

Trabalho 3

Ronnypetson Souza da Silva - RA 211848

Introdução ao Processamento Digital de Imagem

Professor: Hélio Pedrini

IC - Instituto de Computação

UNICAMP - Universidade Estadual de Campinas

1) Introdução

As atividades deste trabalho consistem na implementação de dois algoritmos para alinhamento de imagens com texto: um através de função objetivo sobre a projeção horizontal das imagens e o outro através do método conhecido como transformada de linhas de Hough. O alinhamento de imagens com texto é importante para o reconhecimento óptico de caracteres, pois textos desalinhados geralmente tornam inviável o reconhecimento dos caracteres. Textos desalinhados geralmente se originam na digitalização de documentos quando os textos não são posicionados de forma alinhada. Neste trabalho, o script desenvolvido é uma implementação em Python das duas técnicas citadas. Faz-se uso de bibliotecas especiais para a rotação de imagens sem recortes e para o uso de uma função que calcula entropia no método da projeção horizontal.

2) O script ‘alinhar.py’

O script ‘alinhar.py’ contém ambas as implementações dos métodos de projeção horizontal e linhas de Hough. Para escolher entre os dois métodos, deve-se especificar o parâmetro ‘modo’ na linha de comando. Outros parâmetros que devem ser especificados são o nome do arquivo de entrada (que deve estar no mesmo diretório do script) e o nome do arquivo de saída, como é mostrado abaixo.

```
$ python alinhar.py <imagem_entrada.png> <modo> <imagem_saida.png>
```

O parâmetro <imagem_entrada.png> deve conter o nome de uma imagem no formato .png dentro do diretório do script. <imagem_saida.png> é o nome do arquivo a ser criado com a imagem alinhada no diretório atual. Os valores que <modo> pode assumir são ‘hough’ e ‘proj’, que indicam o método das linhas de Hough e o método da projeção horizontal, respectivamente.

A seguir são descritas as etapas principais na execução desse script e são mostrados resultados de alguns casos de execução.

