

Semestrální projekt VYF – závěrečná zpráva

**Převod barev do odstínů šedi pro zachování rozlišovací
schopnosti**

Bc. Rostislav Kučera, rkucer0a@stud.fit.vutbr.cz

1 Úvod

V současné době je používáno několika algoritmů pro převod barevných obrázků do odstínů šedi. Vznikne tak jakási černobílá fotografie. Nicméně každý takovýto algoritmus poskytuje odlišné výsledky. Některé z nich, ač jsou implementačně jednoduché, nemusí vhodným způsobem interpretovat původní barevnou fotografii, může dojít ke ztrátě detailů, které mohou být ve výsledné fotografii však chtěné. Proto dochází k vymýšlení algoritmů takových, které dokáží převést barevný rozsah původních fotografií do černobílých obrázků tak, aby určité detaily zůstaly dobře rozlišeny.

2 Nejpoužívanější převod do stupňů šedi

Nejčastěji používaným, a zároveň taky nejjednodušším takovým algoritmem je převod využívající standardně používaného modelu RGB. Výsledný stupeň šedi je pak vypočten podle následujícího jednoduchého vzorce:

$$GREY = 0,21R + 0,72G + 0,07B \quad (1)$$

Ačkoliv je tento způsob velmi jednoduchý, velmi často dochází ke ztrátě detailů, protože některé barvy mohou být takto převedeny na stejný odstín šedé. Dochází k tomu velmi často v případě normálně velmi výrazných, od sebe odlišitelných barev. Proto je třeba tento algoritmus označit za nevhodný při převodu obrázků, které ukazují různé grafy, zejména tzv. koláčové, kde výsledné na sebe navazující barvy mohou působit jako jeden odstín šedé.

3 Popis problému a řešení

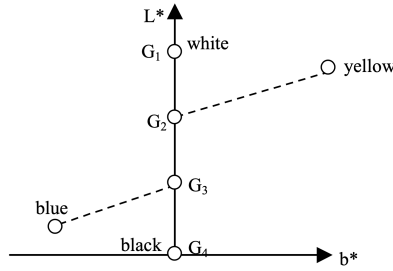
V mé práci se tak budu věnovat implementaci algoritmu popsaného v práci [1]. Algoritmus se zakládá na výpočtu výskytu všech různých barev ve vstupním obrázku a jeho ideálního namapování na stupeň šedi (na stupnici 0 - 100). V tomto článku jsou konkrétně popsány dva algoritmy, které implementují algoritmus se stejnou problematikou, liší se umístěním výsledných barev na šedotónové stupnici.

3.1 Stejná světelná vzdálenost

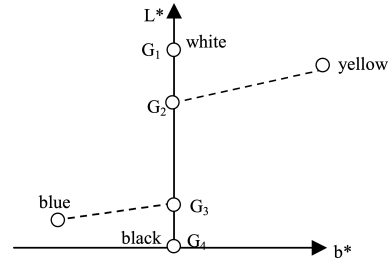
Základním algoritmem, který dokáže převést barvy vstupního obrázku na šedotónovou stupnici a zároveň zachovat rozlišovací schopnosti, je založen na rovnoměrném rozdělení šedotónové stupnice. V obrázku je tedy spočten počet barev a ty jsou rovnoměrně rozloženy na této stupnici.

3.2 Vážená světelná vzdálenost

Podobně pak funguje i algoritmus, který nebere v potaz jen světelný rozdíl mezi barvami, ale také samotný barevný rozdíl definovaný standardem CIE 1976. Popsán je následujícím vzorcem:



Výsledek mapování barev do stupňů šedi pomocí techniky stejné světelné vzdálenosti[1].



