

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.

1.2 Técnica Baseada em Projeção Horizontal

A detecção de inclinação baseada em projeção horizontal é realizada variando o ângulo testado e projetando a quantidade de pixels pretos em cada 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.

A Figura 1(a) mostra uma imagem com inclinação de 10 graus no sentido anti-horário e sua respectiva projeção horizontal, enquanto a Figura 1(b) mostra a imagem após a correção da inclinação e a projeção horizontal correspondente. O histograma possui amplitude máxima quando o texto na imagem é posicionado em 0 grau com respeito ao eixo horizontal, uma vez que o número de pixels colineares é máximo nessa situação.



Figura 1: (a) Imagem com inclinação de 10 graus no sentido anti-horário e (b) imagem após correção da inclinação.

O grau de precisão da técnica pode ser controlado com a mudança gradual do passo entre os ângulos testados. A remoção de certos componentes do documento, tais como figuras e tabelas, pode reduzir o custo computacional e melhorar o desempenho da técnica baseada em projeção.

O Algoritmo 1 apresenta os principais passos da técnica baseada em projeção horizontal. A partir da imagem de entrada, o algoritmo calcula a quantidade de pixels pretos para cada linha da imagem e retorna o ângulo de inclinação da imagem.

Algoritmo 1: Algoritmo de detecção de inclinação baseada em projeção horizontal.

Entrada: F : imagem de entrada**Saída:** θ_M : ângulo de inclinação

```
1 início
2    $n = \text{número de linhas da imagem}(F)$ 
3   para  $i = 1$  até  $n$  faça
4      $\text{Perfil}[i] = \text{soma}(F.\text{linhas}[i])$ 
5   fim
6   para  $\theta = \theta_1$  até  $\theta_2$  faça
7      $\text{valor}[\theta] = \text{função\_objetivo}(\text{Perfil})$ 
8   fim
9    $\theta_M = \max(\text{valor})$ 
10  retorna  $\theta_M$ 
11 fim
```

1.3 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 θ do documento. Uma estrutura é empregada para acumular o número de vezes em que a combinação de ρ e θ ocorre na imagem. A granularidade do espaço de busca depende do grau de precisão do eixo θ .

O Algoritmo 2 apresenta os principais passos da técnica baseada na transformada de Hough. A função objetivo é aplicada sobre o acumulador para encontrar o ângulo de inclinação.

Algoritmo 2: Algoritmo de detecção de inclinação baseada na transformada de Hough.

Entrada: F : imagem de entrada**Saída:** θ_M : ângulo de inclinação

```
1 início
2    $n = \text{número de linhas da imagem}(F)$ 
3    $m = \text{número de colunas da imagem}(F)$ 
4   para  $i = 1$  até  $n$  faça
5     para  $j = 1$  até  $m$  faça
6       if  $F[i, j]$  é preto then
7          $\text{Acumulador} = \text{Transformada\_Hough}(i, j)$ 
8       end
9     fim
10  fim
11   $\theta_M = \text{função\_objetivo}(\text{Acumulador})$ 
12  retorna  $\theta_M$ 
13 fim
```

Alguns exemplos de função objetivo são a maximização da taxa de mudança nos valores do acumulador para cada ângulo, o ângulo médio das linhas dominantes encontradas pela transformada de Hough, o ângulo correspondente à célula do acumulador com valor máximo ou um critério de votação sobre os ângulos de todas as linhas encontradas pela transformada de Hough.

1.4 Forma de Execução

Uma sugestão para execução do programa é dada a seguir:

```
python alinhar.py imagem_entrada.png modo imagem_saida.png
```

em que os parâmetros são:

- `alinhar.py`: programa que realiza o alinhamento da imagem.
- `imagem_entrada.png`: imagem no formato PNG antes do alinhamento.
- `modo`: técnica utilizada no alinhamento da imagem.
- `imagem_saida.png`: imagem no formato PNG após o alinhamento.

Para a técnica baseada em transformada Hough, aplique inicialmente um detector de bordas (por exemplo, Sobel) na imagem de entrada para gerar um mapa de bordas, o qual poderá ser utilizado pela transformada para encontrar linhas na imagem.

O código deve retornar o ângulo (em graus) detectado para alinhar a imagem de entrada com respeito ao eixo horizontal, bem como a imagem após a correção do ângulo de inclinação.

Um sistema de reconhecimento automático de caracteres pode ser aplicado às imagens de entrada e saída para verificar seu desempenho antes e após o processo de alinhamento das imagens. Um exemplo de código aberto para reconhecimento de caracteres em imagens é o Tesseract OCR (*Optical Character Recognition*).

2 Entrada de Dados

As imagens de entrada estão no formato PNG (*Portable Network Graphics*). Alguns exemplos encontram-se disponíveis no diretório: http://www.ic.unicamp.br/~helio/imagens_inclinadas_png/.

3 Saída de Dados

As imagens de saída devem estar no formato PNG (*Portable Network Graphics*). Resultados intermediários podem ser também exibidos na tela.

4 Especificação da Entrega

- A entrega do trabalho deve conter os seguintes itens:
 - código fonte: o arquivo final deve estar no formato *zip* ou no formato *tgz*, contendo todos os programas ou dados necessários para sua execução.
 - relatório: deve conter uma descrição dos algoritmos e das estruturas de dados, considerações adotadas na solução do problema, testes executados, eventuais limitações ou situações especiais não tratadas pelo programa.
- O trabalho deve ser submetido por meio da plataforma *Google Classroom*.
- Data de entrega: 22/05/2024.

5 Observações Gerais

- Os programas serão executados em ambiente Linux. Os formatos de entrada e saída dos dados devem ser rigorosamente respeitados pelo programa, conforme definidos anteriormente. Não serão aceitos trabalhos após a data de entrega.
- Os seguintes aspectos serão considerados na avaliação: funcionamento da implementação, clareza do código, qualidade do relatório técnico.