Introdução ao Processamento Digital de Imagem

Trabalho 3

Pedro Terra Delboni
RA 082494

I. Introdução

Esse documento tem como objetivo explicar o funcionamento e resultado do programa desenvolvido para o *Trabalho 3* da matéria *MO443*. O trabalho tem como objetivo, dado a imagem de um texto, reconhecer a inclinação da imagem retornar uma nova imagem corrigindo essa inclinação (o texto resultante idealmente deveria estar em uma horizontal perfeita).

A seção II inclui exemplos de execução do programa, os possíveis parâmetros de entrada e o que eles fazem e descrição de quais as possíveis saídas geradas por ele.

II. Execução

Dentro do *zip* referente ao trabalho encontra-se o arquivo *lab3.py*. Esse arquivo é um código em *python3* desenvolvido para realizar as tarefas especificadas na descrição do trabalho. Nessa seção serão apresentados:

- Dependências necessárias para a execução.
- Funcionalidades implementadas e seu mapeamento para as taréfas exigidas.
- Possíveis parâmetros aceitos pelo programa e o que eles fazem.
- Exemplos de execução.

A. Dependências

1) matplotlib: Usada para gerar os histogramas das imagens.

2) numpy: Usada para manipulação de vetores.

3) sys: Usada para aquisição dos argumentos passados na linha de comando.

4) scipy.stats: Usada para aquisição de entropia de vetores.

5) scipy.ndimage: Usada rotacionar matrizes 2D (imagens).

6) skimage.transform: Usada para realizar a transformada de Hough para linhas nas imagens.

7) PIL: Usada para abrir e salvar imagens.

B. Funcionalidades

Essa seção tem como finalidade explicar brevemente as funcionalidades implementadas. Uma descrição mais profunda contendo detalhes da implementação pode ser encontrada em III e análise de resultados pode ser encontrado na seção IV

1) Alinhar documento baseado em projeção horizontal: Como especificado pelo item 1.2 do trabalho, utiliza-se de projeção horizontal para descobrir a inclinação do texto e alinhá-lo para o angulo correto.

2) Alinhar documento baseado em Transformada de Hough: Como especificado pelo item 1.3 do trabalho, utiliza-se da transformada de Hough para descobrir a inclinação do texto e alinhá-lo para o angulo correto.

3) Statísticas: Uma funcionalidade extra implementada foi a extração de estatísticas relacionadas as diferentes formas de descobrir a inclinação do documento. Esse modo de execução foi utilizado para avaliar as diferentes técnicas durante o desenvolvimento, o que permitiu iterar sobre elas e encontrar formas mais robustas para realizar a tarefa.

C. Parâmetros

O programa pode ser executado do seguinte modo:

python3 lab3.py INPUT [OUTPUT] [MODE] [METRIC] [ANGLE1] [ANGLE2] [VERBOSE] [STATS]

Letras maiúsculas simbolizam os parâmetros que serão descritos a seguir. Parâmetros entre chaves são opcionais.

1) INPUT: Único parâmetro obrigatório, informa a imagem de documento de deve ser alinhada. Ex: *neg_4.png*

2) OUTPUT: Parâmetro opcional usado para informar o nome que a imagem alinhada do documento deve ter na saída. Se o parâmetro não for informado, a imagem terá o nome *output.png*. O parâmetro deve ser especificado da seguinte forma:

-o NAME

Onde NAME é o nome desejado do arquivo de saída.

3) MODE: Esse parâmetro determina a forma com que o programa vai encontrar a inclinação do texto. As seguinte opções estão disponíveis:

- **-hough_m**: Transformada de Hough escolhendo a moda dos ângulos dos picos.
- **-hough_p**: Transformada de Hough escolhendo o ângulo do maior pico.
- **-hp**: (Horizontal Projection) Projeção horizontal, iterando por todos os ângulos e encontrando o melhor entre eles.
- **-fhp**: (Fast Horizontal Projection) Técnica baseada na entropia da projeção horizontal. Tenta minimizar o número de ângulos testados para encontrar a inclinação do documento

As diferentes técnicas podem retornar resultados diferentes. Desempenho de cada uma será discutido em IV. Quando nenhum modo é informado o programa usa o modo equivalente a **-hough_m**. No caso onde mais de um parâmetro é informado para modo, apenas o último será usado.

