

## Trabalho de implementação 1

**Instruções:** este trabalho deve ser realizado preferencialmente em duplas. Os alunos têm liberdade para escolha da linguagem na qual irá implementar os exercícios propostos, mas é dada preferência à linguagem Python. Além do código implementado, a dupla deve redigir um breve relatório (2-3 páginas) explicando as implementações e testes. Você pode adicionar imagens em um anexo, extrapolando o número de páginas sugerido. Até o dia **20/01/2023**, cada dupla deverá fazer o *upload* de um arquivo .zip contendo código + relatório + quaisquer imagens de exemplo adicionais usadas no Moodle da disciplina.

**Motivação:** Embora a grande maioria das fotografias atualmente já sejam capturadas em formato digital, algumas pessoas possuem fotos impressas. Uma maneira simples de “digitalizá-las” é tirar uma fotografia digital da foto impressa usando, por exemplo, um celular. Por causa do posicionamento da câmera, a foto impressa – originalmente retangular – possivelmente terá um formato de quadrilátero mais genérico. O objetivo deste trabalho é manipular digitalmente a versão digital para que ela fique mais fiel à versão original impressa.

**O trabalho proposto:** assuma que a foto a ser escaneada tem um domínio retangular com dimensões conhecidas (e.g., foto  $10\text{cm} \times 15\text{cm}$ ). A foto impressa é colocada sobre uma superfície planar de cor distinta da foto e capturada digitalmente. O objetivo deste trabalho é achar a homografia planar dada pela matriz  $H_{3\times 3}$  que “retifique” a região da captura digital correspondente à fotografia original impressa para uma região retangular cujas dimensões correspondam às da fotografia\*, e então faça um recorte (*crop*) da imagem retificada para recuperar apenas a parte correspondente à fotografia.

Para achar a homografia, explore o fato de que deve haver correspondências entre as 4 quinas da fotografia na captura original e na retificada. Neste trabalho, você pode informar as coordenadas das quinas manualmente – clicando nos pixels correspondentes –, mas existem vários algoritmos para extração de *corners* na literatura (e.g. pesquise sobre o extrator de corners de Harris) caso a dupla deseje desenvolver uma técnica automática. **Importante:** assuma que o processo de captura das imagens siga o modelo de câmera *pinhole*, e desconsidere eventuais efeitos da distorção radial.

Veja que o mapeamento usado na retificação pode/deve gerar coordenadas não-inteiras de pixel, e há diferentes maneiras de lidar com isso (*warping*). Você pode olhar alternativas na seção 3.6 do livro disponível em <https://szeliski.org/Book/>.

Neste trabalho, as imagens fornecidas foram geradas de acordo com o seguinte protocolo: uma fotografia digital é usada como ponto de partida (Figura 1a), e uma versão impressa é gerada. A versão impressa é colocada sobre uma superfície, e uma nova fotografia digital é tirada com um certo ângulo de captura (Figura 1b). Um possível resultado de retificação é ilustrado na Figura 1c, e o resultado final após o recorte da região de interesse é mostrado na Figura 1d.

Para o relatório final, você deve testar sua implementação usando as imagens fornecidas com o enunciado do trabalho, mostrando o resultado de cada uma após a estimativa da homografia, warping e crop. As dimensões físicas das fotografias originais são 150mm de largura vs. 114mm de altura as duas imagens fornecidas. Como teste adicional, você pode gerar algumas imagens com seu celular e avaliar (visualmente) a qualidade da versão retificada.

Também serão fornecidas digitais originais das fotos impressas, que podem ser consideradas como *ground truth* (arquivos cujo nome é terminado em *\_gt*). As capturas são dadas pelos nomes terminados em *\_cap*. Façam uma análise qualitativa dos resultados obtidos através da

\*Na verdade, o importante é manter o *aspect ratio* da região, visto que a resolução final da imagem pode ser definida através da função de *warping*.

