

## Relatório Octave Tarefa 3

Nome: Thiago Heron Albano de Ávila

### 1. Introdução

Implemente uma função para implementar a correção gama (transformação de potência), denominada ***function corrigida = corr\_gama(imagem, c, gama)***. Para esse problema, utilizei esses exemplos de entradas:

- `corr_gama('lena.bmp', 1, 5);`
- `corr_gama('lena.bmp', 1, 0.5);`
- `corr_gama('lena.bmp', 1.5, 0.5);`

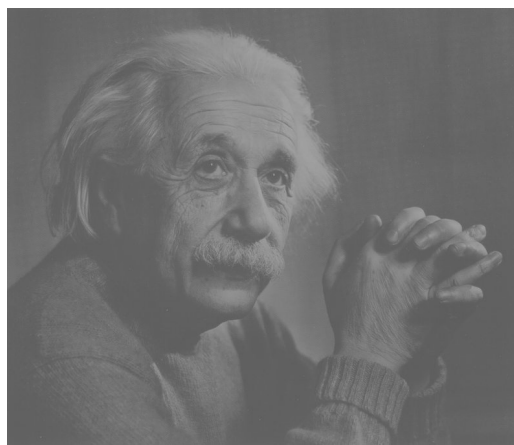
Implemente uma função para implementar a transformação geométrica denominada cisalhamento, denominada ***function cisalhada = cisalhamento(imagem, cv, ch)***. Para esse problema, utilizei esses exemplos de entradas:

- `cisalhamento('lena.bmp', 0.2, 0);`
- `cisalhamento('lena.bmp', 0, 0.2);`
- `cisalhamento('lena.bmp', 0.2, 0.2);`

### 1.2 Imagens Utilizadas



"Lena\_cinza.bmp"



"Fig0241\_conv.png"

## 2. Executando o Código

Foi criado um arquivo **main.m**, que tem como objetivo:

- Carregar uma Imagem
- Seleciona o Tipo da Operação
  - 1 - Transformação de Potência - Correção Gama
  - 2 - Transformação Geométrica - Cisalhamento Vertical e/ou Horizontal
- Exibe a Imagem Resultante
- Salvar a Imagem em “images/output/image\_resized.png”
- Retorna o nome da imagem

## 3. Correção Gama - Transformação de Potência

A implementação da correção gama (transformação de potência) foi utilizado a fórmula  $s = cr^Y$ , vista em aula.

Para repassar essa fórmula, primeiramente precisei calcular o R isolado, para depois elevar a potência da gama, sendo que para isso, necessitei utilizar a função max() para obter o valor resultante. Por fim, realizei um cálculo de ponto a ponto para cada valor.

Notei que, para os valores de gama menores que 1, há poucos valores escuros. Enquanto que, para valores gama maiores que 1, há muitos valores escuros.

```
r = double(imagem)/double((max(max(imagem))));  
imageCorrected = c.*(r.^gama);
```

### 3.1 Resolução dos Exemplos

Resultados dos Testes da Correção Gama (Transformação de Potência) da Tarefa 3.

**A. Imagem Original**

**B. `corr_gama('lena.bmp', 1, 5);`**

**C. `corr_gama('lena.bmp', 1, 0.5);`**

**D. `corr_gama('lena.bmp', 1.5, 0.5);`**



*A) Imagem Original*



*B)  $c = 1 \ \gamma = 5.0$*



*C)  $c = 1 \ \gamma = 0.5$*



*D)  $c = 1.5 \ \gamma = 0.5$*

#### 4. Transformação Geométrica - Cisalhamento Vertical e Horizontal

A implementação do Cisalhamento Vertical e Horizontal (transformação geométrica) foi aplicado de **forma direta**, ou seja, percorrido as coordenadas da imagem original (v, w), gerando as coordenadas (x, y) da imagem transformada, assim mapeamento as novas posições de (x,y) da imagem transformada conforme os parâmetros cv (cisalhamento vertical) e ch (cisalhamento horizontal)

No Cisalhamento Vertical, utilizando o resultado da Transformação Afim com as posições da imagem, para cada pixel da matriz original utilizado:

- $x = v + cv * w;$
- $y = w;$

Assim, para cada nova posição, era verificado se a posição estava dentro do tamanho dos Eixos X e Y, caso essa condição fosse válida, o valor era repassado, caso contrário, o valor não era repassado, o que resultaria em um pixel preto.

```
# CV
for w =1:sizeImageAxisW;
    for v =1:sizeImageAxisV;
        x = floor(v + cv * w);
        y = w;

        if (x <= sizeImageAxisV && y <= sizeImageAxisW)
            imageCorrected(y, x) = imagem(w, v);
        endif
    endfor
endfor
```

Enquanto isso, o mesmo código foi feito para o Cisalhamento Horizontal, porém utilizando o cálculo das novas posições da seguinte forma:

- $x = v$ ;
- $y = ch * v + w$

```
# CH
for w =1:sizeImageAxisW;
    for v =1:sizeImageAxisV;
        x = v;
        y = floor(ch * v + w);
        if (x <= sizeImageAxisV && y <= sizeImageAxisW)
            imageCorrected(y, x) = imagem(w, v);
        endif
    endfor
endfor
```

