

## Trabalho 1: Homografia Planar

Diogo Campos da Silva

### Introdução

Este trabalho teve como objetivo implementar a remoção de distorção perspectiva de fotos contendo folhas de papel, para obter uma aproximação do resultado que seria obtido com um scanner de documentos.

### Implementação

A aplicação desenvolvida neste trabalho foi implementada na linguagem JavaScript, por ser uma linguagem com a qual eu tenho bastante familiaridade, e porque ela simplifica a construção e distribuição de programas que podem ser executados em qualquer dispositivo que tenha um navegador web.

A funcionalidade principal da aplicação está implementada nos arquivos do diretório *Codigo/src/lib* e começa pela função *rectifyImage*, no arquivo *rectify.js*. Essa função, por sua vez, chama as funções *findHomography*, que calcula a matriz  $H_{3 \times 3}$  da homografia planar que retifica a região da imagem que contém a folha de papel, e *applyHomography*, que aplica a matriz e retorna a imagem resultante.

O cálculo da matriz  $H$  é feito através da solução de um sistema de oito equações lineares — duas equações para cada canto da região retangular correspondente à folha de papel. Cada canto com coordenadas  $c_i = (u_i, v_i)$  na imagem original corresponde a um ponto  $p_i = (x_i, y_i)$  na imagem que queremos obter. Para cada  $i \in \{0, 1, 2, 3\}$ , temos:

$$\begin{aligned}x_i a_1 + y_i a_2 + a_3 - u_i x_i a_7 - v_i y_i a_8 &= u_i \\x_i a_4 + y_i a_5 + a_6 - v_i x_i a_7 - v_i y_i a_8 &= v_i\end{aligned}$$

onde  $a_1, a_2, \dots, a_8$  são coeficientes da matriz  $H$  e os pontos  $p_i$  são os cantos da imagem desejada, ordenados no sentido horário a partir do canto superior esquerdo:

$$H = [a_1 \ a_2 \ a_3 \ ; \ a_4 \ a_5 \ a_6 \ ; \ a_7 \ a_8 \ 1]$$

$$p_0 = (0, 0), \ p_1 = (w, 0), \ p_2 = (w, h), \ p_3 = (0, h)$$

A largura  $w$  da imagem de saída é determinada pela maior das distâncias euclidianas  $d(c_0, c_1)$  entre os cantos superiores e  $d(c_2, c_3)$  entre os cantos inferiores da região selecionada na imagem original. A altura  $h$  é então calculada para obter uma proporção (*aspect ratio*) igual à de uma folha A4, cujo tamanho padrão é de  $210 \times 297$  mm:

$$w = \max \{d(c_0, c_1), d(c_2, c_3)\}$$

$$h = \frac{297}{210}w$$

O sistema linear de oito equações é resolvido utilizando o método da eliminação de Gauss–Jordan, e sua solução fornece os oito coeficientes desconhecidos da matriz  $H$ .

Para obter cada pixel da imagem resultante, multiplica-se a matriz  $H$  pelas coordenadas homogêneas  $(x, y, 1)$  do pixel, o que resulta nas coordenadas homogêneas  $(ku, kv, k)$  do ponto correspondente  $(u, v)$  na imagem original. Como esse processo geralmente resulta em coordenadas não-inteiras, os valores finais de intensidade do pixel são então calculados por interpolação bilinear.

## Utilização

A aplicação pode ser executada localmente, bastando abrir o arquivo *index.html* na pasta *Codigo/dist* em um navegador web, ou acessada online pelo endereço:

<https://diogocampos.github.io/rectify/>

A utilização consiste dos seguintes passos:

1. Abrir a imagem a ser corrigida:
  - clicar no botão **Imagem...** e escolher o arquivo na janela que aparece, ou
  - arrastar o arquivo e soltar dentro da aplicação.
2. Selecionar a região correspondente à folha de papel, posicionando os cantos da seleção sobre os cantos do papel.
3. Clicar no botão **Retificar**.
4. Quando a imagem resultante aparecer, clicar no botão **Salvar** para obter o arquivo.

