

# Dithering rastrového obrazu

Petr Fiedler

ČVUT–FIT

fiedlpe5@fit.cvut.cz

7. května 2022

## 1 Úvod

Práce se věnuje tvorbě aplikace pro kvantizaci barev rastrového obrazu a následný dithering, tedy proces, který dokáže omezit barevnou paletu obrazu tak, aby byl výsledek co nejpodobnější původnímu obrazu. Může tedy dojít ke snížení z původních tisíců barev použitých v obrázku až na pouhé desítky barev, přičemž nově vzniklý obrázek je při dostatečném oddálení skoro nerozeznatelný od obrázku originálního.

Cílem práce není vytvořit efektivní nástroj pro časté zpracovávání obrazu, ale spíše nástroj pro demonstraci různých metod používaných při ditheringu. S tím také souvisí forma programu - aplikace s grafickým uživatelským rozhraním. Tato forma je vhodná pro experimentaci s různým nastavením zpracovávajících metod. Aplikace umožňuje zpracovávání pouze statických obrázků.

## 2 Kvantizace barev

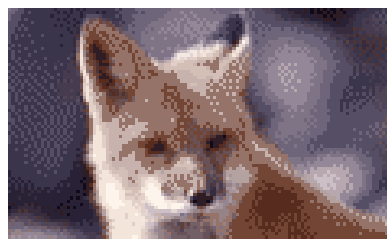
Prvním krokem při zpracovávání obrazu je kvantizace barev, neboli zúžení barevné palety využitě pro vykreslení obrázku. Bez omezení barevné palety by samotný dithering neměl smysl, protože by barevné pixely obrázku už byly rozloženy ideálním způsobem.

### 2.1 Algoritmus Median Cut

Jednou z implementovaných možností pro výběr barevné palety je algoritmus Median Cut. Tento algoritmus spočívá v rozdělování barevného prostoru v mediánu barevné složky, která má největší rozsah. Pro  $n$  zvolených iterací<sup>1</sup>, je barevný prostor rozdělen do  $2^n$  úseků. Výsledná paleta vznikne zpřůměrováním barevných hodnot v každém z těchto úseků.

Tento algoritmus vybere barvy, které nejlépe reprezentují vstupní obrázek. Dle mého pozorování není moc vhodný pro dithering, protože výsledná paleta se skládá z dominantních, tedy často podob-

ných, barev. Většinou tedy nelze dosáhnout vykreslení všech odstínů barev na obrázku, zejména v detailech s nedominantní barvou.

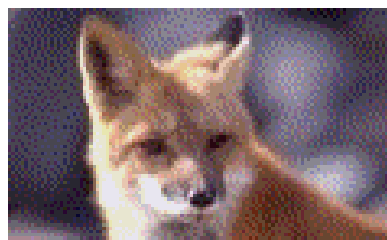


Obrázek 1: Barevná paleta vytvořená algoritmem Median Cut s 5 bity barev

### 2.2 Metoda ořezávání bitů

Pro zúžení barevné palety může být využita i metoda ořezávání bitů z konců hodnot jednotlivých barevných kanálů. Ztráta informace je poté nahrazena konstantní hodnotou, aby se ztráta kompenzovala.

Výhoda této metody je, že jsou použity barvy původní barevné palety, tedy nedojde k výrazné ztrátě rozsahu odstínů. Naproti tomu se však odstíny mírně změní a může tedy docházet ke změně odstínu obrázku jako celku nebo k přepálení.



Obrázek 2: Barevná paleta vytvořená ořezáním 5 bitů z barevných kanálů

### 2.3 Statické barevné palety

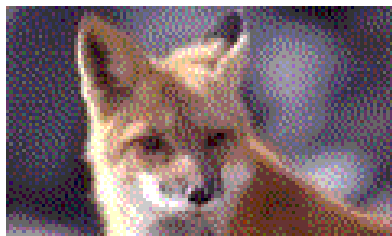
Vybraná barevná paleta může být i statická. Tedy například monochromatická paleta nebo paleta bezpečných barev<sup>2</sup>, což je paleta obsahující 216 rovnoměrně rozložených barev. Tato paleta má spíše

<sup>1</sup>bitů, protože velikost palety roste v závislosti na počtu iterací exponenciálně se základem 2

<sup>2</sup>Web Safe Palette

historický význam. Byla používána pro bezpečný přenos obrazu přes internet. Na to byl dithering velmi užitečný.

Protože je tato paleta rozložena rovnoměrně, je na dithering poměrně vhodná, protože dokáže vhodným uspořádáním barevných pixelů vytvořit libovolný odstín. Při použití této palety je však poměrně viditelný barevný šum při pohledu na jednotlivé pixely.



Obrázek 3: Paleta bezpečných barev

### 3 Dithering

Pokud bychom vstupnímu obrázku přiřadili pouze nejbližší barvu z vybrané palety, nedosáhli bychom kýženého výsledku. Aby se výstupní obrázek více podobal vstupnímu, musíme barevné pixely vhodně přeuspořádat.

