

Universidade de Brasília  
PPGI

## Relatório de Distorção de Perspectiva

30 de Outubro de 2023

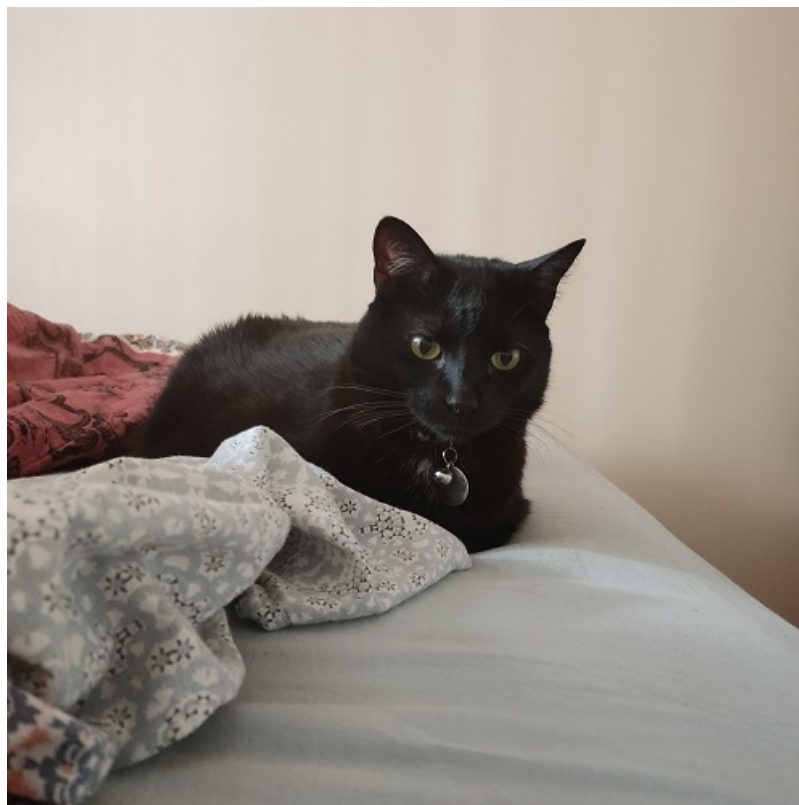
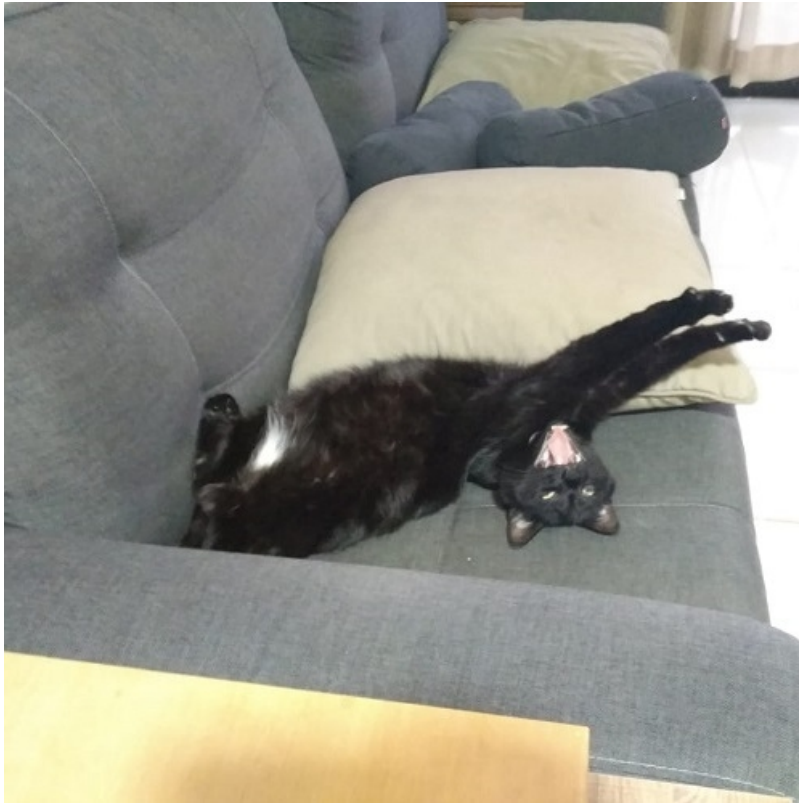
Disciplina: Processamento de Imagens  
Professor: Ricardo Lopes de Queiroz  
Aluno: Henrique Tibério Brandão Vieira Augusto (221101092)

