horário caso o ângulo for negativo e no sentido anti-horário caso o ângulo seja positivo. Para imagem o que ocorre é justamente uma rotação de  $180^\circ$  a partir do ângulo de resposta original, no sentido horário ou anti-horário a depender do sinal do ângulo.

Quanto a rotação em si, poderiam ser feitas duas escolhas: rotação com perda de informação e sem perda de informação. Ambos os casos podem ser vistos na figura a seguir,



(a) Sem perda



(b) Com perda

Figura 3 – Rotação com e sem perda

Para o uso e exibição das informações, foi escolhido o resultado de rotação sem perda e com resize. Por mais que o modelo de rotação com perda tenha vantagens, como manter o tamanho original da imagem e, portanto, mantém a proporção original dos objetos, ainda assim, o fato de esconder caracteres ou elementos do texto é possível que seja prejudicial, haja visto que, nessas áreas escondidas, pode haver informações primordiais para o entendimento da mensagem.

Para além disso, as áreas que não existiam antes da rotação são assignadas com preto (tom de cinza 0), isso foi feito para não inserir novas informações na imagem, bem como para acentuar qual foi o ângulo de correção tomado. Para o método de projeção horizontal, essa escolha não impacta no resultado.

#### 4.4. Análise dos resultados da Projeção Horizontal

Levando todos os passos em conta, agora basta executar o algoritmo e verificar quais serão os resultados e quais as considerações a serem feitas. O algoritmo foi testado com três classes de imagens, uma imagem mais simples, um texto grande como um artigo e uma partitura, uma estrutura ainda mais complexa.

Para todos os casos, o resultado foi plenamente satisfatório, sem a necessidade, inclusive, da binarização da imagem. Aplicando o algoritmo de projeção horizontal nas imagens originais foi o suficiente para obter os resultados desejados.

Para imagens mais simples, em que o texto é o elemento central e não existem outros elementos na imagem, que possivelmente interferem na resposta, o algoritmo comporta-se muito bem. Para o caso de uma imagem maior e com mais elementos, o único ponto a ser ressaltado é que, o algoritmo leva um tempo grande para computar. Por exemplo, para a imagem sample2.png o ângulo de resposta é obtido após 18 segundos, varrendo de  $-180^\circ$  a  $180^\circ$  com passo de  $1^\circ$  grau. Essa é uma imagem de dimensões grandes e isso é determinante para que o algoritmo leve mais tempo para computar. Uma imagem de dimensões menores, leva 7 segundos para devolver o valor de resposta. Isso mostra que, devido ao funcionamento do algoritmo, as dimensões da imagem são determinantes quanto a sua complexidade e tempo.

Por fim, para o caso da partitura, tem-se uma imagem com tons de cinza diversos no conteúdo da imagem. Ao aplicar o método de projeção, sem realizar nenhuma binarização, ele encontra o ângulo de rotação e alinha a imagem corretamente. O que mostra que o método tem um funcionamento plenamente satisfatório mesmo para imagens com uma gama maior de intensidades.

Abaixo estão dispostas as três imagens de teste utilizadas com as respectivas imagens de saídas com os ângulos de resposta.