4) METRIC: Esse parâmetro só é utilizado quando o programa é executado no modo **-hb**. Ele é usado para determinar qual a métrica que sera usada para avaliar a projeção da imagem. As diferentes métricas disponíveis são:

- **-wlc** (White Lines Count) Retorna a quantidade de linhas brancas horizontais encontradas na projeção horizontal.
- **-wgc** (White Gap Count) Retorna a quantidade de espaços brancos contínuos horizontais encontrados na projeção horizontal.
- **-dp** (Darkest Peak) Métrica proposta pela descrição do trabalho, retorna a linha de maior intensidade encontrada na projeção horizontal.
- **-dm** (Dark Mean) Retorna a média das intensidades da projeção horizontal.
- (-ent) (Entropy) Retorna a entropia da projeção horizontal.
- **-ene** (Energy) Retorna a soma do quadrado dos valores encontrados na projeção horizontal.

Se nenhum dos seguintes parâmetros for informado, o programa usa como padrão o equivalente a **-ent**. No caso onde mais de um parâmetro é informado para métrica, apenas o último será usado.

5) ANGLE1 e ANGLE2: Esses dois parâmetros são usados para especificar quais os possíveis ângulos para a inclinação do texto na imagem original. Podem ser especificados da seguinte forma:

6) VERBOSE: Informado pela flag **-v**, ele faz com que o programa imprima informações extras em relação a execução.

-ang1 X
-ang2 Y

Onde X representa o menor ângulo possível para a inclinação, e Y o maior. O programa assume como padrão os valores de -45 para X e 45 para Y. X deve ser sempre menor que Y.

7) STATS: Informado pela flag **-stats**, habilita a saída de informações estatísticas referente as diferentes técnicas de correção de inclinação usada. Mais informações sobre essas saídas podem ser encontradas na seção III-C

D. Exemplos de execução

Moda dos picos de Hough entre -45 e 45:
python3 lab3.py input.png -o output.png

Maior pico de Hough entre -60 e 60:
python3 lab3.py input.png -o output.png -hough_p -ang1 -60 -ang2 60

Pico da projeção horizontal entre -45 e 45:
python3 lab3.py input.png -o output.png -hp -dp

Busca reduzida de entropia em projeção horizontal entre -45 e 45 e geração de estatísticas:
python3 lab3.py input.png -o output.png -fhp -stats

Energia da projeção horizontal entre -50 e 45 verboso:
python3 lab3.py input.png -o output.png -hp -ene -v

III. Implementação

Nessa seção serão descritos os paços e motivações que levaram para o estado final da implementação do programa e uma breve descrição de seu funcionamento.

Universidade Estadual de Campinas
Instituto de Computação
Introdução ao Processamento Digital de Imagem (MC920 / MO443)
Professor: Hélio Pedrini

Trabalho 3

1 Especificação do Problema

O objetivo deste trabalho é implementar algoritmos para alinhamento automático de imagens de documentos.

1.1 Análise de Documentos

A análise de documentos pode ser empregada na conversão automática da informação contida nas imagens de documentos em texto editável. A digitalização de documentos tem sido amplamente utilizada na preservação e disseminação de informação em formato eletrônico.

Um problema frequente que ocorre no processo de digitalização é o desalinhamento do documento, ou seja, o posicionamento do papel com uma inclinação diferente do eixo do digitalizador. A correção da inclinação é fundamental para o adequado funcionamento de sistemas de reconhecimento ótico de caracteres.

Dois algoritmos para detecção e correção de inclinação de documentos devem ser implementados neste trabalho, um baseado em projeção horizontal e outro baseado na transformada de Hough.

Figura 1. Imagem usada como referência de texto alinhado

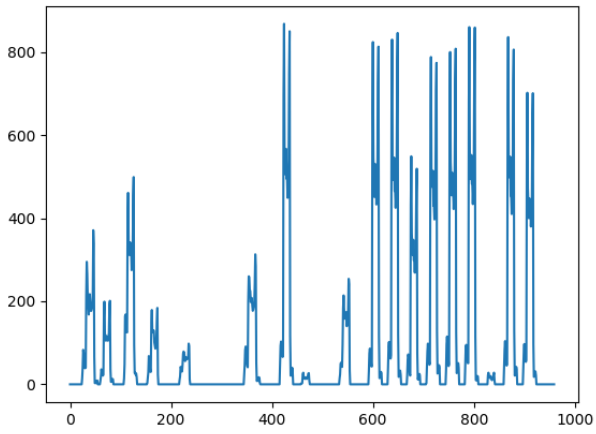

Figura 2. Projeção horizontal da figura 1

A. Projeção Horizontal

Para uma compreensão inicial do problema pegamos uma imagem da descrição do trabalho que já está alinhada e não possui ruídos. A imagem pode ser vista na figura 1. Passamos a imagem por um processo de binarização simples (pixeis com intensidade acima da média são considerados brancos, e abaixo pretos) e fizemos a projeção horizontal que pode ser vista na figura 2.

