Trabalho_03

October 27, 2021

MC920 / MO443 (Introdução ao Processamento Digital de Imagem)

Prof. Hélio Pedrini

Matheus Xavier Sampaio - RA 220092

1 Instruções

1.1 Arquivos

O arquivo `Trabalho_03.ipynb` possui um notebook executável com os códigos e o relatório do trabalho.

O arquivo `Trabalho_03.pdf` possui o notebook em formato `pdf` com o relatório do trabalho.

O arquivo `Trabalho_03.py` possui um script com os códigos do trabalho.

1.2 Ambiente e Execução

O arquivo `environment.yml` pode ser utilizado para criar um ambiente `conda` com todas as dependencias para executar o trabalho.

Caso tenha o Anaconda instalado, basta executar o comando: `$ conda env create`

Apos instalar o ambiente, ative usando o comando: `$ conda activate mo443`

Por fim, execute o script python para criar as imagens com a visualização de cada questão: `$ python Trabalho_03.py`. Para ver as opções possíveis para a execução do script, execute `$ python Trabalho_03.py -h`.

Se preferir, abra o notebook e execute uma célula por vez.

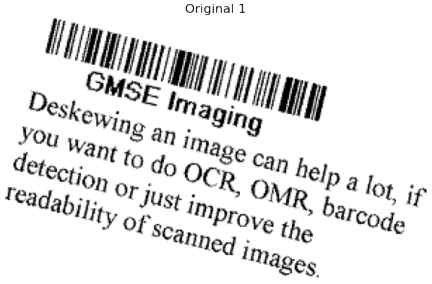
2 Alinhamento de texto na imagem

2.1 Técnica Baseada em Projeção Horizontal

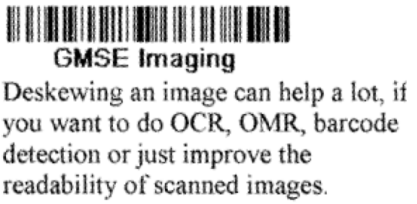
A detecção de inclinação baseada em projeção horizontal é realizada variando o ângulo testado e proje- tando a quantidade de pixels pretos em cada linha linha de texto. O ângulo escolhido é aquele que otimiza uma função objetivo calculada sobre o perfil da projeção horizontal. Um exemplo de função objetivo é a soma dos quadrados das diferenças dos valores em células adjacentes do perfil de projeção.

projecao_horizontal

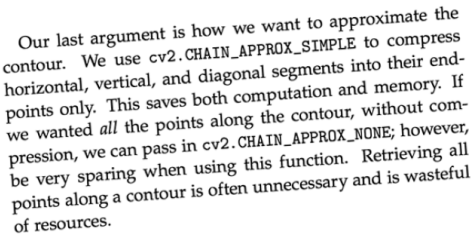
Original 1



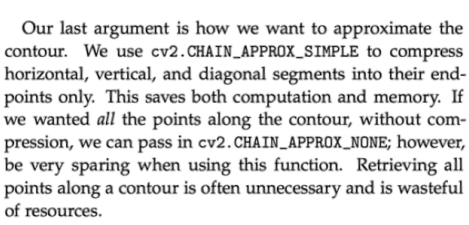
Rotação em 14°



Original 2



Rotação em -4°



Texto imagem original 1:

Texto imagem 1 rotacionada em 14°:
AUREL TN

GMSE Imaging
Deskewing an image can help a lot, if
you want to do OCR, OMR, barcode
detection or just improve the
readability of scanned images.

Texto imagem original 2:
Our last argument is how we want to approximate the
contour. We use v2. CHAIN_APPROX_SIMPLE to compress
horizontal, vertical, and diagonal segments into their end-
points only. This saves both computation and memory. If
we wanted all the points along the contour, without com-
pression, we can pass jn cv2 "CHAIN_APPROX_NONE; however,
be very sparing when using, this function. Retrieving all
points along a contour is often unnecessary and is wasteful
of resources.

Texto imagem 2 rotacionada em -4°:

```
Our last argument is how we want to approximate the
contour. We use cv2.CHAIN_APPROX_SIMPLE to compress
horizontal, vertical, and diagonal segments into their end-
points only. This saves both computation and memory. If
we wanted all the points along the contour, without com-
pression, we can pass in cv2. CHAIN_APPROX_NONE; however,
be very sparing when using this function. Retrieving all
points along a contour is often unnecessary and is wasteful
of resources.
```

Idéia: Ao testar várias rotações na imagem, calculando a quantidade de píxels contendo informação referentes ao texto em cada linha, identificar o angulo que melhor representa a rotação necesária para alinhar a imagem.