## Metodologia

Escolher imagens para realizar os seguintes experimentos:

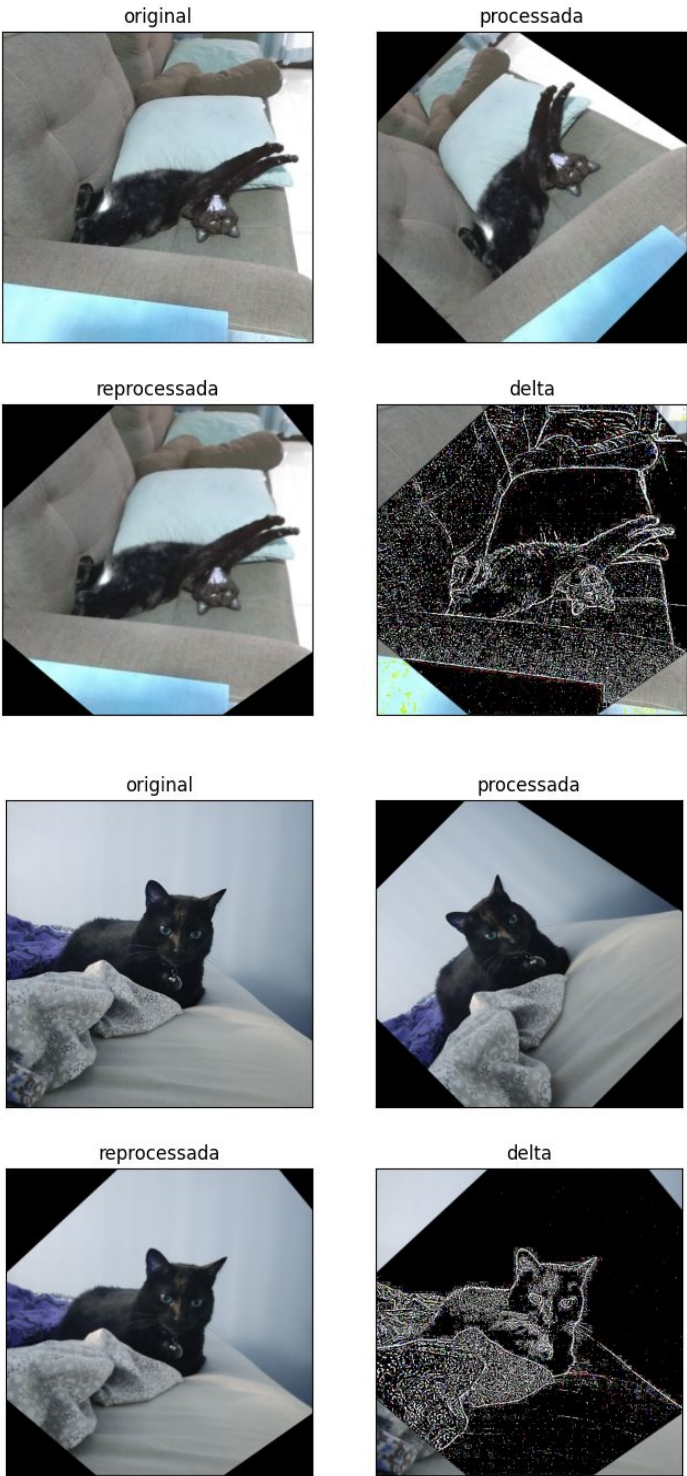
- Escolher um par de imagens coloridas (entre 500 e 700 pixels de largura ou altura);
- Processar cada imagem aplicando uma distorção de perspectiva de forma que a base da imagem fique 20% menor que o topo, seguido de uma rotação de 45 graus do resultado, formando uma nova imagem  $y$ ;
- Inverter os passos para recuperar a imagem  $x$ ;
- Comparar  $x$  e  $x'$ , exibindo a imagem de erro  $e(x) = x - x' + 1$ .