A projeção mostra a quantidade de pixeis pretos no eixo Y pelas linhas no eixo X. Conseguimos com uma certa facilidade identificar picos em linhas que deveriam conter textos, e vales nos espaços brancos entre elas. Esses detalhes não estão presentes na figura 3 que contem a projeção da imagem rotacionada em 10 graus, ao invés disso percebemos uma distribuição mais aleatória de intensidades. A partir de agora a projeção será referida como uma imagem unidimensional, onde cada pixel representa uma linha da imagem original e a intensidade do pixel representa o resultado da projeção daquela linha.

Antes de testar fazer uma implementação do algoritmo

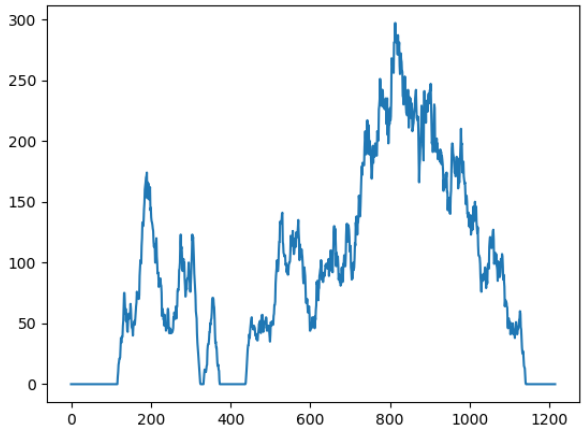

Figura 3. Projeção horizontal da figura 1 rotacionada 10 graus

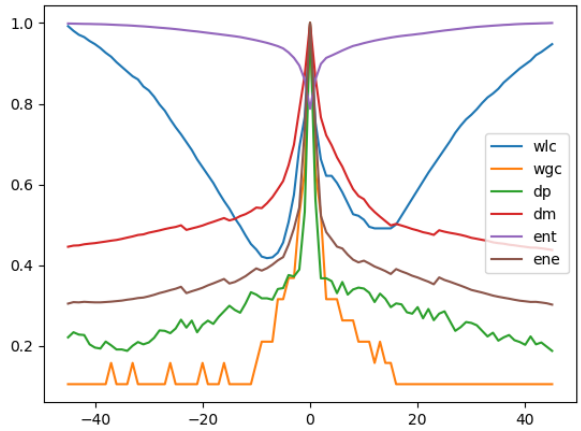

Figura 4. Extração de métricas para cada rotação da figura 1

proposto na especificação do trabalho, tentamos propor novas métricas para qualificar uma projeção horizontal baseado na análise das duas projeções mostradas.

Foram propostas as seguinte outras medidas:

- Maior pico da projeção (**dp**): Métrica proposta no trabalho. Parece ser verdade, pois como podemos ver na projeção, os picos são proporcionais ao tamanho das linhas de texto, logo o texto alinhado maximizaria a quantidade de pontos nas linhas.
- Quantidade de linhas brancas (**wlc**): Um texto desalinhado deveria minimizar a quantidade de linhas brancas, pois os espaços entre as linhas de texto iriam sumir na projeção.
- Quantidade de espaços contínuos brancos (**wgc**): A ideia é a mesma do item anterior, mas dessa vez o tamanho dos espaços em branco não influencia na medida, apenas a sua existência.
- Média da região preenchida da projeção (**dm**): Divide a soma das intensidades da projeção (que representa a quantidade total de pixeis) e divide pelo número de linhas que possui alguma intensidade. O número de linhas brancas deveria subir com o alinhamento, logo o número de linhas preenchidas deveria diminuir, aumentando então a média.
- Entropia (**ent**): A entropia de uma imagem aumenta a medida que os seus pixeis tem uma distribuição de intensidade mais aleatória. No caso da projeção, temos uma imagem unidimensional que apresenta um comportamento aparentemente bem definido, com concentrações de pixels de valor 0 (linhas brancas) e de pixeis com valor alto (linhas de texto).
- Energia (**ene**): Definimos como energia a soma dos quadrados dos pixeis da projeção. Sabemos que a soma da intensidade dos pixeis é sempre a mesma, mas quanto mais concentrado é essa intensidade, maior fica a soma dos seus quadrados. Logo essa medida deveria ser máxima quando as intensidades da projeção estão concentradas em menos pontos.

Uma vez proposta as medidas pegamos a figura 1 e rotacionamos entre cada ângulo inteiro entre -45 até 45 e calculamos as métricas para as projeções de cada uma das rotações. O resultado pode ser visto na figura 4, onde o eixo X são os ângulos entre -45 e 45 e o eixo Y é a métrica escalada para caber entre valores de 0 e 1.