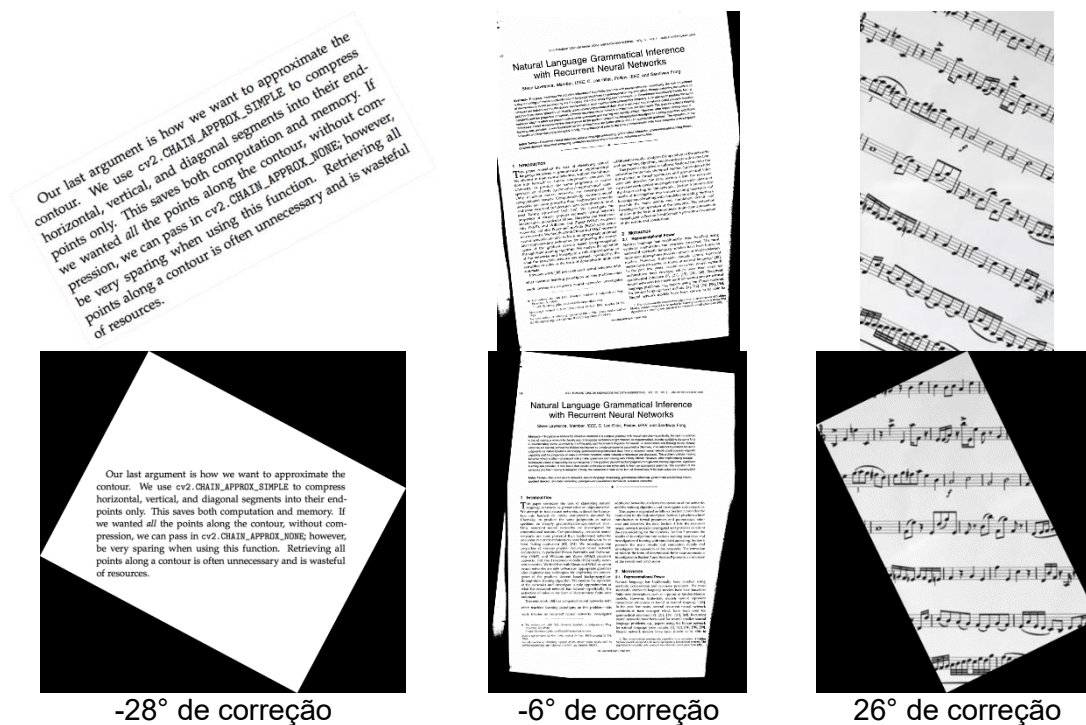


Figura 4 – Resultado da aplicação do método projeção horizontal

#### 4.5. Análise dos resultados da Transformada de Hough

Neste momento, aplica-se o método de transformada de Hough anteriormente descrito anteriormente. De mesmo modo, como entrada foram usadas imagens mais simples, mais complexas e com dimensões maiores e com símbolos e elementos diferentes.

Em relação à projeção horizontal, esse método leva menos tempo e devolve uma resposta plenamente satisfatória. Isso ocorre uma vez que se realiza menos computações na matriz, sem a necessidade de passar por todos os ângulos. Em ambos os métodos, as respostas foram iguais para todas as instâncias de entrada.

O resultado de ângulo de saída é sensível ao threshold que seleciona os picos da transformada de Hough. Caso o threshold for muito alto, corre o risco de não selecionar nenhum pico na transformada, da mesma forma, se ser muito baixo, ele seleciona um ângulo que não faz sentido com relação ao alinhamento da imagem. Dito isso, foram realizados alguns testes para assim chegar ao threshold selecionado, que no caso é  $0,3M$  em que  $M$  é o número de colunas.

Um outro desafio encontrado, que também tem interferência pelo limiar selecionado, é o fato do ângulo de alinhamento de imagens mais complexas, que



possuem mais linhas em seu conteúdo, podem ter influência de outros elementos, como por exemplo a partitura. Para textos simples, isso não ocorre.

