

Introdução ao Processamento Digital de Imagem (MC920) - Trabalho 3

Heitor Boschirolli - 169477

2018
May

1 Executando o código

O tar `HeitorBoschirolli_169477.tar` contém o código `align.py` - responsável pelo processamento das imagens -, seis imagens para teste e este PDF.

Para executar o arquivo `align.py`, basta digitar `"python align.py"` em um terminal linux com python 2, seguido do nome da imagem que deve ser processada, do método utilizado para o alinhamento (0 ou 1) e do nome da imagem de saída. O método 0 corresponde ao alinhamento usando a projeção horizontal e o método 1 corresponde ao alinhamento usando a transformada de Hough. Os nomes da imagem de entrada e de saída devem conter sua extensão. Assim que o código for executado, uma imagem com o nome especificado (nome da imagem de saída) aparecerá no mesmo diretório em que o arquivo `align.py` se encontra; a imagem corresponde a imagem de entrada com uma rotação tal que o texto nela contido esteja alinhado.

Abaixo segue um exemplo de chamada do programa.

```
python align.py input.png 0 output.png
```

Onde `input.png` é o nome da imagem com o texto a ser alinhado e `output.png` é o nome com o qual a imagem contendo o texto alinhado deve ser salva.

2 Solução

2.1 Alinhamento usando a projeção horizontal

A projeção horizontal de uma imagem que contém um texto pode indicar se o texto está ou não alinhado. Se o texto está alinhado, o histograma da projeção horizontal terá picos seguidos de vales nas regiões com texto, como pode ser visto na figura 1 que é a projeção horizontal da imagem da figura

2, se o texto não estiver alinhado, o histograma na região do texto terá poucas variações entre as barras vizinhas, como pode ser visto na figura 3 que é a projeção horizontal da imagem da figura 4.

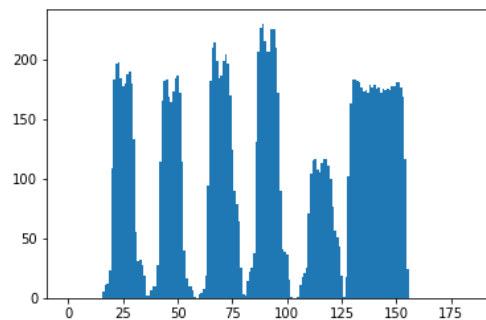


Figure 1: projeção horizontal da imagem da figura 2

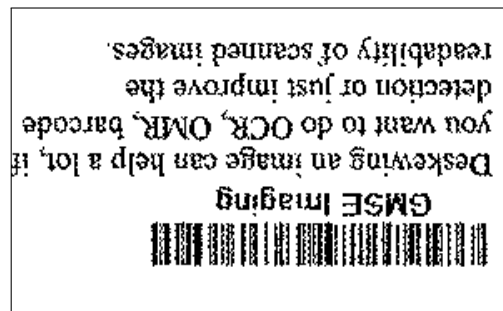


Figure 2: exemplo de imagem com texto alinhado (estar ou não de ponta cabeça é indiferente para o critério da projeção horizontal)

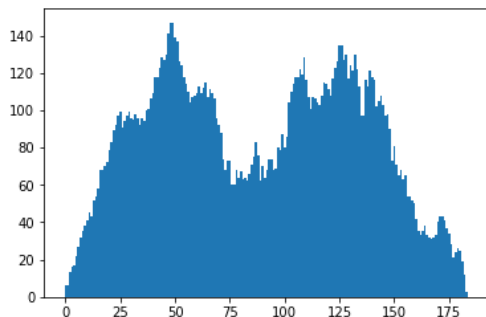


Figure 3: projeção horizontal da imagem da figura 4

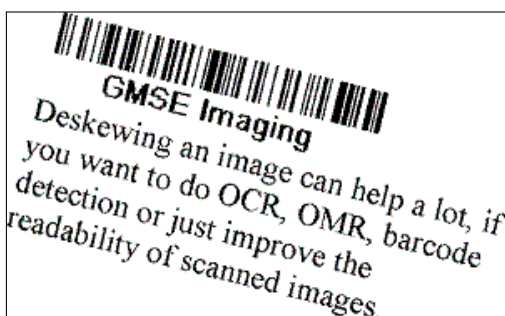


Figure 4: exemplo de imagem com texto não alinhado

Assim sendo, para alinhar o texto de uma imagem, rotacionou-a em diferentes ângulos, salvando qual era o ângulo que proporcionava o histograma da projeção horizontal mais adequado e salvou-se a imagem rotacionada com o melhor ângulo.

2.1.1 Pré-processamento

Para que a projeção horizontal fosse calculada, foi primeiro feita a binarização da imagem utilizando a função `adaptiveThreshold` da biblioteca OpenCV que utiliza o método de binarização local.

Como ruído podia ter uma alteração significativa no resultado da projeção, passou-se um filtro da mediana com um kernel 3x3 usando a função `medianBlur` da biblioteca OpenCV.

Por fim, foi feita a dilatação do texto (ou quaisquer pixels da cor do texto) para atenuar os efeitos da projeção horizontal. Com a dilatação, os picos do histograma gerado pela projeção horizontal tornam-se maiores. A dilatação foi feita com a função `dilate` da biblioteca OpenCV, foi usado um elemento estruturante 3x3 com todos os pixels ativos.

