

Tarefa 1

- **Discente:** Álvaro Ferreira Pires de Paiva (2016039162, alvarofepipa@gmail.com)
- **Disciplina:** DIM0141 - Visão Computacional
- **Professora:** Rafael Beserra Gomes

Respostas

1. Descreva a distribuição de células fotorreceptoras na retina e seu impacto na percepção visual

Na retina existem dois tipos de fotorreceptores: bastonetes e cones. Ambos possuem seus nomes devido aos seus formatos.

Os cones concentram-se mais na fóvea e são cerca de 6-7 milhões no olho humano, nos permitem enxergar na presença de luz e cores. Os bastonetes encontram-se mais na periférica da retina e são cerca de 75-100 milhões no olho humano, nos permitem enxergar a presença de luz em ambientes mais escuros, ou seja, são muito mais sensíveis a luz do que os cones, porém sua percepção monocromática.

2. Qual a diferença entre os cones S, M e L? Quais deles são mais ativados quando uma luz amarela é projetada na retina?

Os cones S, M e L correspondem a cones que percebem diferentes comprimentos de ondas:

- Cones S (*Short*): são os que percebem um comprimento de onda curta, de cor azulada. Seu comprimento de onda de pico é de 420-440 nanômetros.
- Cones M (*Medium*): são os que percebem um comprimento de onda média, de cor verde. Seu comprimento de onda de pico é de 534-545 nanômetros.
- Cones L (*Large*): são os que percebem um comprimento de onda mais longa, de cor vermelha. Seu comprimento de onda de pico é de 564-580 nanômetros.

Quando a luz amarela é projetada na retina, os cones de percepção das cores verde (cones M) e vermelho (cones L) são os mais ativados, pois sua percepção de comprimentos de onda são os que chegam perto do comprimento de onda da cor amarela.

3. Escreva um programa para:
 - a. abrir uma imagem e exibir na tela os 3 canais separadamente
 - b. abrir uma imagem e exibir na tela a imagem invertida horizontalmente
 - c. abrir duas imagens (a e b) de mesmo tamanho e exibir na tela uma nova imagem (c)

com o blending entre ambas, usando uma combinação linear entre elas
d. salvar uma nova imagem com o seguinte gradiente vertical

```
# Questão 3.a
import cv2

class OnlyChannel:
    """Classe responsável por realizar a seleção dos canais de cores da imagem.
    """
    CHANNEL_RED = 2 # Valor do canal de cor vermelho
    CHANNEL_GREEN = 1 # Valor do canal de cor verde
    CHANNEL_BLUE = 0 # Valor do canal de cor azul

    @staticmethod
    def to_red(img):
        """Seta como zero os canais azul e verde, restando apenas o canal vermelho da imagem."""
        img_red = img.copy()
        img_red[:, :, OnlyChannel.CHANNEL_BLUE] = 0
        img_red[:, :, OnlyChannel.CHANNEL_GREEN] = 0

        return img_red

    @staticmethod
    def to_green(img):
        """Seta como zero os canais azul e vermelho, restando apenas o canal verde da imagem."""
        img_green = img.copy()
        img_green[:, :, OnlyChannel.CHANNEL_BLUE] = 0
        img_green[:, :, OnlyChannel.CHANNEL_RED] = 0

        return img_green

    @staticmethod
    def to_blue(img):
        """Seta como zero os canais verde e vermelho, restando apenas o canal azul da imagem."""
        img_blue = img.copy()
        img_blue[:, :, OnlyChannel.CHANNEL_GREEN] = 0
        img_blue[:, :, OnlyChannel.CHANNEL_RED] = 0

        return img_blue

# Imagem original
filename = 'imagem.jpg'
img = cv2.imread(filename)

# RGB - Red
cv2.imshow('R-Red', OnlyChannel.to_red(img))
# RGB - Blue
cv2.imshow('B-Blue', OnlyChannel.to_blue(img))
# RGB - Green
```

```
cv2.imshow('G-RGB', OnlyChannel.to_green(img))
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