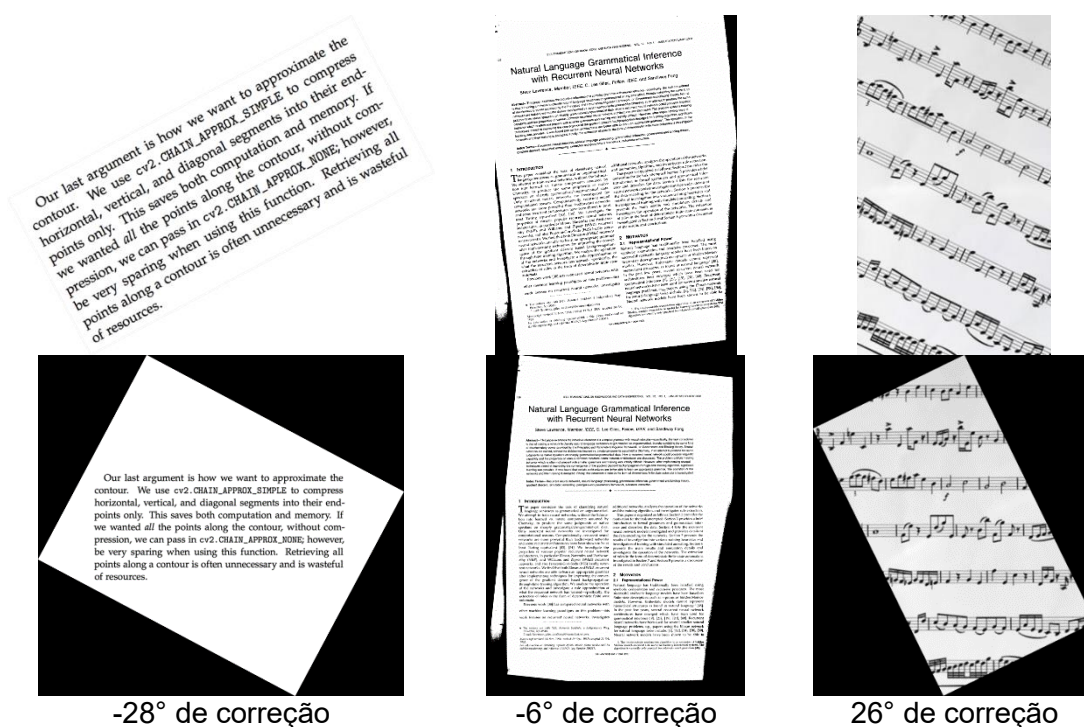


Figura 5 – Resultado da aplicação da transformada de Hough

#### 4.6. Desafios e Dificuldades

Uma das grandes dificuldades durante o processo de construção dos algoritmos foi definir os thresholds utilizados no funcionamento dos métodos. Em alguns casos funcionavam, mas em outros casos de testes, falhavam ou não faziam o alinhamento correto.

Outro ponto, foi qual seria a função objetivo a ser usado para definir a qualidade da rotação para um determinado. Inicialmente, a variância do vetor havia sido utilizada, mas os resultados não eram bons. E isso foi corrigido posteriormente utilizando a soma das diferenças quadradas entre os vizinhos.

## 5. Conclusão

Dessa forma, com o que foi apresentado, permite-se chegar à conclusão de que o objetivo foi atingido da maneira desejada, isto é, dada uma imagem de entrada com seu conteúdo inclinado, o programa devolve uma imagem de saída com o seu conteúdo alinhado corretamente com ambos os métodos.

Para o método de projeção horizontal, pode-se identificar que seu funcionamento é eficaz e não faz uso de cortes arbitrários os ditos limiares. Uma vez que a varredura pelo melhor ângulo de alinhamento é feita ângulo a ângulo. Esse fato faz com que seu funcionamento não seja eficiente com relação tempo. Ele toma um pouco mais de tempo para executar todas as computações necessárias. O que determina quanto tempo leva a mais ou a menos para devolver a resposta é a dimensão da imagem  $N \times M$ .

Enquanto para o método de transformada de Hough, é um método mais eficiente com relação ao tempo, haja visto que sua implementação é mais esperta, no sentido de usar o conceito de retas presente na imagem e identificar seu ângulo e assim realizar a correção. O ponto primordial dentro de Hough é definir o melhor threshold para diferentes casos, a depender do número utilizado, o método pode selecionar ângulos indesejados.

Em suma, ambos os métodos, com as devidas parametrizações, tiveram resultados plenamente satisfatórios. Para todos os casos de entrada, foi possível obter o alinhamento correto para os dois algoritmos.