Výsledek mapování barev do stupňů šedi pomocí techniky vážené světelné vzdálenosti[1].

$$L^*_{n,out} = \begin{cases} L^*_{\min} & \text{if } n = 1 \\ L^*_{\min} + (L^*_{\max} - L^*_{\min}) \frac{\sum_{i=2}^n \Delta E_{i,i-1}}{\sum_{i=2}^N \Delta E_{i,i-1}} & \text{if } 2 \leq n \leq N \end{cases}$$

Figure 2: Vzorec algoritmu mapování barev do stupňů šedi pomocí techniky vážené světelné vzdálenosti. n značí pořadí barvy od nejtmaší po nejsvětlejší a N pak celkový počet barev v obrázku[1].

Každý obrázek však obsahuje spoustu barevného šumu, který je třeba pro vhodný výsledek obou těchto algoritmů odstranit. K tomu jsem využil algoritmu *K-Means Clustering*, který dokáže vytvořit shluky nejvíce se vyskytujících barev. Díky tomu můžeme ve výsledném algoritmu počítat pouze s těžišti těchto shluků a omezit tak barevný šum.

4 Implementace

Program je implementován v jazyce Python.

Samotný program je pak tvořen funkcí `main`, která zpracuje vstupní argumenty a následně zavolá funkci `equal` pro algoritmus stejné světelné vzdálenosti nebo `col_diff` pro váženou světelnou vzdálenost. Obě funkce nejdříve načtou obrázek a následně je původní 3D pole převedeno na 2D pole, pro které je následně zavolána funkce `kmeans` z knihovny `scipy`. Následně složky původních barev převedeme na složky těžiště, do jehož shluku barva patří. Poté už je vypočítána šedotónová intenzita pomocí vzorce 2.

Zde se již oba algoritmy rozcházejí, zatímco algoritmus stejné světelné vzdálenosti počítá pouze tuto hodnotu šedé, druhý zmíněný ještě počítá samotné vzdálenosti původních barev v modelu CIE1976. Vzdálenost i převod je volán z knihoven `colormath`. Následně je pak obrázek zpět převeden a uložen.

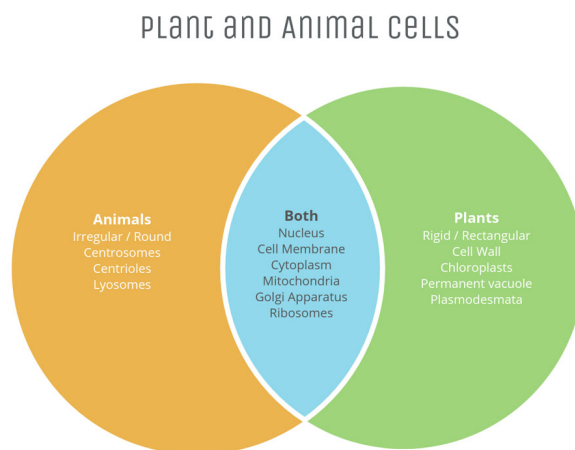
Program je možné spustit následujícími dvěma způsoby (módy *equal* a *weighted*, *num_colors*

představuje výsledný počet barev):

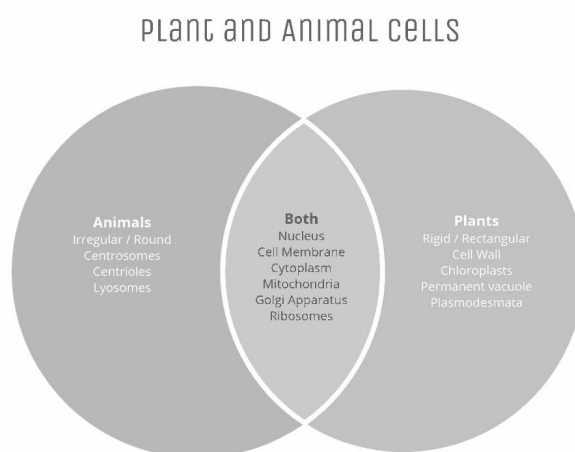
```
python3 greyscale.py -m <mode> -n <num_colors> -i <inputfile> -o <outputfile>
python3 greyscale.py --mode <mode> --colors <num_colors> --ifile inputfile\
--ofile <outputfile>
```