Essa imagem nos ajuda a concluir que para um texto bem comportado como o da figura 1, todas as métricas possuem um máximo ou mínimo no ângulo desejado, mas podemos perceber que a métrica de quantidade de linhas brancas não funciona tão bem quanto esperávamos. Ela aumenta a medida que rotacionamos a imagem. Isso acontece por causa das bordas da imagem. A quantidade de linhas brancas na borda é a menor distância entre a borda superior e o texto. Ao rotacionar, a borda superior passa a ser definida pelos cantos das bordas, e a distância entre eles e o texto é maior do que a dos outros pontos da borda, o que gera um aumento na quantidade de linhas brancas. Esse aumento entretanto não é visto na métrica **wgc** pois ela ignora aumentos de espaços brancos pois conta apenas a quantidade deles.

Outra conclusão importante que podemos chegar a partir da figura 4 é que, por mais que as métricas sejam máximas (ou mínimas) no ângulo correto, comparar a métrica de dois ângulos não nos diz qual deles é o melhor. Podemos ver que o gráfico de quase todas as métricas apresentam saliências durante a curva. A métrica de duas rotações não nos diz qual delas é melhor, a única conclusão válida é quando testamos para todos os ângulos e escolhemos entre todos eles o de maior métrica. O único caso onde isso não foi verdade foi para a entropia. A entropia obtida para as rotações da imagem cresceu assintoticamente com a distância entre o ângulo de rotação e o ângulo correto. Futuramente veremos que isso não é verdade para todos os casos, mas para os casos onde isso se aplica, é possível encontrar a entropia mínima sem explorar todos os

ângulos exaustivamente.

1) Implementação: Implementamos duas versões da solução baseada em projeção horizontal. A primeira recebe como parâmetro a métrica que nós desejamos usar e a faixa de ângulos que queremos trabalhar. Essa solução funciona como a descrita na especificação do trabalho, iterando sobre todas os ângulos inteiros da faixa explicitada e retornando o ângulo que maximiza (ou minimiza no caso da entropia) a métrica desejada. Os resultados da implementação podem ser vistos na seção IV.

Uma segunda implementação, chamada de *Fast Horizontal Projection* foi feita para explorar a propriedade de crescimento assintótico da entropia. Ela funciona da seguinte maneira: Dado uma faixa de ângulos, dividimos ela em 4 pontos: Os dois extremos, e mais dois pontos no meio de forma a dividir igualmente em 4 partes a faixa de ângulos. Calculamos a entropia para a projeção dos 4 pontos escolhidos. Procuramos qual dos pontos possui a entropia menor que o ponto seguinte. Sabemos então que a entropia mínima deve estar próxima desse ponto. Tomamos como a nova faixa de ângulo os pontos que contornam ele (isso diminui a amostra em 1 terço caso o ponto seja um dos dois do meio, e 2 terços caso ele seja um dos dois do extremo). Repetimos agora o processo recursivamente para a nova faixa, até que a faixa tenha largura de 1 grau, onde retornamos o centro da faixa como resposta. Dessa forma, a quantidade de rotações que fazemos deixa de crescer linearmente com a quantidade de ângulos da faixa, e passa a crescer de forma logarítmica, o que diminui o número de execuções. Na seção IV vamos mostrar que ela é mais eficiente, mas infelizmente a propriedade em questão não é verdade para todas as imagens. A figura 5 mostra a extração de métricas para a imagem *sample2.png* fornecida na descrição do trabalho. Essa foi a imagem com maior número de pixeis escuros que não faziam parte do texto. Como podemos ver na figura, boa parte das métricas não funcionaram de forma adequada para ela. A entropia continuou sendo mínima no ângulo certo, mostrando-se como uma métrica robusta, mas como podemos ver a sua curva, ela deixou de crescer assintoticamente a medida que se distância do bom ângulo, pois depois de um certo ponto ela volta a cair. Nessa imagem por exemplo não foi possível obter um bom resultado com esse método rápido, mas o resultado continuou bom com o método original.

