

Trabalho prático de Processamento digital de imagens

Yvson Nunes Figueiredo

Relatório elaborado para o Trabalho prático da disciplina de processamento digital de imagens,
ministrada pelo Professor Leonardo Vidal Batista do Centro de
Informática da Universidade Federal da Paraíba.

Este projeto teve como objetivo principal aplicar os conhecimentos adquiridos no curso de Processamento Digital de Imagens. Desenvolvi funções para converter entre os sistemas de cores RGB e HSB, explorando diversas abordagens, como a manipulação da matiz (H), saturação e transformações negativas. Além disso, testei a eficácia de filtros, como Filtros Box e Sobel, com diferentes configurações. Essas experimentações aprofundaram minha compreensão das técnicas de processamento de imagens e suas aplicações práticas.

Palavras-chave: RGB, HSB, Filtros, Box e Sobel.

1 Introdução

O objetivo central deste projeto foi a aplicação prática dos conhecimentos adquiridos durante o curso de Processamento Digital de Imagens. Ao longo do desenvolvimento do trabalho, foram elaboradas funções destinadas à conversão entre os sistemas de cores RGB e HSB, bem como à transição reversa de HSB para RGB. Além disso, foram conduzidas diversas experimentações com o intuito de explorar as conversões, assim, as seguintes operações foram feitas: manipulação da matiz (H) e saturação, seguida da conversão de volta para o espaço de cores RGB e também transformações negativas canal por canal no espaço RGB e na banda B do espaço de cores HSB, seguidas novamente pela conversão para o espaço RGB.

Além disso, foi implementado também a operação de correlação por meio de testes com alguns filtros, a saber: Filtros Box e Sobel. Na aplicação do filtro Box, foram feitas duas abordagens distintas. A primeira envolveu a aplicação de um filtro quadrado de tamanho 15x15. A segunda abordagem aplicou dois filtros em sequência: um 1x15 seguido por um 15x1. Desse modo, testes de desempenho foram realizados para determinar qual dessas abordagens se mostrou mais eficaz.

Por fim, o filtro Sobel foi aplicado individualmente a cada canal da imagem e, para fins de visualização, aplicado o valor absoluto seguido de uma expansão de histograma, ajustando os valores para o intervalo de 0 a 255.

2 Metodologia

A linguagem de programação escolhida para a implementação dos algoritmos foi o python, de modo que a biblioteca OpenCV foi utilizada exclusivamente para a leitura da imagem que é convertida para um Tensor numpy de dimensão <width x height x 3> em que 3 é o número de canais. Ademais, matplotlib foi utilizado para a exibição da imagem e a biblioteca joblib para a paralelização de alguns processos.

2.1 Conversão entre os espaços de cores

Para a realização da conversão entre o espaço RGB (Red, Blue, Green) foi levado em consideração que os valores numéricos em cada canal estariam em um intervalo de 0 à 255 pela razão de que as imagens possuem 256 níveis de quantização (2^8) advindas da utilização de 8 bits por canal.

De posse dessa premissa, as conversões entre o espaço RGB e HSB (Hue, Saturation, Brightness) e vice-versa se deram utilizando os métodos descritos por [1].

$$H = \begin{cases} 0 & \text{se } MAX = MIN \\ 60 \cdot \left(\frac{G-B}{MAX-MIN} \right) + 0 & \text{se } R = MAX, G \geq B \\ 60 \cdot \left(\frac{G-B}{MAX-MIN} \right) + 360 & \text{se } R = MAX, G < B \\ 60 \cdot \left(\frac{B-R}{MAX-MIN} \right) + 120 & \text{se } G = MAX \\ 60 \cdot \left(\frac{R-G}{MAX-MIN} \right) + 240 & \text{se } B = MAX \end{cases}$$
$$S = \begin{cases} 0 & \text{se } MAX(R, G, B) = 0 \\ 1 - \frac{MIN(R, G, B)}{MAX(R, G, B)} & \text{se } MAX(R, G, B) \neq 0 \end{cases}$$
$$B = MAX(R, G, B)$$

$$H_i = \left\lfloor \frac{H}{60} \right\rfloor \mod 6$$

$$f = \frac{H}{60} - H_i$$

$$p = V \cdot (1 - S)$$

$$q = V \cdot (1 - f \cdot S)$$

$$t = V \cdot (1 - (1 - f) \cdot S)$$

$$\begin{cases} \text{se } H_i = 0 : R = V, G = t, B = p \\ \text{se } H_i = 1 : R = q, G = V, B = p \\ \text{se } H_i = 2 : R = p, G = V, B = t \\ \text{se } H_i = 3 : R = p, G = q, B = V \\ \text{se } H_i = 4 : R = t, G = p, B = V \\ \text{se } H_i = 5 : R = V, G = p, B = q \end{cases}$$