A Figura 1, a seguir, mostra uma captura de tela da aplicação após a seleção da região desejada e a conclusão do processo de retificação da imagem fornecida.

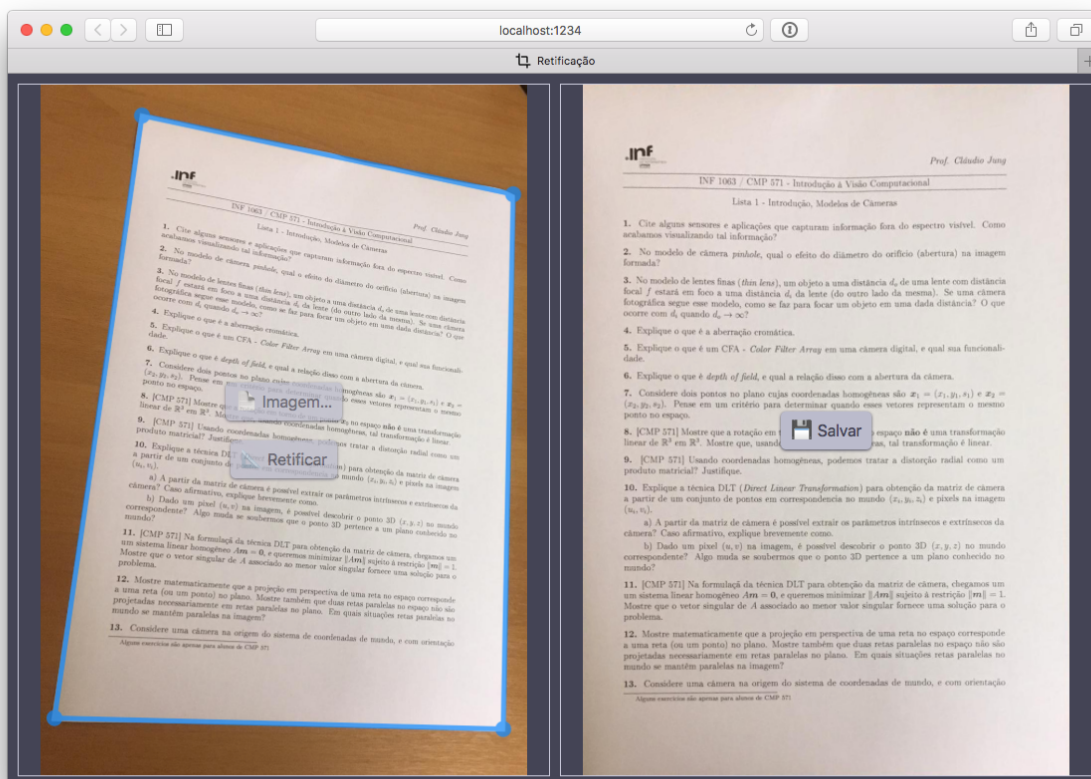


Figura 1. Captura de tela.

O arquivo *Codigo/README.md* também contém, além da explicação acima, instruções de como recompilar a aplicação a partir do código fonte.

## Resultados

A aplicação foi testada com as imagens fornecidas junto à especificação deste trabalho, além de mais algumas fotos tiradas por mim. Os arquivos resultantes, assim como os originais, estão incluídos na pasta *Resultados* que acompanha este relatório.

## Fontes consultadas

- Eliminação Gauss-Jordan: [https://en.wikipedia.org/wiki/Gaussian\\_elimination](https://en.wikipedia.org/wiki/Gaussian_elimination)
- Interpolação bilinear: [https://en.wikipedia.org/wiki/Bilinear\\_interpolation](https://en.wikipedia.org/wiki/Bilinear_interpolation)