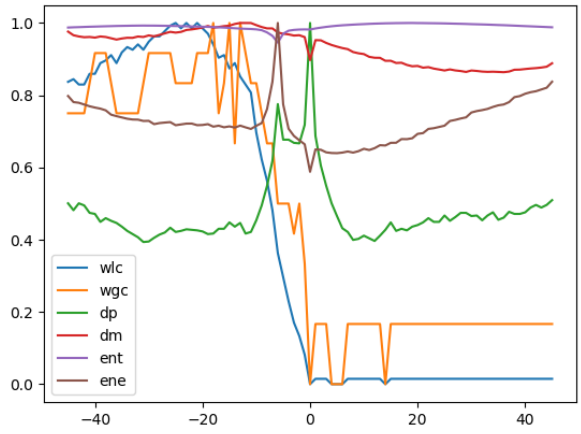

Figura 5. Extração de métricas para cada rotação da imagem *sample2.png* fornecida na descrição do trabalho

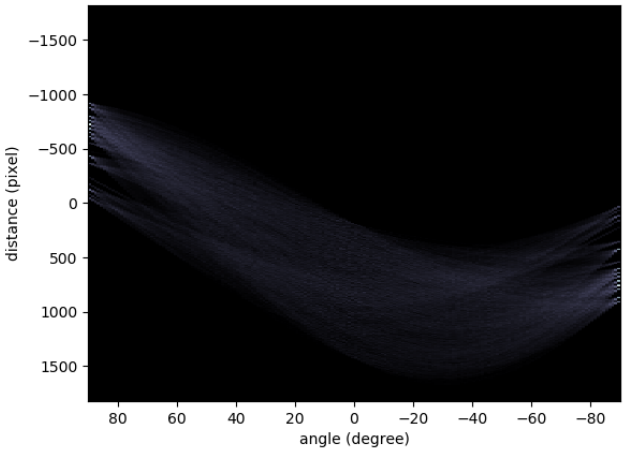

Figura 6. Resultado da transformada de Hough da figura 1

B. Transformada Linear de Hough

A transformada de hough pega todos os pixels diferentes de 0 de uma imagem e transforma eles em um seno no espaço de Hough. Nesse espaço, o eixo X representa um ângulo, e o eixo Y uma distância. Cada pixel do espaço de Hough vai ter sua intensidade determinada pela quantidade de senos que passam por ele. Não vamos entrar muito em detalhe sobre como funciona a transformada de Hough pois acreditamos que isso foge do escopo do relatório, o importante aqui é saber que a intensidade de um pixel na posição (A, B) do espaço de Hough representa quantos pixeis diferentes de 0 existem em uma reta na imagem original que possui ângulo A, e distância B da origem.

Para fazer a transformada é recomendado fazer uma extração de bordas da imagem original. Isso não foi feito nesse trabalho. No caso de uma imagem normal (foto), esse paço é muito importante, pois normalmente deseja-se encontrar as retas que passam por essas bordas. O caso de texto é diferente, deseja-se encontrar as retas que passam por dentro do máximo possível de letras. Logo para o nosso resultado não deveria fazer diferença usar as letras inteiras ou apenas as suas bordas, a maior diferença seria em relação a desempenho, mas a execução do programa já foi rápida o suficiente para se justificar fazer um passo inicial de extração de bordas.

A figura 6 mostra a transformada do espaço de Hough da figura 1, e a figura 7 mostra a transformada do espaço de Hough da mesma figura, mas rotacionadas 10 graus em sentido anti-horário. Analisando essas figuras podemos destinguir um

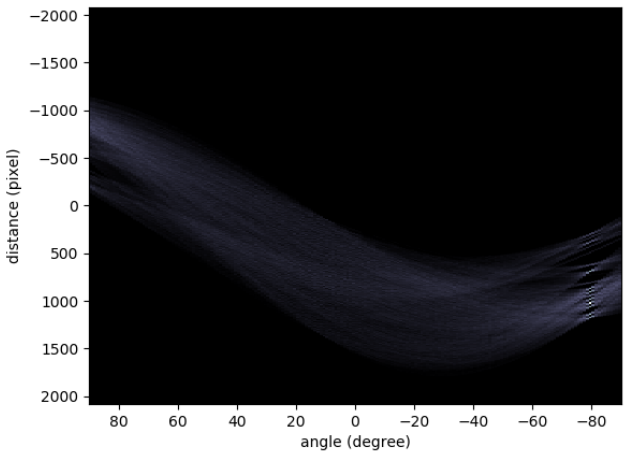

Figura 7. Resultado da transformada de Hough da figura 1 rotacionada 10 graus no sentido anti-horário