2.2 Alteração das propriedades dos espaços de cores

Para a alteração da Matiz e Saturação no espaço HSB é passado como parâmetro um ângulo $[0, 360]$ e um fator multiplicativo que indica o quanto em termos percentuais a Saturação deve ser alterada. Ademais, para o efeito negativo, a função: $f(r) = L - r$ é utilizada nas duas abordagens. Na primeira abordagem que é feita no espaço RGB, a função é aplicada a cada canal com $L = 255$ de modo que se é mapeado cada cor para seu equivalente negativo. Na segunda abordagem, o mapeamento ocorre apenas no canal B do sistema HSB $[0, 1]$ em que o valor de L é 1.

2.3 Aplicação dos filtros Box

Para a aplicação dos filtros, a operação de correlação é empregada de modo que o filtro é carregado de um arquivo txt e então aplicado dentro do programa à imagem. Duas abordagens são utilizadas para o teste: a aplicação de um filtro 15×15 e a aplicação de um filtro 1×15 , seguido de um 15×1 . Em tese, como a dimensão das imagens resultantes são iguais, é esperado que as duas técnicas resultem no mesmo efeito de suavização. Além disso, a carga computacional é esperada que seja menor para a segunda abordagem.

2.4 Aplicação do filtro de Sobel

Para o efeito do filtro de sobel, novamente se é carregado dos arquivos txts as máscaras horizontal e vertical. Assim, a operação de correlação é aplicada separadamente nos canais com os filtros verticais e horizontais, produzindo a detecção de bordas para o R, G e B isoladamente. Posteriormente as imagens são mescladas e para a visualização a expansão de histograma é empregada.

3 Detalhes de implementação

3.1 Mudança de sistemas

Cada canal na imagem foi tratado de modo individual, de forma que uma rotina de mudança de sistema era aplicada pixel a pixel, resultando no fim em uma imagem no sistema de cores desejado.

3.2 Correlação

A função de correlação foi implementada recebendo como parâmetros: a imagem a ser transformada, o Kernel ou máscara

a ser aplicada e o stride de deslocamento do filtro. Para a implementação efetiva da operação, levando em consideração o stride, o método descrito em [2] foi utilizado para o cálculo da dimensão final da imagem. Por fim, para essa implementação, foi considerado um stride igual para as dimensões verticais e horizontais.

4 Experimentos

4.1 Alteração de matiz e saturação no HSB, com posterior conversão a RGB

Para o primeiro experimento, foi realizada uma conversão do sistema RGB para o HSB de modo que a matiz e saturação fossem alteradas e a imagem fosse posteriormente convertida para o RGB para a visualização.

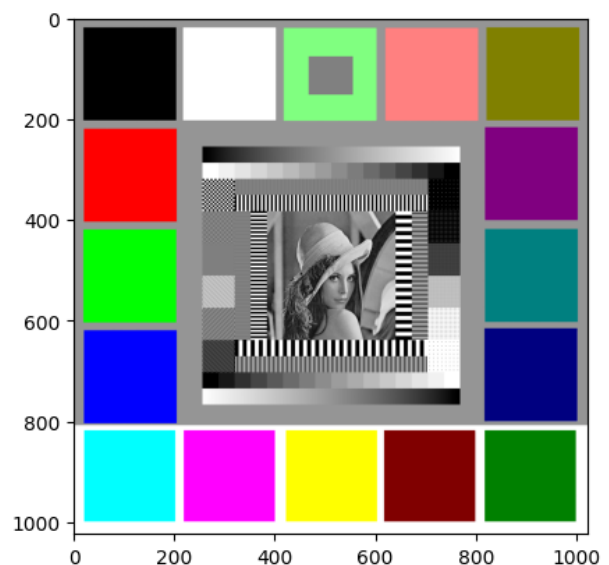


Figura 1: imagem de referência 1.