#### 3.1 Metoda rozptylu kvantizační chyby

Jednou z implementovaných metod ditheringu je metoda rozptylu kvantizační chyby<sup>3</sup>. Jedná se o rozložení rozdílu původní a nově přiřazené barvy mezi blízké pixely.

Pro vhodný rozptyl byly nalezeny mapy poměru rozložení kvantizační chyby, takové, aby došlo k co nejlepšímu vizuálnímu výsledku. Za zmínku stojí Floyd-Steinbergova mapa, která ovlivňuje pouze 4 sousední pixely, a přesto se výsledek dá srovnávat s mapami, které ovlivňují blízkých pixelů více.

Algoritmus pro metodu rozptylu chyby je velice pomalý. Je vhodný<sup>4</sup> pouze pro statický obrázek. Zato se ve většině případů jedná o metodu ditheringu s nejlepšími výsledky.

#### 3.2 Ordered dithering

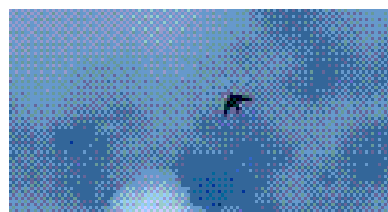
Další metodou je takzvaný ordered dithering. Tato metoda spočívá ve změně barevného prahu podle předem dané matice vzoru. Tato matice také závisí na rozložení barev v barevném prostoru. Ve výsledku je tedy vytvořen šum, který zjemní přechody mezi barevnými odstíny.



Obrázek 4: Dithering pomocí Floyd-Steinbergovy metody

V této práci jsem využil Bayerovské mapy jako matice barevného prahu. Tyto mapy generují pomocí bitových operací v závislosti na délce strany. Při použití Bayerovských matic jsou na obrázku vidět zřejmé opakující se vzory.

Ordered dithering je vhodný pro obrázky s rozsáhlými oblastmi s podobným barevným odstínem. Je také rychlejší než metoda rozptylu kvantizační chyby.



Obrázek 5: Dithering s využitím Bayerovské matice velikosti 8x8

### 4 Uživatelské rozhraní

Nečekaně významnou částí této práce se stala tvorba samotného uživatelského rozhraní, které umožňuje nastavovat metody procesu zpracování obrázku. Také umožňuje obrázek zobrazovat a přibližovat. Zde nebylo snadné učinit přibližování výpočetně méně náročné a takové, aby se jednotlivé pixely zobrazovaly správně.

Dále jsem zde narazil na nutnost práce s vlákny, která byla potřeba použít, protože je vhodné, aby šlo uživatelské rozhraní ovládat, zatímco se obrázek zpracovává. Také jsem potřeboval najít způsob, aby šlo vlákno ukončit na vyžádání.

### 5 Výsledky

Podařilo se mi implementovat algoritmy nejvýznamnějších metod pro dithering a barevnou kvantizaci obrazu. Očekával jsem však, že se mi podaří zahrnout metod více. Myslím si ale, že pro demonstraci tohoto procesu zpracování obrázku je výsledek dostatečný a cíl práce byl naplněn.

<sup>3</sup>Error Diffusion

<sup>4</sup>i vizuálně

## 6 Závěr

Vytvořil jsem aplikaci vhodnou pro experimentaci s ditheringem obrazu. Aplikace je dobrá pro jednorázovou tvorbu grafického díla, kde je možné měnit výstup dle nastavených parametrů, a tedy zkoušet dosáhnout co nejlíbivějšího výsledku. Nemyslím si, že by měla aplikace významné praktické využití, zvláště, když v dnešní době moderních zobrazovacích technologií už dithering není tolik potřebný. Dalo by se říci, že aplikace může mít spíše umělecký význam. Například by se dala využít v textilním průmyslu při vyšívaní.

V budoucnu by bylo možné do aplikace přidat více algoritmů ditheringu a více možností výběru barevné palety, obzvláště import palety vlastní. Přestože jsem projevil snahu o optimalizaci, určitě by bylo možné proces optimalizovat ještě více<sup>5</sup>, aby nezabíral takové množství času.

## Reference

- [1] Bisqwit. Arbitrary-palette positional dithering algorithm. online, 2021. [cit. 2022-05-07] <https://bisqwit.iki.fi/story/howto/dither/jy/>.
- [2] CodeProject. Octree color palette. online, 2022. [cit. 2022-05-07] <https://www.codeproject.com/Articles/109133/Octree-Color-Palette>.
- [3] Finxter. How to kill a thread in python? online, 2022. [cit. 2022-05-07] <https://blog.finxter.com/how-to-kill-a-thread-in-python/>.
- [4] Tanner Helland. Image dithering: Eleven algorithms and source code. online, 2020. [cit. 2022-05-07] <https://tannerhelland.com/2012/12/28/dithering-eleven-algorithms-source-code.html>.

---

<sup>5</sup>také ale na úkor srozumitelnosti kódu