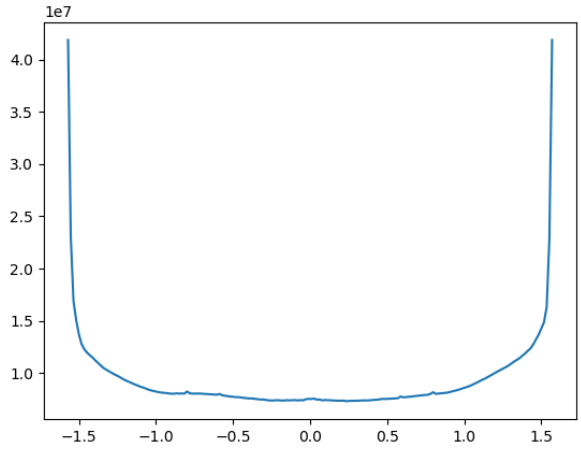

Figura 8. Projeção vertical dos quadrados das intensidades da figura 6

padrão de picos em uma das retas verticais da imagem. Esse padrão significa que existem diversas retas com o ângulo correspondente ao ângulo da sua posição no eixo X. Como as linhas de um texto são todas paralelas, isso é um forte indício de que cada uma dessas linhas representa uma linha de texto. Em uma das imagens os picos estão no ângulo de 90 graus, e na outra no de 80. Isso também é o que esperávamos, já que em imagem de textos alinhadas o texto deve estar perpendicular ao eixo X da imagem, e como sabemos que uma das imagens está desalinhada a -10 graus, o seu texto também deveria estar desalinhado em -10 graus.

Outra conclusão interessante é que ambas as figuras são iguais, apenas deslocadas em relação uma a outra. Isso mostra que a mesma imagem quando rotacionada gera um espaço de Hough muito semelhante, então ao contrário da solução dada por projeção horizontal, não é necessário fazer várias transformadas de Hough para diferentes rotações da imagem. A partir de uma única transformada podemos extrair o ângulo.

Para nos ajudar a visualizar os picos da transformada de Hough fizemos uma projeção vertical do quadrado das intensidades da imagem (sobre o eixo dos ângulos). O motivo para usar os quadrados é que uma simples projeção vertical retornaria uma reta, pois cada pixel faz parte de exatamente 1 uma reta de cada ângulo. O quadrado das intensidades permite que pixeis do espaço de Hough de maior intensidade se sobressaiam na projeção. O resultado pode ser visto na figura 8, onde o eixo X mostra os ângulos em radianos. Podemos ver claramente os picos nos ângulos de 90 e -90.

1) Implementação: Baseado nas informações até então extraídas, decidimos fazer duas variações para a implementação baseada na transformada de Hough oferecida pela biblioteca *skimage*.

A primeira é simples, uma vez feita a transformada de Hough, extraímos picos da mesma e calculamos o nosso ângulo a partir do maior pico encontrado. Essa implementação é ativada pela flag **-hough_p**

A segunda um pouco mais complexa, extrai os picos da transformada de Hough e busca a moda do eixo X desses picos. Ou seja, busca a posição do eixo X que contem a maior concentração de picos. Retornamos então o ângulo correspondente a esse eixo. Isso foi feito pois, como podemos ver na figura 6, nós temos um pico no espaço de Hough para cada linha de texto, e essas linhas deveriam todas ter o mesmo ângulo. Dessa forma, ao invés de escolhermos o ângulo com o maior pico, que poderia ser causado por algum artefato na imagem, nos atemos para o ângulo com maior números de picos. Essa implementação pode ser ativada pela flag **-hough_m**. Ela também é o modo de execução padrão caso nenhuma flag seja apresentada.

Um detalhe de ambas as implementações é que uma vez feita a transformada de Hough, os pixeis que correspondem a ângulos que estão fora da faixa especificada pelo usuário são zerados. Dessa forma o programa ignora uma possível existência de conjuntos de picos que possam atrapalhar o resultado.

O resultado de ambas as implementações e seus desempenhos podem ser encontrados na seção IV

C. Estatísticas dos diferentes modos

Ao executar o programa com a flag **-stats**, o programa além de corrigir o texto do modo especificado, ele imprime os resultados do ângulo que seria escolhido por todos os outros modos na saída padrão e as seguintes imagens:

- binary_image.png - Resultado da transformação da imagem original em binária que será usada pelo programa para fazer tanto a projeção horizontal quanto a transformada de Hough.
- horizontal_projection.png - Resultado da projeção horizontal da imagem original (antes de corrigir o ângulo)
- hough_line_transform.png - Imagem resultante da transformação para o espaço de Hough.

Children in netting "expressing themselves towards a communal end"

All caught up in art

COMMUNITY art is the name of the game at Edinburgh's Fruit Market Gallery Exhibition titled "Organised Accident is Art."

Art or not, one young schoolboy summed up the consensus view of school parties invited to attend when he declared: "This place is daft."

The exhibition is the Scottish Arts Council's latest sponsored venture in the capital.

With sponsorship to the tune of nearly £3000 — the Arts Council contribution is £1800 — the organisers have a lot of material and a host of community artists to make use of.

When I visited the gallery it was alive with activity as children scrambled over a net intertwined with hemp hanging from the ceiling and painted their names on bricks, handicapped people painted on paper, and some office girls drew plans of their homes on a wall.