4.1.1 Preservação da saturação e modificação da matiz por uma soma angular de 85 graus

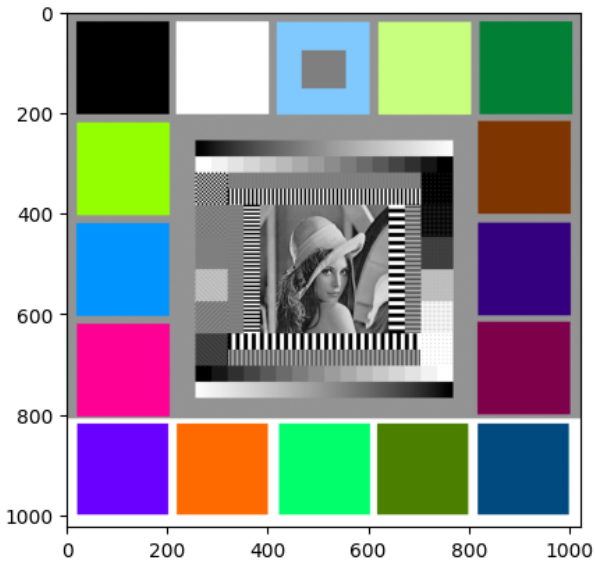


Figura 2: Ângulo 85° saturação 1.

4.1.3 Saturação em 35% e modificação da matiz por uma soma angular de 85 graus

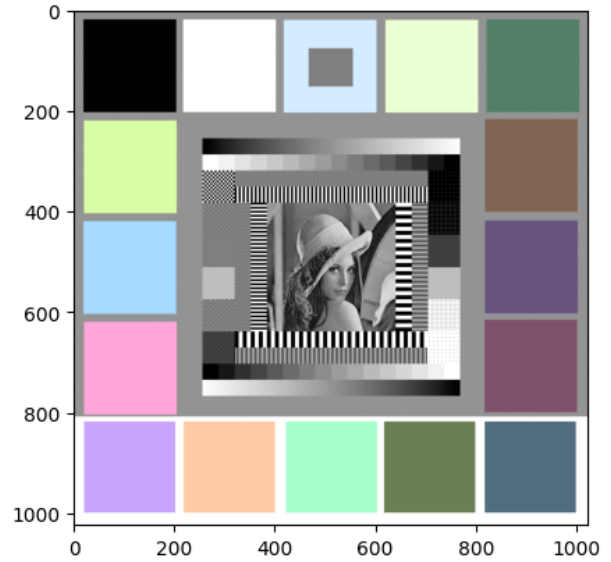


Figura 4: Ângulo 85° saturação 0.35.

4.1.2 Dobro da saturação e modificação da matiz por uma soma angular de 85 graus

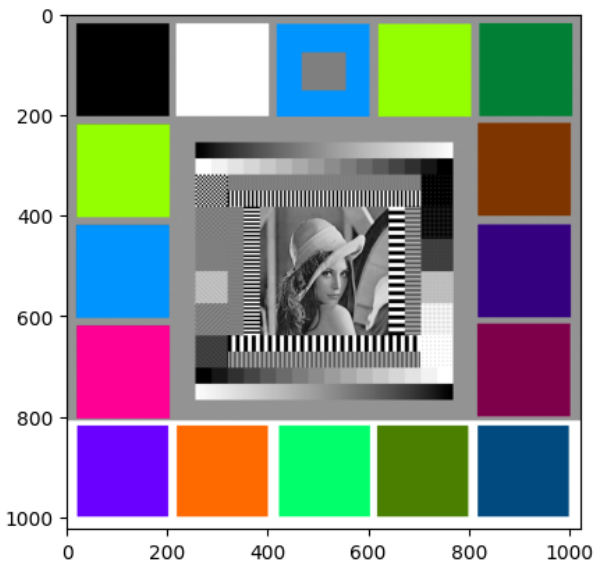


Figura 3: Ângulo 85° saturação 2.

4.2 Negativo

Para o segundo experimento foi feita a aplicação do efeito negativo sobre a imagem utilizando a abordagem no espaço RGB e no espaço HSB, negando o B.