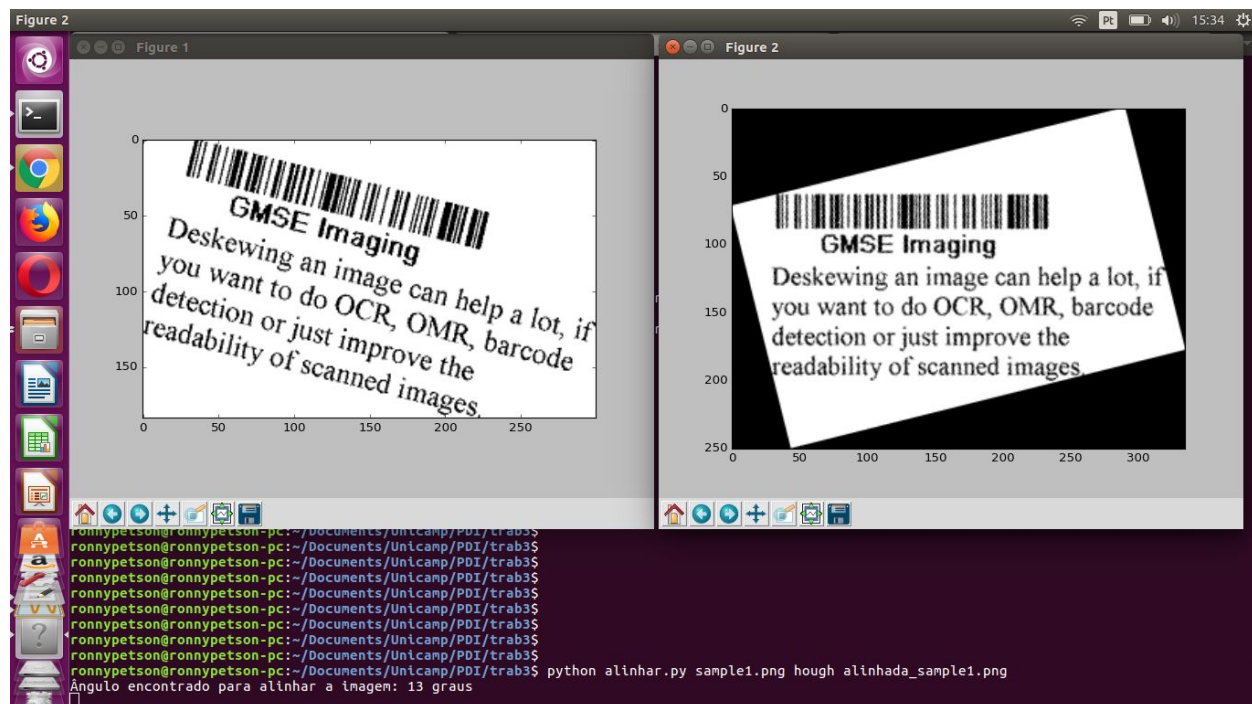
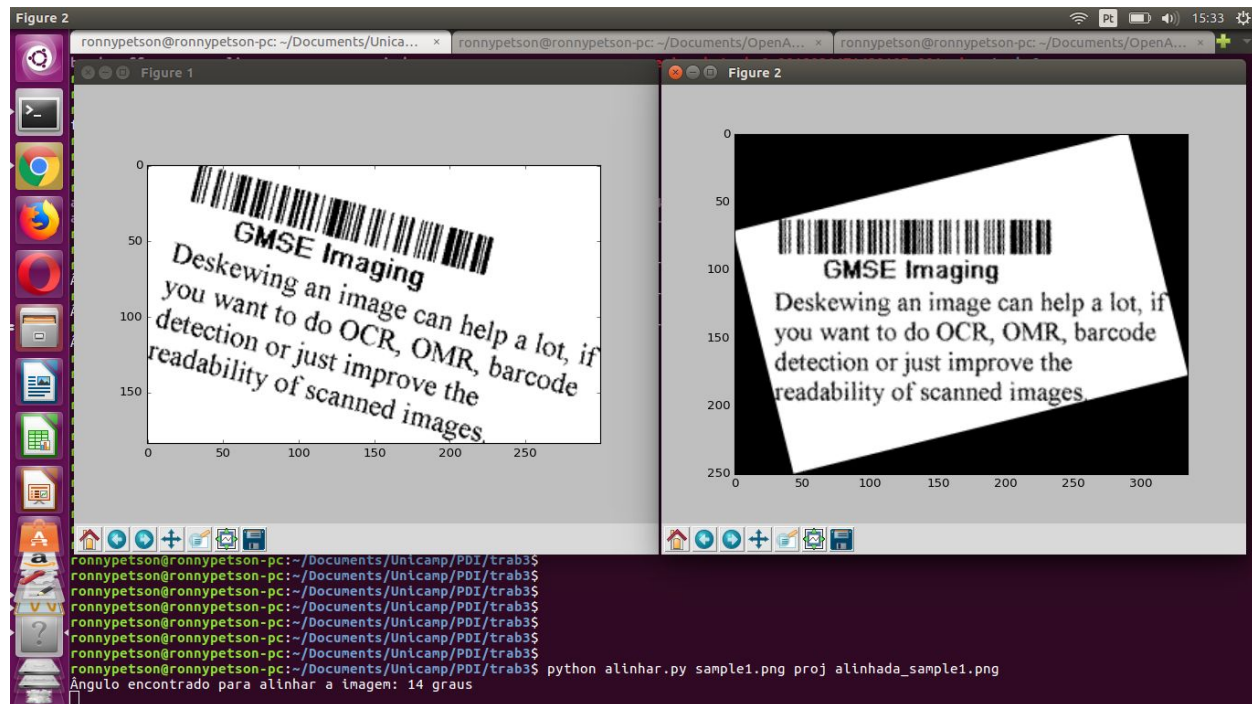
1. No caso em que os parâmetros são passados corretamente, a imagem de entrada é carregada e convertida para escala de cinza, a fim de simplificar o processo de alinhamento. Para fins de comparação visual com a imagem resultante, ambas as imagens (de entrada e resultante) são exibidas em janelas no final da execução do script.
2. Se o parâmetro ‘modo’ for ‘proj’, antes da execução do método são aplicados três passos de pré-processamento: filtro de mediana, binarização e obtenção da imagem negativa. O filtro de mediana é aplicado para remover ruídos do tipo “sal e pimenta”, enquanto que o negativo da imagem binarizada resulta em projeções melhores, pois em alguns casos o fundo preto obtido se combina com o fundo preto gerado ao se rotacionar a imagem, o que elimina a interferência do fundo com os caracteres na projeção horizontal nesses casos e, de modo geral, ameniza a

