

Dokumentace úlohy číslo 77 Konstrukce histogramu pro rastr v rámci zkoušky z předmětu PřfUK Úvod do programování (ZS 2023/24)

Konstrukce histogramu pro rastr

Pro vstupní rastr ve formátu JPG, PNG, GIF vytvořte histogram znázorňující absolutní četnost zastoupení jednotlivých barev, a to po jejich R, G, B složkách. Barevnou informaci v každém pixelu dekomponujte na RGB hodnoty v intervalu $<0,255>$. Pro vykreslení histogramu použijte Vámi zvolenou grafickou knihovnu.

Cílem tohoto kódu je vytvořit histogram základních barev rastrového obrázku. Histogram je grafická reprezentace, která ukazuje distribuci intenzity barev v obrázku. Každý kanál barvy (červená, zelená a modrá) je zpracován samostatně.

Existující algoritmy

Existují různé metody pro vytvoření histogramu barev. Nejběžnější metoda je [first-fit bin packing](#), rozdělit každý kanál barvy na několik binů (např. 256 pro 8bitové obrázky) a spočítat počet pixelů v každém binu. Mezi další metody patří třeba [Bucket sort](#).

Zvolený algoritmus

Tento kód používá knihovnu [NumPy](#) pro vytvoření histogramu a [Matplotlib](#) pro jeho zobrazení. Pro každý kanál se vytvoří histogram pomocí funkce `histogram` z knihovny NumPy. Funkce `histogram` právě využívá algoritmu *first-fit bin packing*. Poté se histogram zobrazí pomocí funkce `plot` z knihovny matplotlib.

Struktura programu

Nejdříve na začátku kódu importujeme používané knihovny.

```
# Import necessary libraries
import imageio.v3 as iio
import numpy as np
import matplotlib.pyplot as plt
```

Poté je kód strukturován v rámci zásad objektivě orientovaného programování (OOP) do dvou tříd `ImageReader` a `ImageHistogram`.

Třída `ImageReader` slouží k načtení obrázku do Numpy pole a pro zobrazení načteného obrázku.

```
# Define a class to read and display an image
class ImageReader:
    # Define the __init__ method to read the image
    def __init__(self, path):
        self.image = io.imread(path)

    # Define a method to display the image
    def show_image(self):
        plt.imshow(self.image)
        plt.show()
```

Definujeme třídu `ImageHistogram` podle zásad OOP dědí funkce třídy `ImageReader`.

```
# Define a class to create and display a histogram of an image
class ImageHistogram(ImageReader):
    def __init__(self, path):
        # Call the __init__ method of the parent class
        ImageReader.__init__(self, path)
```

Definujeme metodu `get_histogram`, která bude vytvářet a zobrazovat histogram zvoleného obrázku.

Pomocí knihovny `Matplotlib` vytvoříme graf s ohraničený na ose X od 0 do 256. Zároveň definujeme seznam tří barev spektra pro iteraci.

```
# Define a method to display the histogram of the image
def get_histogram(self):
    # Create a new figure and set the x-axis limits
    plt.figure()
    plt.xlim([0, 256])

    # Create a histogram for each color channel
    colors = ('r', 'g', 'b')
```

Pro cyklus procházející jednotlivými barvami spektra vytvoří pro každou barvu samostatnou křivku, kterou následně zakreslí do předem definovaného grafu.

```
# Loop over each color channel
for i, color in enumerate(colors):
    # Calculate the histogram for the current color channel
    histogram, bin_edges = np.histogram(
        self.image[:, :, i], bins=256, range=(0, 256)
    )
    # Plot the histogram for the current color channel
    plt.plot(bin_edges[0:-1], histogram, color=color)
```

Doplnění grafu o název a popis hlavních os a následné zobrazení grafu.

```
# Add a title and labels to the plot
plt.title('Color Histogram')
plt.xlabel('Color Value')
plt.ylabel('Pixel Count')

# Display the histogram
plt.show()
```

Datové struktury

Hlavní datovou strukturou je NumPy pole (array), které reprezentuje obrázek. Každý pixel je reprezentován trojicí hodnot (R, G, B). Další datovou strukturou je seznam (list) využitý na uložení těchto tří barev.

Vstupní/výstupní data

Vstupní data jsou obrázek ve formátu JPG, PNG, nebo GIF. Výstupem je histogram zobrazený pomocí Matplotlib.

Problematická místa kódu

Jedním z problematických míst kódu je, že kód je pevně nastaven na zpracování obrázků se třemi kanály barev. To znamená, že nebude fungovat správně pro černobílé obrázky nebo obrázky s alfa kanálem.

Možná vylepšení

Kód by mohl být rozšířen tak, aby automaticky detekoval počet kanálů barev obrázku a správně zpracoval obrázky s různým počtem kanálů. Také by mohl být rozšířen o možnost nastavení počtu binů v histogramu.