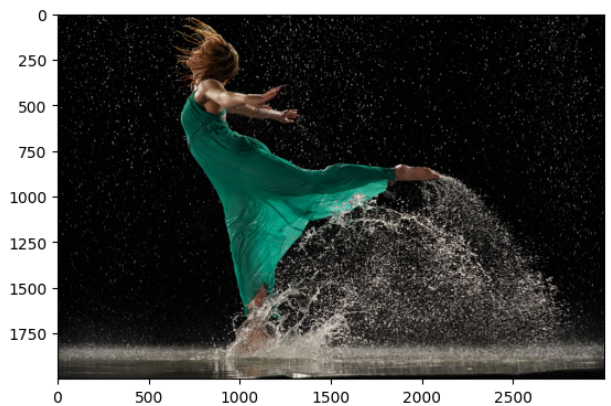


Figura 5: imagem de referência 2

4.2.1 Modificação dos canais RGB aplicando efeito negativo canal a canal

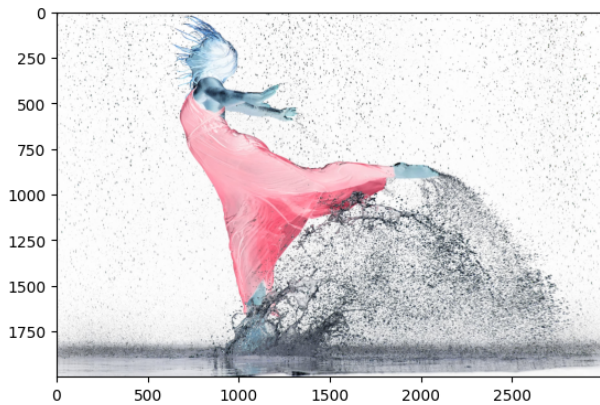


Figura 6: negativo nos canais RGB

4.3.1 Aplicação do filtro box 15x15



Figura 9: Filtro 15x15.

4.2.2 Modificação do canal B no HSB e conversão posterior para RGB

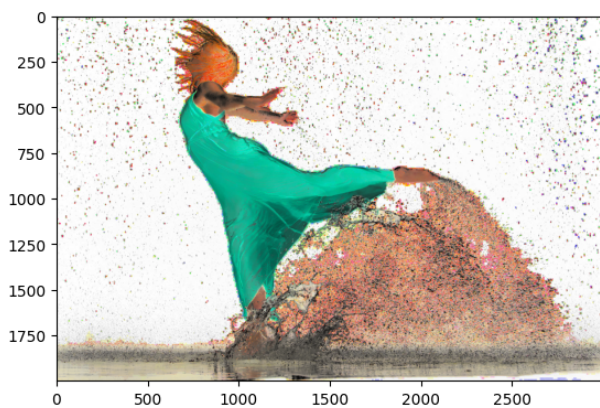


Figura 7: negativo no canal B

Tempo de execução: 7578.928ms.

4.3.2 Aplicação do filtro box 1x15 seguido de um 15x1



Figura 10: Filtro 1x15 e 15x1.

4.3 Filtro Box

Para o terceiro experimento foi aplicado um filtro média 15x15 e dois filtros 1d sobre a imagem e medidos os tempos de execução dos processos.