Foram utilizadas as seguintes imagens com altura e largura de 500 pixels:



**Resultados**

Foram obtidas as imagens abaixo:



## **Discussão dos Resultados**

A medida que processamos a imagem, perdemos informação e detalhes, uma vez que na contração que ocorre durante a redução da base da imagem, há uma substituição de pixels. Ao reprocessar a imagem e tentar restaurar a imagem original percebemos a correção da orientação reforçada pela posição e orientação de um elemento visual (o gato deitado). A imagem reprocessada e a original parecem idênticas quando analisadas a olho nu. Entretanto, ao calcular o erro  $e(x)$  percebemos pixels claros, o que significa que uma diferença de valores, portanto, uma perda real de informação que antes parecia inexistente.

## **Conclusão**

Com base nos resultados deste experimento, podemos concluir que as operações aplicadas à imagem, embora aparentem gerar uma imagem idêntica, ocasionam em perda de informação. Não tendo sido essa perda perceptível na primeira impressão, podemos supor que seria possível representar uma mesma imagem utilizando menos valores, uma vez que apenas ao calcular o erro  $e(x)$  é que conseguimos perceber essa diferença.

## Anexo

Detalhes da implementação e resultados estão disponíveis no repositório <https://github.com/htbrandao/ppgi-proc-img/tree/master/trab03>.

- Código em Python para processamento das imagens:

```
import cv2
import numpy as np

from PIL import Image

def processa(imagem, proporcao=0.8, angulo=45):
    altura, largura = imagem.shape[0], imagem.shape[1],
    top_left = [0, 0]
    top_right = [largura, 0]
    bottom_right = [largura, altura]
    bottom_left = [0, altura]
    nova_largura_base = largura * proporcao
    pts1 = np.float32([top_left, top_right, bottom_right, bottom_left])
    pts2 = np.float32([top_left, [nova_largura_base, 0], bottom_right, [0, altura]])
    matriz_perspectiva = cv2.getPerspectiveTransform(pts1, pts2)
    imagem_distorcida = cv2.warpPerspective(imagem, matriz_perspectiva, (largura, altura))
    matriz_rotacao = cv2.getRotationMatrix2D((largura / 2, altura / 2), angulo, 1)
    imagem_rotacionada = cv2.warpAffine(imagem_distorcida, matriz_rotacao, (largura, altura))
    return imagem_rotacionada

def desprocessa(imagem, proporcao=1.25, angulo=-45):
    altura, largura = imagem.shape[0], imagem.shape[1],
    matriz_rotacao = cv2.getRotationMatrix2D((largura / 2, altura / 2), angulo, 1)
    imagem_rotacionada = cv2.warpAffine(imagem, matriz_rotacao, (largura, altura))
    top_left = [0, 0]
    top_right = [largura, 0]
    bottom_right = [largura, altura]
    bottom_left = [0, altura]
    nova_largura_base = largura * proporcao
    pts1 = np.float32([top_left, top_right, bottom_right, bottom_left])
    pts2 = np.float32([top_left, [nova_largura_base, 0], bottom_right, [0, altura]])
    matriz_perspectiva = cv2.getPerspectiveTransform(pts1, pts2)
    imagem_distorcida = cv2.warpPerspective(imagem_rotacionada, matriz_perspectiva, (largura,
altura))
    return imagem_distorcida
```