Video tape was also in use and a camera team roamed the floor filming the day's events for an evening discussion session.

Artist Ken Wolverton explained: "A major function of the event reflects a major function of community arts — to enable all participants, whether artists or not, to express themselves, towards a communal end. The event will bring together individuals, images, ideas and skills."

The organisers, composed mainly of Craigmillar Festival Society and Theatre Workshop, Edinburgh, are uncertain what "works" will be left over and how they will be disposed of.

A bank clerkess, Fiona Johnstone, from Dalkeith, said: "I know art is more than a painting or some music, but it must be more than this, just making a mess. There must be something to show at the end."

Ironically, the most lasting impression the event is likely to leave on the city is a futuristic mural painstakingly painted on to a wall of a crumbling building opposite the gallery.

Derek Bateman

Figura 9. Imagem usada para testes do programa.

- hough_squared_projection.png - Projeção vertical do quadrado das intensidades da transformada de Hough.
- rotation_stats.png - Gráfico mostrando o valor de cada uma das métricas usadas para avaliar as projeções horizontais para cada um dos ângulos.

IV. Resultados

A. Precisão

As imagens usadas para testar o programa foram as 6 imagens propostas pela especificação do trabalho (*neg_4.png, neg_28.png, pos_24.png, pos_41.png, sample1.png e sample2.png*), a figura 1 e a figura 9 (fornecidas no .zip com o nome good_straight.png e scanned.jpg respectivamente). Os resultados podem ser vistos na tabela I

Pelos resultados da tabela I podemos ver que das soluções baseadas em projeção horizontal, os melhores resultados foram obtidos usando a entropia como métrica. Dessa forma foi possível se obter o resultado correto para todos os casos. Imagens que incluíam ruído ou artefatos que diferem do texto geraram um comportamento ruim para as implementações, como pode ser visto nos resultados das imagens *sample2.png e scanned.jpg*. Esses casos poderiam ser facilitados com um preprocessamento da imagem para tentar remover os artefatos. Um exemplo seria aplicar técnicas de erosão seguida por dilatação para ajudar a remover os ruídos da figura 9 pois eles parecem ser mais finos que as letras. Uma técnica de erosão seguida de dilatação também poderia ser utilizada para identificar os artefatos da imagem *sample2.png*, pois eles são claramente mais grossos que as letras, logo eles poderiam ser encontrados e removidos da imagem original.

Um resultado interessante de se analisar foi o das transformadas de Hough sobre a imagem *sample2.png*. As figuras

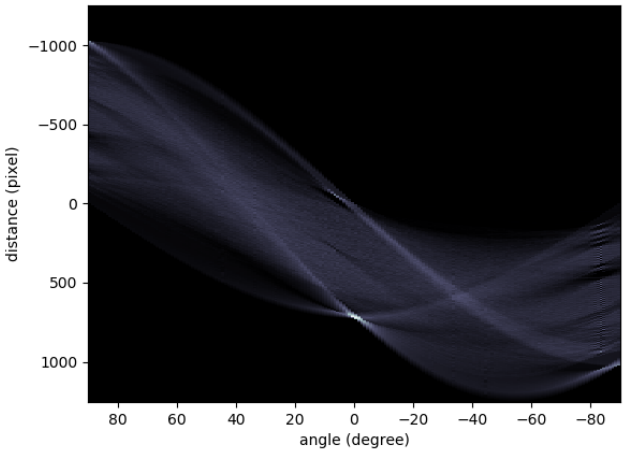

Figura 10. Transformada de Hough da imagem sample2.png

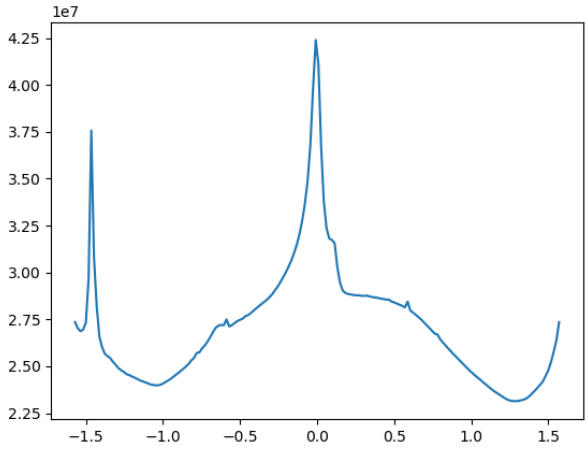

Figura 11. Projeção vertical dos quadrados das figura 10.