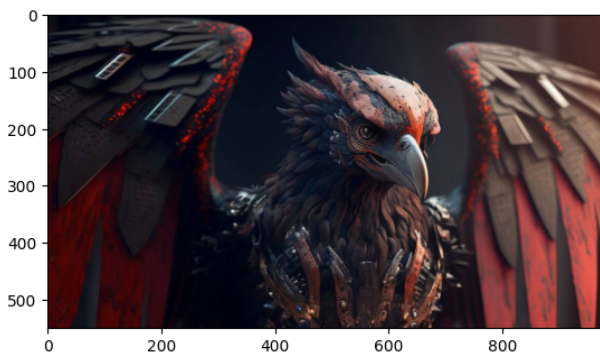


Figura 8: imagem de referência 3

Tempo de execução: 14847.475ms

4.4 Filtro de Sobel

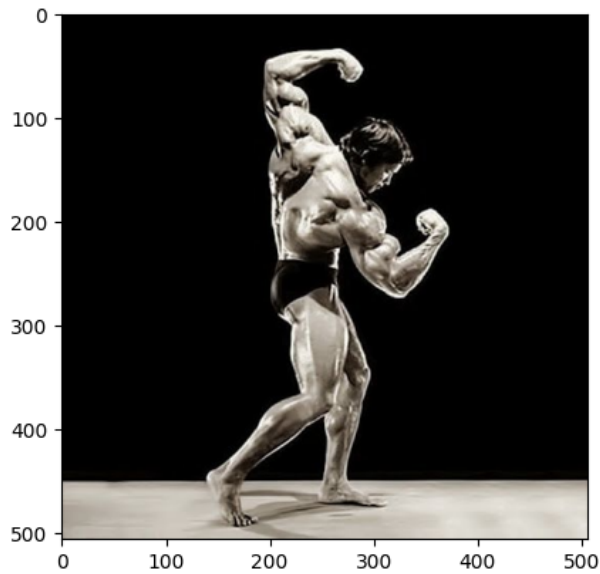


Figura 11: imagem de referência 4

4.4.1 Filtro de Sobel horizontal com expansão de histograma

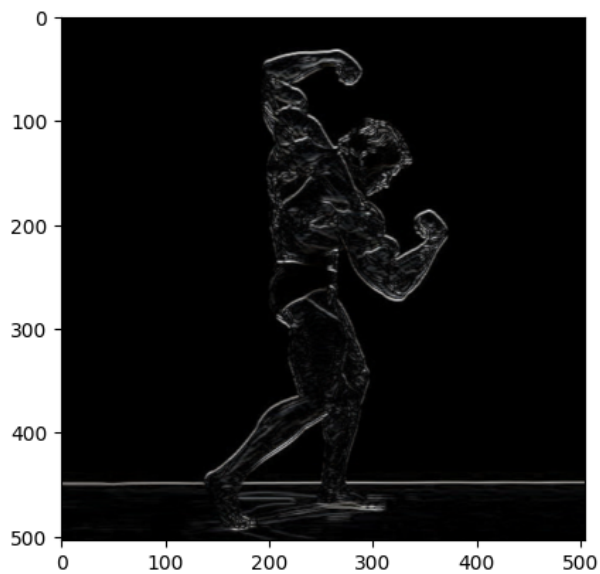


Figura 12: sobel horizontal

4.4.2 Filtro de Sobel vertical com expansão de histograma

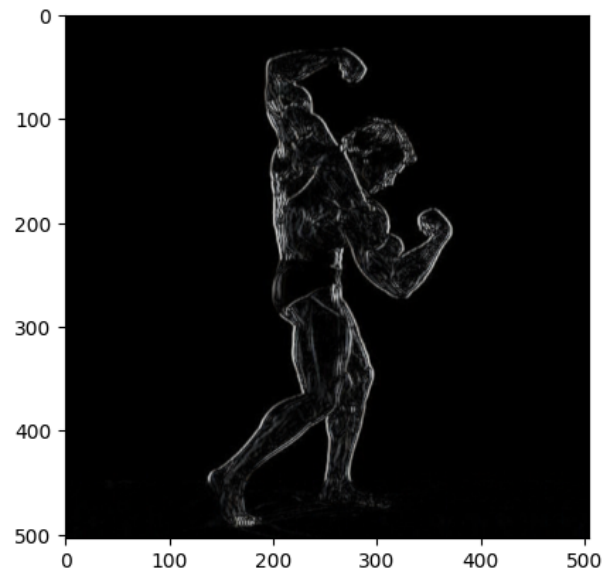


Figura 13: sobel vertical

4.4.3 Filtro de Sobel completo com expansão de histograma

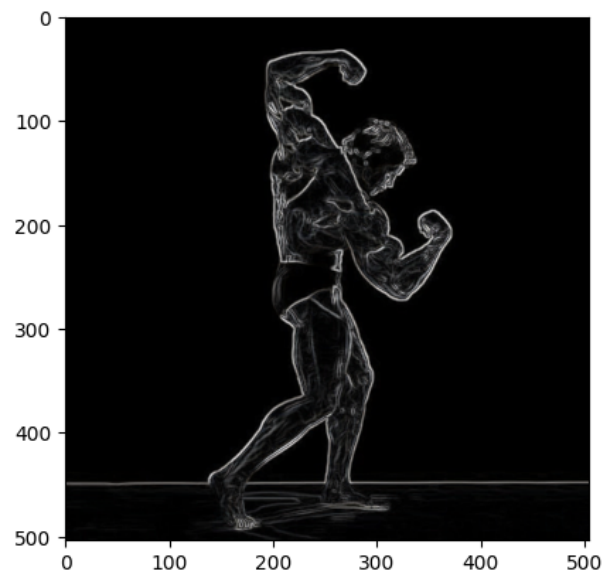


Figura 14: sobel completo

5 Discussão

5.1 Transformação no espaço de cores

Tomando como base a Figura 1, foram realizadas transformações graduais na matiz e saturação da imagem. Na Figura 2, a matiz foi alterada, adicionando um acréscimo de 85° a cada pixel, criando uma nova paleta cromática. Em seguida, na Figura 3, a saturação foi intensificada, proporcionando cores mais vibrantes e vivas. Por fim, partindo da Figura 2 como referência, a Figura 4 foi desaturada, incorporando mais tons

de cinza e conferindo um aspecto mais suave e desbotado às cores. Além disso, quando se trata das transformações negativas, a partir da imagem 5, a Figura 6 apresenta uma transformação de negativo global, na qual o efeito negativo é aplicado a todos os canais de cor RGB, resultando em uma imagem com uma aparência totalmente invertida. Por outro lado, na Figura 7, a negação é aplicada apenas ao canal de brilho (B) do espaço de cores HSB, criando um efeito negativo no brilho em relação à imagem original.

5.2 Filtros Box

Com base na imagem de referência 8, observa-se que, do ponto de vista visual, as duas técnicas de suavização produziram resultados similares. Notavelmente, a aplicação do filtro de média 15x15 resultou em um efeito equivalente à aplicação sequencial de um filtro 1x15 seguido por um 15x1. Entretanto, em contraste à premissa teórica, a implementação prática dessas abordagens neste trabalho revelou um desempenho inferior para a segunda alternativa. A realização do filtro 1x15 seguido por um 15x1 acarretou em um tempo de processamento dobrado em relação ao filtro 15x15, mesmo com algumas otimizações empregadas, incluindo o uso de bibliotecas como o NumPy e técnicas de processamento paralelo.