5 Výsledky

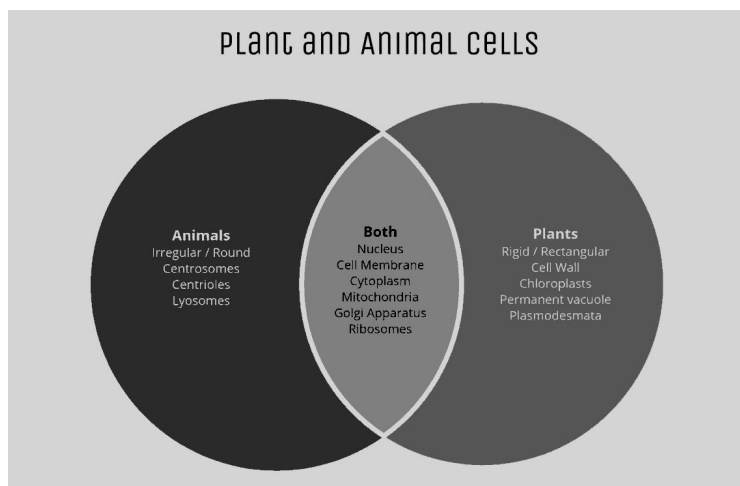
Program v jednom běhu uloží jak obrázek zpracovaný jednou z metod, tak také obrázek převedený standardně používaným algoritmem. Je tedy možné porovnat všechny tři metody. Na těchto obrázcích lze vidět zásadní rozdíl mezi jednoduchým převodem a taky oběma mnou implementovanými algoritmy. Na výstupech lze vidět, že metoda vážených světelných vzdáleností má daleko lepší rozlišovací schopnosti a lépe reprezentuje původní barvy (viditelné na příkladu bílého pozadí).



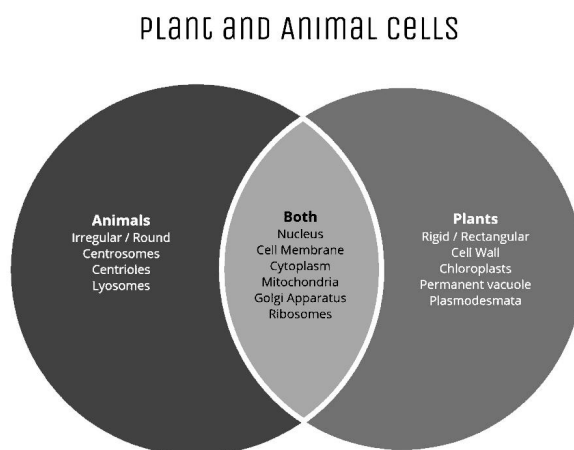
Vstupní barevný obrázek.



Výsledek mapování barev do stupňů šedi pomocí standardního převodu pomocí vzorce 2.



Výsledek mapování barev do stupňů šedi pomocí techniky stejné světelné vzdálenosti.



Výsledek mapování barev do stupňů šedi pomocí techniky vážené světelné vzdálenosti.

6 Závěr

V této práci došlo k popisu dvou metod pro převod barevného obrázku na černobílý, který zachovává původní rozlišovací schopnosti. Algoritmy se tedy hodí především na převod obrázků zobrazujících grafy či diagramy, pro klasické fotografie nemá tato metoda význam (omezuje počet barev). Zároveň jsem také musel vyřešit problém barevného šumu, který má zásadní vliv na výslednou kvalitu převodu. Proto jsem musel použít vhodnou metodu pro jeho odstranění.

V budoucnu by bylo možné tuto metodu implementovat do některého s volně dostupných komplexních nástrojů pro úpravu obrázků. Nešlo by tak o tzv. volně stojící aplikaci, ale vzniklo by mnohem komplexnější řešení, které by se mohlo dostat k velkému množství uživatelů.

References

- [1] BALA, R., AND BRAUN, K. Color-to-grayscale conversion to maintain discriminability. *Proc. SPIE 5293* (01 2004).