interferência (ver imagem a seguir). Daí é feita uma busca por ângulos no intervalo de -45 graus à 45 graus, inclusive, pois considera-se que, correções com ângulos de -90 à -45 graus e 45 à 90 graus só podem ser detectadas por projeções verticais. Como o método de projeção é horizontal, esses intervalos de ângulos foram descartados. O intervalo de 180 graus restante também pode ser descartado, pois cada transformação nesse intervalo possui uma correspondente na outra metade dos ângulos. Durante essa busca, os ângulos são testados com a ajuda da função `imutils.rotate_bound` da biblioteca `imutils`, que aplica a rotação na imagem sem cortá-la. Daí as projeções horizontais de cada ângulo são calculadas de forma vetorizada com a função `np.sum` da biblioteca `numpy` e a entropia da projeção é calculada pela função `scipy.stats.entropy` da `scipy`. Vale destacar que a entropia da projeção horizontal é a função objetivo escolhida no trabalho e que deseja-se encontrar o ângulo que resulte em uma projeção com entropia mínima.



3. Se o parâmetro 'modo' for 'hough', primeiro é aplicado o filtro Canny para detecção de bordas e então os pares (ρ , θ) da transformada de Hough para as linhas com maior acumulador na transformada são obtidos. O parâmetro de 'threshold' da transformada na função `cv2.HoughLines` foi ajustado manualmente após testes com as imagens fornecidas na descrição do trabalho e outras imagens buscadas na Internet. Visto que, os ângulos retornados estão em radianos, estes são convertidos para graus, daí seu complemento é colocado em uma lista (para alinhar as coordenadas com o eixo horizontal), para que então o mínimo dos ângulos seja usado, pois deve-se escolher entre os vários ângulos retornados.

4. O ângulo obtido é mostrado na saída padrão e a imagem em escala de cinza é rotacionada de acordo com o ângulo encontrado e salva no arquivo com nome especificado. A seguir estão as saídas para a imagem “sample1.png” para os modos ‘proj’ e ‘hough’, respectivamente.



- 3) As estruturas de dados** do trabalho são *arrays* de 1, 2 e 3 dimensões e estruturas de dados das bibliotecas usadas. Os arrays numpy possuem a propriedade *shape* das imagens, que são arrays de uma dimensão, enquanto que as imagens monocromáticas são arrays bidimensionais cujos valores vão de 0 à 255 (profundidade 8).
- 4) Considerações especiais.** O método da transformada de Hough pode, dependendo do parâmetro de threshold, não retornar nenhum par (ρ, θ). Portanto, o parâmetro usado para as imagens deste trabalho pode não funcionar com outras imagens. Outra consideração é que as bibliotecas numpy, scipy, imutils, cv2 e matplotlib são requisitos para a execução do script 'alinhar.py'.
- 5) Testes.** Testes foram realizados com as imagens baixadas no endereço http://www.ic.unicamp.br/~helio/imagens_inclinadas_png e algumas imagens buscadas no Google Imagens. Todas as imagens foram testadas em ambos os modos.
- 6) Sobre as limitações.** No Fedora 27, pode ser necessário atualizar o gerenciador de arquivos para ver todas as imagens geradas no diretório. Outra limitação diz respeito ao diretório do arquivo de entrada, pois estes devem estar na mesma pasta do script sendo executado.