

Correção de inclinação de imagens de documentos baseadas na técnica de projeção horizontal e transformada de Hough.

Stephane de Freitas Schwarz

Hélio Pedrini

I. INTRODUÇÃO

Um dos problemas mais recorrentes no processo de digitalização de documentos é a correção da orientação da imagem resultante. Atualmente existem diversas técnicas capazes de reparar a inclinação de forma automática, isso é, sem a necessidade de um operador. Nesse trabalho foram exploradas duas técnicas para detecção e correção da inclinação de documentos digitais contendo textos baseados na análise de projeção horizontal e na transformada de Hough. O algoritmo proposto foi testado nas amostras disponíveis no repositório da disciplina¹, e funciona perfeitamente para todos os casos.

II. ARCABOUÇO PROPOSTO

Entender o problema e cogitar soluções para resolvê-lo não é uma tarefa muito complicada, contudo desenvolver um método eficiente para estimar o ângulo de inclinação da imagem não é trivial. A ideia central desses algoritmos consiste basicamente em duas etapas, identificar o ângulo de inclinação e rotacionar a imagem até que satisfaça a condição ótima. Nas próximas subseções serão apresentados os passos tomados no desenvolvimento dos algoritmos supramencionados, limitações e resultados obtidos.

a. Projeção horizontal

Esse método supõe que os documentos são constituídos majoritariamente por linhas que representam um texto. Assim, regiões que contém linhas tendem a apresentar uma maior quantidade de pixels pretos (assumindo que o fundo é branco) se sua angulação for zero em relação ao eixo. A figura 1.a ilustra a projeção horizontal de uma imagem alinhada e a figura 1.b uma rotacionada. É possível perceber que a amplitude da imagem inclinada a 0° (grau) é maior em relação à outra.



Figura 1: Imagens com diferentes inclinações e suas respectivas projeções horizontais.

A projeção é calculada pela somatória da quantidade de pixels pretos em cada linha da matriz, assim a forma mais rudimentar para determinar a inclinação correta da imagem é determinar um intervalo de ângulos e para cada um computar a projeção. Por fim usando uma função de custo, avaliar qual é o valor ótimo, nesse trabalho o critério adotado foi o desvio padrão. Essa decisão foi tomada, pois quando o documento está na orientação correta, o desvio padrão entre as somatórias das linhas é maior, isso ocorre porque há mais regiões com apenas a cor branca. No entanto quando a imagem está de alguma forma, fora do eixo, a distribuição dos pixels pretos entre as linhas da matriz é maior, em outras palavras o desvio padrão é menor.

¹ http://www.ic.unicamp.br/~helio/imagens_inclinadas_png/

Um ponto importante nessa abordagem é como converter uma imagem monocromática em binária. Embora não pareça ser um ponto crítico, a escolha do limiar para converter a imagem é fundamental para o resultado final, especialmente para o reconhecimento de caracteres. Usar um valor alto ou baixo pode prejudicar o processo de determinação do ângulo ideal, dado que valores pertencentes ao texto podem ser interpretados como parte do fundo. A fim de evitar tal impasse, o método de Otsu foi aplicado para gerar o valor do limiar de maneira automática.

Nessa abordagem não é preciso computar as projeções para todos os ângulos, basta variar de 90 a -90 com isso todos os possíveis casos serão avaliados. Não é difícil perceber que esse processo é bastante custoso, com o propósito de minimizar os impactos causados pelo excessivo uso de memória, ao invés de criar um array de duas dimensões para armazenar o ângulo e o desvio padrão de uma projeção, um vetor de apenas uma dimensão armazena os valores do desvio padrão e cada índice representa o ângulo que gerou tal valor.

Esse algoritmo foi testado com algumas imagens disponibilizadas em², o resultado obtido foi satisfatório, contudo quando as imagens resultantes são submetidas a um método de reconhecimento de caracteres a conversão imagem texto não fica perfeita, algumas falhas são inevitáveis, e isso se justifica pois no processo de transformação de imagem monocromática para binária pixels do texto foram perdidos e representados como fundo ou o documento digitalizado contém ruídos (marcas deixadas da digitalização, logotipos etc), por isso o algoritmo de reconhecimento de caracteres tem dificuldade de identificar corretamente algum elementos do texto.

b. Transformada de Hough

A transformada de Hough para detecção de linhas retas em uma imagem se baseia no fato de que para um ponto qualquer (x, y) existem infinitas retas que passam por tal região. Cada pixel pode pertencer a uma linha, assim os valores das coordenadas cartesianas são convertidos em curvas nas coordenadas polares, dessa forma quando os pixels pretos da imagem estiverem alinhados, caracterizando uma linha, haverá picos no plano de Hough. Esse processo pode ser entendido como uma votação, onde cada pixel preto vota em uma potencial linha, por fim basta apenas detectar as linhas com mais votos.

Com o advento das bibliotecas de processamento de imagem, essas funções estão disponibilizadas para uso na maioria dos pacotes. Com isso o processo se resume basicamente em dois passos, identificar linhas retas e determinar quais são os picos. Na primeira etapa basta passar uma imagem para a função *hough_line* disponível na biblioteca *skimage*, essa função retorna além dos valores da transformada acumulada, os ângulos e distâncias computadas. Após isso, identificar as linhas mais adequadas separadas por certo ângulo e distância, isso é os picos nas coordenadas de Hough com a função *hough_line_peaks* que por sua vez retorna os valores dos picos, ângulos e distâncias. Para resolver o problema da inclinação de documentos basta obter o ângulo máximo em graus calculado, isso implica que usando a implementação pronta dessas funções é preciso converter os valores de radianos em graus. A figura 2 ilustra a projeção da transformada de Hough calculada para a imagem da mais a esquerda.