Também há um terceiro caso quando deseja-se aplicar tanto Cisalhamento Vertical quanto Horizontal

```
for w =1:sizeImageAxisW;
    for v =1:sizeImageAxisV;
        x = floor(v + cv * w);
        y = floor(ch * v + w);
        if (x <= sizeImageAxisV && y <= sizeImageAxisW)
            imageCorrected(y, x) = imagem(w, v);
        endif
    endfor
endfor
```

#### 4.1 Resolução dos Exemplos

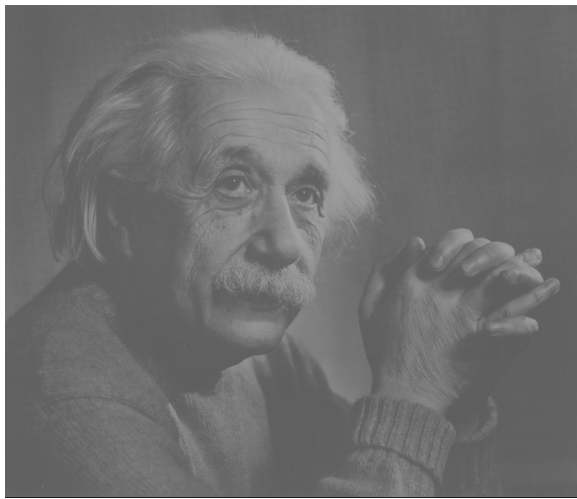
Resultados dos Testes de Cisalhamento Vertical e Horizontal (Transformação Geométrica) da Tarefa 3.

A. Imagem Original

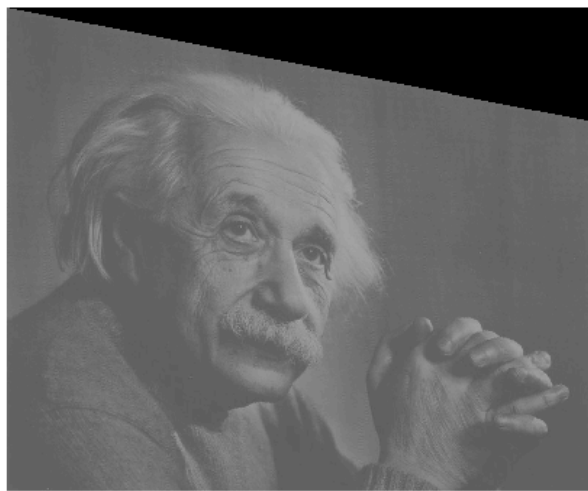
B. `corr_gama('images/input/Fig0241_conv.png' , 0.2, 0);`

C. `corr_gama('images/input/Fig0241_conv.png' , 0, 0.2);`

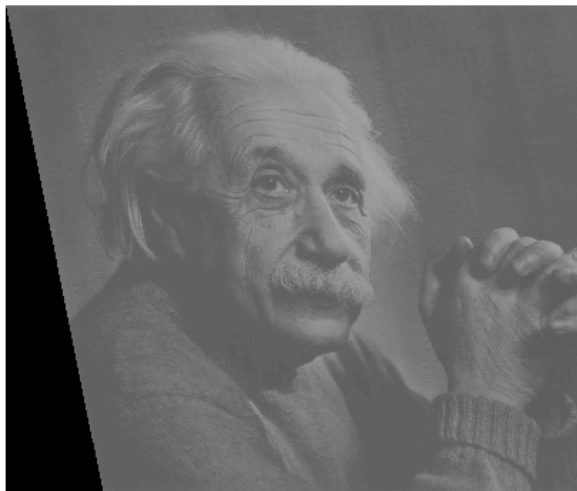
D. `corr_gama('images/input/Fig0241_conv.png' , 0.2, 0.2);`



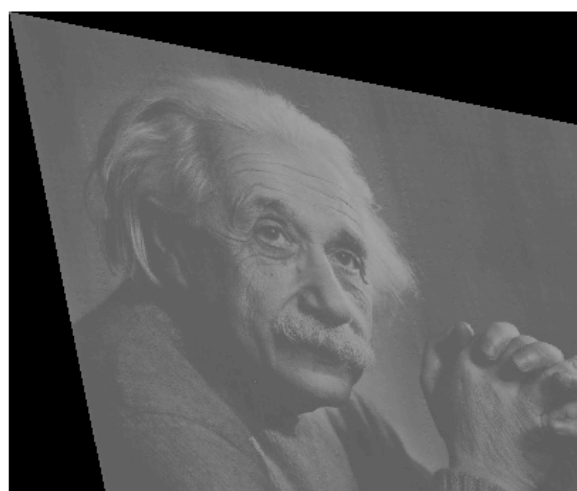
A) *Imagem Original*



B)  $cv = 0.2 \quad ch = 0$



C)  $cv = 0 \quad ch = 0.2$



D)  $cv = 0.2 \quad ch = 0.2$