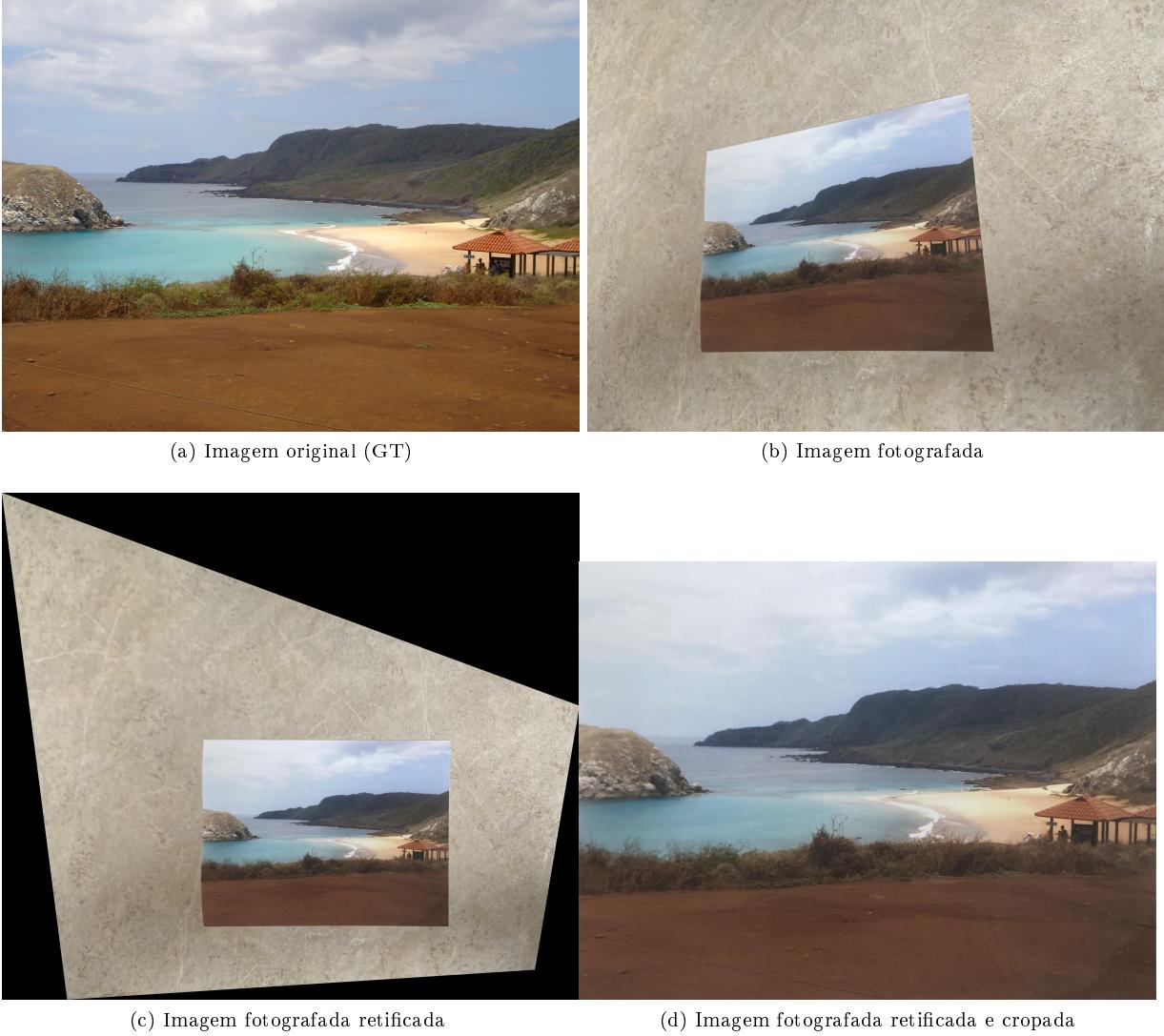


Figura 1: Exemplo de retificação usando homografia planar.

inspeção original da imagem original e da gerada pelo algoritmo desenvolvido. Para uma avaliação quantitativa das técnicas desenvolvidas, use métricas objetivas de qualidade de imagem (como PSNR e SSIM [1]). **Opcionalmente**, vocês podem realizar a comparação usando a métrica perceptual LPIPS [2].

Mais precisamente, convertam cada imagem para tons de cinza e depois realizem a comparação com as duas métricas, assegurando que a resolução das duas imagens comparadas seja a mesma. As imagens estão disponíveis em <https://www.inf.ufrgs.br/~crjung/fvc/trabalho1/>.

## Referências

- [1] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [2] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *CVPR*, 2018.