Implementação: Para acentuar o texto na imagem, foi calculado um limiar para transformar a imagem em tons de cinza em binária. Esse limiar é calculado utilizando a função `skimage.filters.threshold_otsu`, que utiliza do histograma da imagem para calcular um limiar. Este limiar é então utilizado para zerar os valores abaixo e elevar ao máximo os valores acima, de fato binarizando e invertendo a imagem. Com a imagem ajustada, é calculada a projeção e custo para angulo, da seguinte maneira: - Para cada valor de ângulo inteiro θ de $[-90, 90]$, a imagem é rotacionada em θ utilizando a função `skimage.transform.rotate`. - Com a imagem rotacionada, é calculada a sua projeção na forma da soma de cada linha, aplicando sobre o resultado a função de custo na forma do quadrado da soma das diferenças entre cada linha e seu sucessor. - Com o valor de custo calculado para cada ângulo, o θ com maior valor é escolhido.

Chamada e parâmetros: A função `horizontal_projection` recebe a imagem binarizada e o ângulo de rotação, retornando a imagem rotacionada, o perfil de projeção, o ângulo de rotação e o valor.

A função `allign_image_horizontal_projection` recebe a imagem e retorna esta imagem alinhada e o ângulo utilizado para o alinhamento.

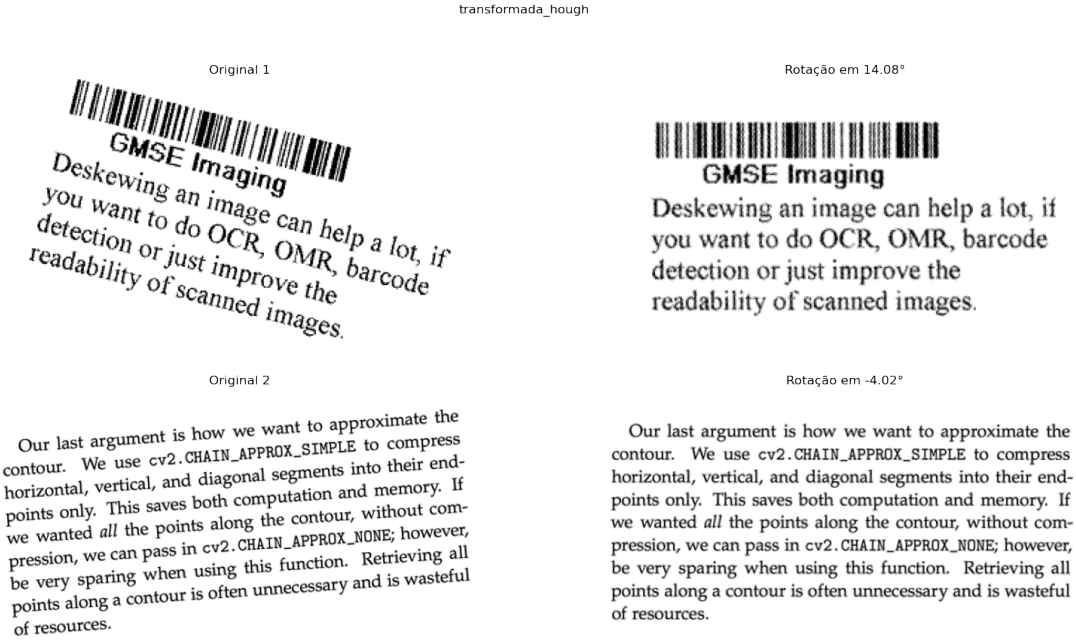
Limitações: O ângulo de rotação da imagem deve estar entre $[-90, 90]$ graus.

Resultados: Observando as imagens, podemos ver que a técnica foi capaz de alinhar as imagens com rotações positivas e negativas de forma bastante satisfatória. Ao aplicar este alinhamento como uma etapa de pre-processamento para técnicas de `OCR`, observamos que para a imagem 1 original não foi possível obter o seu texto, mas ao aplicar uma rotação de 14°, foi possível extrair o texto da imagem por completo. Para a imagem 2, é possível perceber alguns erros de reconhecimento, com alguns caracteres faltantes, simbolos e palavras erradas, que são corrigidos ao aplicar uma rotação de $-4°$ a imagem.