5.3 Filtro de Sobel

Finalmente, a experimentação do uso do filtro de Sobel foi conduzida em três fases, tendo a imagem 11 como ponto de partida. Inicialmente, aplicou-se o filtro horizontal de forma separada a cada canal de cor, capturando assim as bordas que continham componentes horizontais. Posteriormente, essas informações foram unificadas na imagem 12 que evidenciou de maneira clara as contribuições individuais de cada canal para a componente horizontal.

Analogamente, o mesmo foi feito para o filtro vertical. É importante destacar a particularidade desse filtro, que resultou na perda da definição da fronteira entre o piso e a parede, devido à sua orientação vertical, enquanto realçava outras características relevantes.

Como resultado final, as componentes horizontais e verticais são somadas resultando na imagem 14 que captura todas as bordas da imagem original.

robusta para a detecção de bordas, apresentando amplas aplicações no campo da visão computacional. Essas experimentações reforçam a riqueza e a versatilidade das técnicas estudadas neste trabalho.

Referências

- [1] G. Leparmentier, "Manipulating colors in .NET - Part 1," 2007. endereço: <https://www.codeproject.com/Articles/19045/Manipulating-colors-in-NET-Part-1>.
- [2] Z. Lipton, "Dive into Deep Learning - Padding and Stride," 2023. endereço: https://d2l.ai/chapter_convolutional-neural-networks/padding-and-strides.html.

6 Conclusão

Destarte, foram exploradas diversas aplicações dos conceitos desenvolvidos neste trabalho. No âmbito dos espaços de cores, tornou-se evidente como diferentes abordagens para representar o mesmo conceito espacial podem ser realizadas: seja através de modelagens matemáticas e combinatoriais, como no espaço RGB, ou de maneira mais visual e alinhada à percepção humana, como no espaço HSB. Ficou claro que esses sistemas são intercambiáveis, oferecendo flexibilidade no tratamento de cores. Além disso, destacou-se a aplicação do filtro da média e seu papel na suavização de imagens, evidenciando sua utilidade no contexto do processamento de imagens. Por fim, o filtro de Sobel revelou-se uma ferramenta