

Atividade Avaliativa Processamento de Imagens

Arthur Brenno, Caio Cesario

November 11, 2023

1 Introdução

Manipular os pixels da imagem de maneira correta é essencial para garantir uma manipulação efetiva em uma imagem RGB. Sob essa ótica, foi proposta uma atividade a fim de avaliar o aluno quanto a sua capacidade de entendimento acerca das manipulações de brilho nesse tipo de imagem.

2 Integridade na Redução de Brilho

A partir da atividade proposta, questionou-se como garantir que a operação de redução de brilho tenha sua integridade mantida. Para isso, devemos analisar que, na situação de aumento de brilho, devemos limitar o aumento de modo que não ultrapasse o valor 255. Isso garante que a intensidade esteja dentro do valor adequado e não ultrapasse o valor máximo dos componentes de uma imagem RGB. Para analisarmos a redução de brilho, devemos analisar, primeramente, que o intervalo de intensidade dos pixels de uma imagem RGB será entre o intervalo 0 (incluso) e 255 (incluso) para cada canal da imagem, podendo ser melhor expresso a partir da seguinte notação: $[0, 256)$

3 Integridade na Redução de Brilho

```
import ij.ImagePlus;
import ij.plugin.filter.PlugInFilter;
import ij.process.ColorProcessor;
import ij.process.ImageProcessor;
import static java.lang.Math.max;
import static java.lang.Math.min;
public class Darken_RGB_2 implements PlugInFilter {

    /*=====*/

    static final int R = 0;
    static final int G = 1;
    static final int B = 2;

    /*=====*/

    static final int DECREMENTO = 100;

    /*=====*/

    public int setup(String arg, ImagePlus imp) {
        return DOES_RGB;
    }

    public void run(ImageProcessor ip) {
        ColorProcessor cp = (ColorProcessor) ip;
        int[] pixel = new int[3];
```

```

final int totalLinhas = cp.getHeight();
final int totalColunas = cp.getWidth();
final int produtoEixos = totalLinhas * totalColunas;

for (int i = 0; i < produtoEixos; ++i) {
    int u = i % totalColunas;
    int v = i / totalColunas;

    cp.getPixel(u, v, pixel);
    pixel[R] = min(max(f(pixel[R]), 0), 255);
    pixel[G] = min(max(f(pixel[G]), 0), 255);
    pixel[B] = min(max(f(pixel[B]), 0), 255);
    cp.putPixel(u, v, pixel);
}

public int f(int x) {
    return x - DECREMENTO;
}

```

4 Imagens de exemplo

As imagens a seguir foram utilizadas para medição da qualidade do nosso algoritmo:



Figure 1: Imagem sem a aplicação do algoritmo



Figure 2: Imagem com a aplicação do algoritmo

Nota-se que, de maneira eficiente, a operação de redução de brilho ocorreu como esperado. E o algoritmo funcionou perfeitamente.

5 Conclusão

A qualidade do algoritmo foi avaliada em um grupo de imagens de teste, destacando o desempenho do modo de redução de brilho. As figuras comparativas indicam que esta implementação foi capaz de produzir uma imagem processada corretamente dentro das limitações de intensidade especificadas.

Portanto, a atividade proporcionou uma compreensão mais aprofundada dos processos de ajustes de brilho em imagens RGB, além de implementar com sucesso o sistema de redução de brilho. O código foi projetado como um meio de garantir a confiabilidade da imagem em relação ao assunto mencionado acima e à disciplina dos estudantes sobre esse tema.