```
# Questão 3.b
```

```
import cv2
```

```
class Flip:
```

```
    """Classe responsável por realizar o processo de flip na imagem."""
```

```
    FLIP_VERTICAL = 0 # Valor do canal de cor vermelho
```

```
    FLIP_HORIZONTAL = 1 # Valor do canal de cor verde
```

```
    FLIP_BOTH = -1 # Valor do canal de cor azul
```

```
    @staticmethod
```

```
    def flip(img, flip_code = FLIP_BOTH):
```

```
        """Realiza o processo de flip na imagem."""
```

```
        return cv2.flip(img, flip_code)
```

```
# Imagem original
```

```
filename = 'imagem.jpg'
```

```
img = cv2.imread(filename)
```

```
# Flip horizontal
```

```
cv2.imshow('Flip-Horizontal', Flip.flip(img, Flip.FLIP_HORIZONTAL))
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

```
# Questão 3.c
```

```
import cv2
```

```
# Imagens originais
```

```
img_a = cv2.imread('resized/php_logo.png')
```

```
img_b = cv2.imread('resized/python_logo.png')
```

```
# Blend
```

```
img_blending = cv2.addWeighted(img_a, 0.5, img_b, 0.5, 0.0)
```

```
# Show
```

```
cv2.imshow('img-blending', img_blending)
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

```
# Questão 3.d
```

4. Considere o formato de imagem NetPBM
- Qual a diferença entre os números mágicos P1, P2, P3, P4, P5 e P6?
 - Converta uma imagem jpg para PBM (ASCII) utilizando convert:

```
convert -compress none imagem.jpg novaimagem.pbm
```

Exiba utilizando display:

```
display novaimagem.pbm
```

- Converta a mesma imagem para PBM (binário) e para PPM (binário). Compare o tamanho dos 4 arquivos de imagem.
- Por que o formato binário ocupa menos espaço que o formato ASCII?
- Por que o formato PPM binário ocupa mais espaço que o formato PBM binário?

4.a

Os números mágicos servem para identificar o tipo do arquivo e sua codificação. Os 3 primeiros números mágicos são de codificação ASCII, os 3 restantes são de codificação binário. P1 e P4 são do formato `.pbm`, P2 e P5 são `.pgm` e P3 e P6 são `.ppm`.

4.b

```
#!/bin/bash
# Converte
convert -compress none bash_logo.jpg bash_logo_pbm_ascii.pbm
# Exibe
display bash_logo_pbm_ascii.pbm
```

4.c

```
#!/bin/bash
# Converte
convert bash_logo.jpg bash_logo_pbm_binary.pbm
convert bash_logo.jpg bash_logo_ppm_binary.ppm
```

Imagem	Tamanho
bash_logo.jpg	43987 B
bash_logo_pbm_ascii.pbm	1297212 B
bash_logo_pbm_binary.pbm	81012 B
bash_logo_ppm_binary.ppm	1944016 B

A imagem original é a mais leve das quatro. O PBM binário saiu mais leve que o PBM ASCII, ocupando apenas aproximadamente 62,45% do tamanho do PBM ASCII. O PPM binário saiu

muito mais grande que o PBM binário, quase 2400 vezes maior.

4.d

O formato ASCII é legível ao olho humano, utiliza-se a tabela ASCII para codificar os pixels e gasta muito espaço de memória. O formato binário gasta apenas 1 bit por pixel, o que é uma economia muito grande se comparado ao formato ASCII.

4.e

O formato PBM binário gasta apenas 1 bit por pixel, enquanto o formato PPM gasta 24 bits por pixel (8 para cada camada de cor).

5. Quais desses formatos são vetoriais e quais são bitmaps? BMP, SVG, JPG, EPS, PNG

- Vetoriais: SVG e EPS.
- Bitmaps: BMP, JPG e PNG.

6. Imagens de algumas aplicações possuem um nível de ruído considerável, principalmente aquelas que captam em níveis baixos de iluminação, como na captura de imagens astronômicas. Uma das formas de atenuar esse tipo de ruído é através da média de inúmeras imagens. Utilizando as 9 imagens disponibilizadas, crie um programa que gere uma nova imagem com o ruído atenuado.