2 http://www.ic.unicamp.br/~helio/imagens_inclinadas_png/

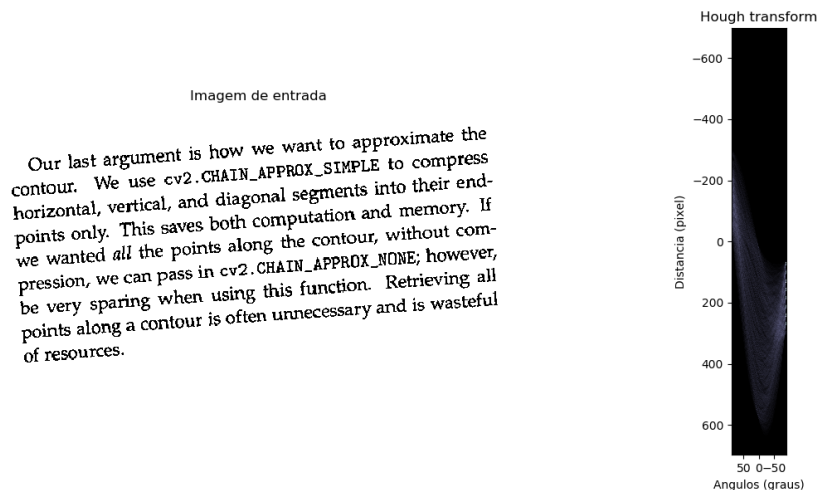


Figura 2: Resultado da aplicação da transformada de Hough (imagem da direita) na imagem da esquerda. Tempo de execução 53 milissegundos.

A imagem de entrada foi convertida para fundo branco conteúdo preto apenas para visualização, pois o algoritmo calcula suas projeções através de pontos não pretos, isso implica que a imagem deve estar no formato fundo preto conteúdo branco. É importante salientar que não é preciso aplicação qualquer detector de bordas na imagem para que o algoritmo funcione corretamente em casos mais simples, entretanto como é possível observar na figura 3, a projeção fica muito mais evidente após o pré processamento da imagem. Utilizando um detector de bordas a transformada de Hough tende a encontrar mais linhas (neste trabalho foi usado o filtro sobel), com isso a probabilidade de a imagem não ser rotacionada da maneira correta diminui, isso se aplica principalmente a imagens com borrões, imagens ou marcas, como é o caso da imagem rotulada *sample2.png* do repositório supramencionado. Com relação ao tempo de execução, as imagens binárias sem filtros de bordas são processadas em média duas vezes mais rápido.

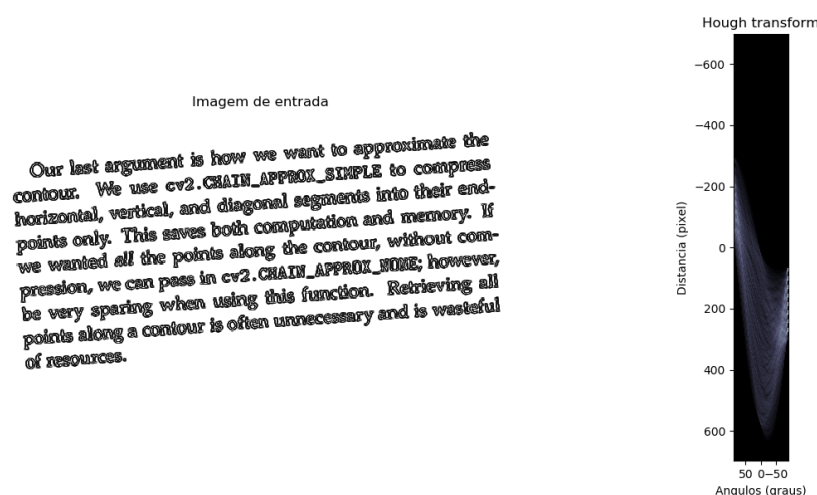


Figura 3: Resultado da aplicação da transformada de Hough (imagem da direita) na imagem da esquerda na qual foi aplicado um filtro sobel para detecção de bordas. Tempo de execução 85 milissegundos.

c. Reconhecimento de caracteres

Após a imagem ser rotacionada para o ângulo correto, foi aplicado um algoritmo de reconhecimento de caracteres, mais especificamente o pytesseract. O resultado obtido foi satisfatório para os

documentos que não contém informações ruidosas como borrões e marcas deixadas no processo de digitalização. Para essas o reconhecimento foi péssimo, a ponto de não ser possível interpretar o conteúdo originado.

III. EXECUTAR O PROGRAMA

Abra o terminal ou prompt de comando e digite *python horizontal_projection.py imagem.png* para executar o código da projeção horizontal. Para o algoritmo baseado na transformada de Hough digite *python hough_transform.py imagem.png*.

IV. REQUISITOS

O algoritmo foi construído para ser executado em qualquer computador desde que respeite as seguintes especificações: Python versão 3 ou superior e bibliotecas scipy, matplotlib, numpy e pytesseract.