2.2 Técnica Baseada na Transformada de Hough

A detecção de inclinação da imagem baseada na transformada de Hough assume que os caracteres de texto estão alinhados. As linhas formadas pelas regiões de texto são localizadas por meio da transformada de Hough, a qual converte pares de coordenadas (x, y) da imagem em curvas nas coordenadas polares (ρ, θ).

Pixels pretos alinhados na imagem (linhas dominantes) geram picos no plano de Hough e permitem identificar o ângulo de inclinação do documento. Uma estrutura é empregada para acumular o número de vezes em que a combinação de e ocorre na imagem. A granularidade do espaço de busca depende do grau de precisão do eixo .




```
Texto imagem original 1:


Texto imagem 1 rotacionada em 14.08°:
AU LARCUE MT

GMSE Imaging
Deskewing an image can help a lot, if
you want to do OCR, OMR, barcode
detection or just improve the
readability of scanned images.


Texto imagem original 2:
Our last argument is how we want to approximate the
contour. We use v2. CHAIN_APPROX_SIMPLE to compress
horizontal, vertical, and diagonal segments into their end-
points only. This saves both computation and memory. If
we wanted all the points along the contour, without com-
pression, we can pass jn cv2 "CHAIN_APPROX_NONE; however,
be very sparing when using, this function. Retrieving all
points along a contour is often unnecessary and is wasteful
```

```
of resources.


Texto imagem 2 rotacionada em -4.02°:
Our last argument is how we want to approximate the
contour. We use cv2.CHAIN_APPROX_SIMPLE to compress
horizontal, vertical, and diagonal segments into their end-
points only. This saves both computation and memory. If
we wanted all the points along the contour, without com-
pression, we can pass in cv2.CHAIN_APPROX_NONE; however,
be very sparing when using this function. Retrieving all
points along a contour is often unnecessary and is wasteful
of resources.
```

Idéia: Ao aplicar a Transformada de Hough em uma imagem com textos, espera-se as linhas horizontais formadas pela transformada seguirão o alinhamento do texto. Utilizando a inclinação destas linhas, podemos encontrar o ângulo de inclinação necessário para alinhar horizontalmente a imagem.

Implementação: Assim como na Projeção Horizontal, podemos melhorar o alinhamento acentuando o texto. Para a Transformada de Hough, aplicamos um filtro de detecção de bordas a imagem e fornecemos a saída ao algoritmo. O filtro escolhido foi o Sobel, utilizando a função `skimage.filters.sobel`. Apos o filtro, aplicamos a função `skimage.transform.hough_lines` que aplica a transformada de hough de linhas retas na imagem. A contagem de linhas, seus angulos e distâncias são usados pela função `skimage.transform.hough_line_peaks` para calcular os seus picos no plano de Hough e permitem e os ângulos de inclinação. A partir destes ângulos, o mais frequente é selecionado e transformado de radiano para graus, aplicando um ajuste de $\pm 90°$, pois o ângulo resultante é referente ao eixo vertical.

Chamada e parâmetros: A função `hough_transform` recebe a imagem original e retorna o ângulo de rotação, retornando a imagem rotacionada, o acumulador´da quantiade de vezes que cada pico ocorre na imagem, os ângulos das transformadas e as distancias entre as linhas formadas pelas regiões de texto.

A função `allign_image_hough_transform` recebe a imagem e retorna esta imagem alinhada e o ângulo utilizado para o alinhamento.

Limitações: O ângulo de rotação da imagem deve estar entre $[-90, 90]$ graus.

Resultados: Os resultados observados são bastantes similares aos da Projeção Horizontal, com os ângulos encontrados diferindo em casas decimais. Isso ocorre devido a nosso espaço de busca na Projeção Horizontal consistir em apenas inteiros. Na aplicação de `OCR`, pelos angulos serem quase identicos, temos os mesmos resultados, atentanto apenas a imagem 1, que teve uma transcrição referente ao código de barras diferente.

2.3 Limitação das Técnicas

Uma limitação encontrada nas técnicas é que estas não conseguem lidar com rotações fora do limite de $[-90, 90]$ graus, o que gera imagens "de cabeça para baixo" quando aplicadas a imagens com

inclinação fora do limite.