10 e 11 indicam que claramente existe uma concentração de retas verticais na imagem. Antes o programa ignorava a faixa de ângulos proposta pelo usuário no caso da transformada de Hough, o que fazia com que nesses casos a imagem fosse rotacionada quase 90 graus. O programa foi então modificado, e agora ele consegue ignorar essas retas e escolher a melhor. O que não é visível na figura 11 mas pode ser visto na 10 é que existe uma concentração alta em um único ponto nos ângulos próximos 90 e -90. Usando o método de escolher o maior pico, esse ângulo seria escolhido, mas o método da moda dos picos se mostrou mais robusto e conseguiu encontrar o resultado certo.

B. Tempo de execução

Usando como referência a figura 1, foi executado 5 vezes para o modo -hp, -fhp e -hough_m sempre buscando entre os ângulos -45 e 45. A média dos tempos pode ser vista na tabela II.

Tabela I
RESULTADOS DOS MODOS

	wlc	wgc	dp	dm	ent	ene	hough_m	hough_p	inclination
good_straight	0	0	0	0	0	0	0	0	0
neg_28	-28	-28	-28	-28	-28	-28	-28.16	-28.16	-28
neg_4	-27	-4	-4	-4	-4	-4	-4.02	-4.02	-4
pos_24	24	24	24	24	24	24	24.14	24.13	24
pos_41	41	41	41	41	41	41	41.23	41.23	41
sample1	14	14	14	14	14	14	14.08	14.08	14
sample2	-25	-18	0	-13	-6	-6	-6.03	0	-6
scanned	-45	-44	1	1	1	-36	2.01	0	1

Tabela II
TEMPO DE EXECUÇÃO PARA FIGURA 1

	-hp	-fhp	-hough_m
tempo (s)	7.23	19.74	0.96

Infelizmente a implementação atual do modo **-hp** sempre extrai todas as métricas, mesmo que apenas uma delas seja usada depois. Ainda assim, a extração de métricas de uma projeção é uma tarefa rápida, o mais demorado é rotacionar, transformar em binária e fazer a projeção para cada um dos ângulos, tanto que conseguimos ver um ganho evidente de performance no modo **-fhp**. Vale ressaltar que **-fhp** funciona apenas em casos onde a entropia se comporta de uma forma específica. Os resultados dele não foram apresentados, mas ele falhou para os arquivos *sample2.png e scanned.jpg*, logo ele não é muito confiável.

O tempo das duas implementações da transformada de Hough também variam muito pouco, por isso mostramos apenas o resultado do modo **-hough_m**, que é o mais confiável dos dois. Como podemos ver, ele é muito mais rápido do que os baseados em projeção horizontal. Isso provavelmente é uma consequência da execução em um computador com processador gráfico, pois a transformada de Hough não é barata computacionalmente, principalmente nessa implementação que não é feita sobre uma extração de bordas e sim sobre a imagem binária direta. Soluções baseadas em transformada de Hough entretanto possuem a vantagem de não precisar rotacionar a imagem para maximizar suas métricas.

C. Tesseract OCR

Na especificação do trabalho foi mencionado que o software Tesseract seria usado para identificar os textos das imagens com inclinação corrigidas, o que serviria avaliar o resultado. A implementação atual do programa não trata de nenhuma forma as imagens recebidas, apenas corrige a inclinação. Aplicando o Tesseract sobre a figura 1, nós podemos já perceber algumas das suas limitações. A figura não possui ruidos e está alinhada, entretanto o programa falhou em reconhecer algumas letras (um caso recorrente é onde ele reconhece a letra *m* como *rn*). Ainda assim, o resultado é impressionante, pois apesar dos pequenos erros, ainda é possível ler o texto e entender sua mensagem, fora isso, o Tesseract assume que o texto está em inglês, o que dificulta o reconhecimento de alguns caracteres como *ç*. Para as imagens n28, n4, p24 e p41 os resultados foram imprecionantes, com raros os casos onde a leitura foi comprometida. Para sample1, o código de barra provavelmente afetou o reconhecimento, pois por mais que o texto ainda seja legível, existem diversos erros. Para sample2 o resultado foi ilegível, e para scanned foi possível distinguir palavras, mas o texto ficou incompreensível.

V. CONCLUSÕES

Nesse trabalho aprendemos como processamento de imagem pode ser utilizado para identificar inclinações de textos. Conseguimos verificar que: assumindo uma imagem sem ruídos e artefatos, encontrar a inclinação e corrigi-la é uma tarefa muito simples quando utilizamos as técnicas propostas. Vimos que com as arquiteturas atuais isso pode ser feito de forma muito rápida através da transformada de Hough, e que em casos mais agressivos (imagens *poluídas*) a entropia da projeção horizontal se mostrou robusta o suficiente para ainda fornecer resultados bons.