2.1.2 Função objetivo

Para determinar qual era o histograma mais adequado precisou-se de uma métrica (função objetivo). Como em um texto alinhado a projeção horizontal possui picos alternados de vales (figura 1), foi escolhida a soma dos quadrados das diferenças, cuja fórmula se encontra abaixo.

$$\sum_{i=2}^n (x_i - x_{i-1})^2$$

Quanto maior a soma dos quadrados das diferenças, maior a variação entre cada barra e seus vizinhos, portanto melhor foi considerado o histograma. O ângulo que maximizava a soma das diferenças dos quadrados foi o utilizado para rotacionar a imagem e gerar a imagem de saída.

2.2 Alinhamento usando a transformada de Hough

2.2.1 Pré-processamento

Para evitar a identificação de retas não desejadas no espaço de Hough, passou-se inicialmente um filtro da mediana com uma janela 3x3 para remoção de ruído; para isso foi usada a função `medianBlur` da biblioteca OpenCV.

Em seguida, foi usado o operador de bordas de Canny para detectar as bordas da imagem, o primeiro limiar foi estabelecido como 50 e o segundo 150; isso foi feito com a função `Canny` da biblioteca OpenCV. A detecção de bordas foi feita pois, ao usar somente as bordas para detectar linhas com a transformada de Hough conseguiu-se resultados igualmente satisfatórios e reduziu-se o custo da operação da transformada de Hough.

2.2.2 Uso das retas para rotacionar a imagem

A transformada de Hough foi obtida com a função `HoughLines` da biblioteca OpenCV, essa função devolve as retas encontradas utilizando a transformada de Hough. Para saber o ângulo de inclinação do texto na imagem, escolheu-se a reta com maior número de pontos correspondentes na imagem original; com a inclinação dessa reta, bastou rotacionar a imagem em uma mesma quantidade no sentido contrário para alinhá-la, isso foi feito com as funções `getRotationMatrix2D` e `warpAffine`, ambas da biblioteca OpenCV.

3 Exemplo de entrada e saída

Para o caso em que a imagem de entrada é input.png (figura 5) e o método usado é o da projeção horizontal, abaixo encontra-se a chamada do programa e na figura 6 a imagem resultante.

```
python align.py input.png 0 out0.png
```

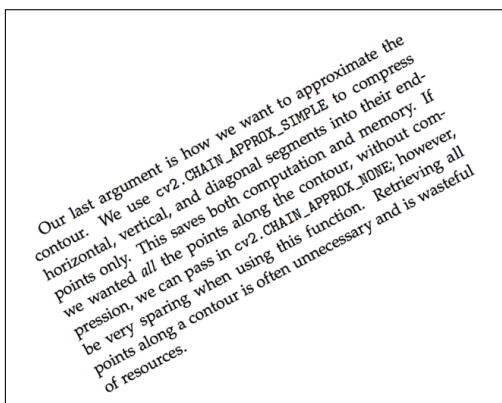


Figure 5: imagem input.png

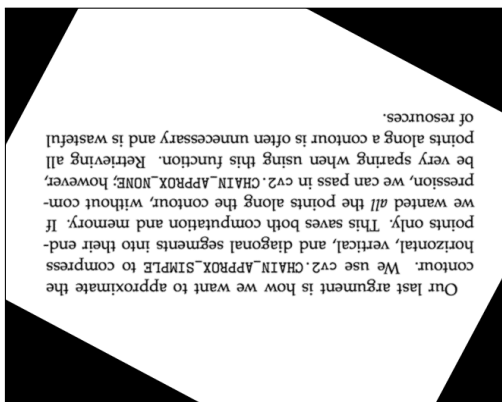


Figure 6: imagem out0.png gerada como saída usando o método da projeção horizontal

Para o caso em que a imagem de entrada é input.png (figura 5) e o método usado é o da transformada de Hough, abaixo encontra-se a chamada do programa e na figura 7 a imagem resultante.

```
python align.py input.png 1 out1.png
```

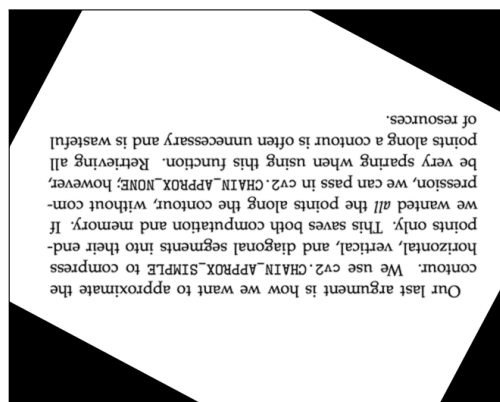


Figure 7: imagem out1.png gerada como saída usando o método da transformada de Hough

4 Limitações

Nenhum dos dois métodos é capaz de diferenciar textos alinhados corretamente e rotacionados em 180 graus, ou seja, é possível que o resultado seja uma imagem com texto de ponta cabeça (como nos exemplos da seção anterior); no caso do método que utiliza a transformada de Hough, também não é possível diferenciar textos rotacionados em 90 ou 270 graus, ou seja, a imagem de saída usando o método baseado em Hough pode ser um alinhamento em 0, 90, 180 ou 270 graus.

O método baseado na transformada de Hough é bastante sensível a desenhos presentes na imagem; se desenhos com retas em orientações quaisquer estão presentes na imagem, a correção de ângulo pode ser feita com base em alguma dessas retas, resultando em um alinhamento incorreto.

Em imagens muito grandes o método baseado na projeção horizontal pode apresentar problemas relacionados ao tempo de execução. Nos testes feitos localmente, para rotacionar uma imagem 727x1024 foram necessários quase 4 segundos, sugerindo que o método não é muito adequado para rotacionar um grande número de imagens em seguida.