

# Počítačová grafika, 2. díl

EDUARD BARTL

Přírodovědecká fakulta UP, Olomouc

Po delší přestávce se vracíme k seriálu o počítačové grafice. Motivací pro napsání nového dílu bylo pro autora mimo jiné setkání se středoškolskými učiteli, které pořádala na konci roku 2021 Katedra informatiky PřF UP v Olomouci [5]. Během setkání totiž vyjádřili přítomní středoškolští učitelé zájem o toto téma.

V předcházejícím dílu jsme uvedli definici digitálního obrazu. V tomto pokračování se budeme zabývat jeho úpravou. Důvody, proč je potřebné věnovat se metodám úpravy digitálního obrazu, jsou nasnadě. Digitálním obrazem může být například snímek pořízený digitálním fotoaparátem, tento snímek však může být nevhodně pořízený (například kvůli nedostatečnému osvětlení) nebo poškozený (kvůli elektromagnetickému rušení). Můžeme také chtít fotografický snímek ozvláštnit nějakým speciálním efektem.

Všechny úpravy, kterými se budeme zabývat, se provádějí v takzvané *prostorové doméně*, tedy přímo s hodnotami pixelů obrázku. Existují i úpravy, které se provádějí ve *frekvenční doméně*. Manipuluje se pak s frekvencemi, které jsou jistým způsobem přítomny v každém obrázku. K pochopení této problematiky je však potřeba zvládnout aparát *teorie signálu*, zejména pak takzvanou *Fourierovu transformaci*. Jakkoli je toto téma zajímavé, zabývat se jím v tomto dílu nebudeme. Zvědavého čtenáře můžeme odkázat na pěkně napsanou knížku [1].

## 1. Převedení barevného obrazu na šedotónový

Představme si následující situaci. Digitálním fotoaparátem pořídíme snímek, který uložíme v podobě rastrového obrazu. Na tomto snímku je znázorněn například hrad Würzburg, obr. 1. Pro zdůraznění nálady tohoto snímku se rozhodneme převést ho na šedotónový.<sup>1)</sup>

---

<sup>1)</sup>V běžné mluvě se často používá termín *černobílý* snímek. Tento termín však poukazuje na skutečnost, že jsou ve snímku použity pouze dvě barvy – černá a bílá. Šedotónový obrázek naproti tomu může obsahovat i různé stupně šedi. Šedotónovým obrázkům se také někdy říká *achromatické*.



Obr. 1 Barevný snímek hradu Würzburg, foto autor

Předpokládejme, že snímek uložený v podobě rastrového obrazu má barevnou hloubkou 24 bitů a pro reprezentaci barev použijeme RGB model. Z předchozího dílu víme, že barva každého pixelu bude popsána trojicí  $\langle r, g, b \rangle$ , kde čísla  $r$ ,  $g$ ,  $b$  vyjadřují zastoupení základních barev RGB modelu a na reprezentaci každého tohoto čísla máme k dispozici jednu třetinu z celkové barevné hloubky, tedy 8 bitů. Číslům  $r$ ,  $g$  a  $b$  budeme stručně říkat *barevné složky*. Tyto barevné složky tedy nabývají hodnot z množiny  $\{0, 1, \dots, 255\}$ ; čím větší číslo, tím větší zastoupení dané barvy. Víme také, že každý pixel získaného šedotónového obrázku bude mít barvu, která se v RGB krychli nachází někde na tělesové úhlopříčce. To znamená, že všechny tři barevné složky budou mít stejnou hodnotu, kterou si označíme  $y$  (tímto označením poukazujeme na anglické slovo *gray* – šedý). Získaný šedotónový obrázek proto bude mít výrazně menší barevnou hloubku, v našem případě osmibitovou.

Zbývá však vymyslet to nejdůležitější – stanovit hodnotu  $y$  tak, aby výsledný šedotónový obrázek odpovídal naší (zkušeností dané) představě. Je zřejmé, že se na hodnotě  $y$  musí podílet všechny tři barevné složky. Vzhledem k tomu, že RGB model používá aditivní skládání barev (jak jsme si vysvětlili v předchozím dílu), musí platit, že čím větší bude některá barevná složka, tím bude výsledná barva světlejší a proto i hodnota  $y$  bude větší. Nabízí se tedy všechny tři barevné složky sečíst:

$$y = r + g + b.$$

Sečtením by však mohla hodnota  $y$  padnout mimo zobrazitelný rozsah (při naší zvolené barevné hloubce nad hodnotu 255). Výsledek je tak potřeba

normalizovat, tedy vydělit třemi (počtem barevných složek). Získáváme tak následující vzorec:

$$y = \frac{r + g + b}{3}. \quad (1)$$

Nesmíme však zapomenout, že stupeň šedi vyjadřujeme přirozeným číslem, výsledek výše uvedeného podílu proto musí být zaokrouhlen.

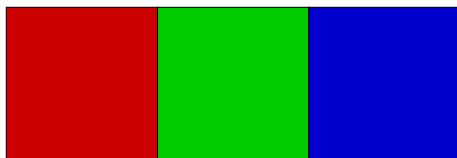
Vidíme, že se ve skutečnosti jedná o aritmetický průměr barevných složek. Toto řešení je z technického pohledu zcela správné. Podívejme se nyní na výsledný šedotónový snímek (obr. 2) a položme si otázku, zdali je výsledek přesně takový, jak jsme očekávali.



Obr. 2 Šedotónový snímek hradu Würzburg

Při podrobnějším prostudování obrázku možná dojdeme k závěru, že jednotlivé stupně šedi neodpovídají naší zkušenosti. Konkrétněji řečeno, části obrazu, které byly zelené, se mohou jevit jako příliš tmavé. Naopak části obrazu, které byly modré, jsou v šedotónovém obrazu příliš světlé.

Tento jev je dán skutečností, že lidské oko (přesněji řečeno mozek) vnímá zelenou barvu světlejší, než je ve skutečnosti, a naopak modrou barvu vnímá tmavější. Demonstrovat to můžeme na barevných čtvercích zobrazených na obr. 3. První čtverec je obarven barvou  $\langle 204, 0, 0 \rangle$ , druhý čtverec barvou  $\langle 0, 204, 0 \rangle$  a třetí čtverec barvou  $\langle 0, 0, 204 \rangle$ . Pro každý čtverec platí, že jedna barevná složka nabývá hodnoty 204, zbývající dvě složky pak nabývají hodnoty 0. Všechny tři čtverce bychom tedy měli vnímat jako stejně světlé. Ve skutečnosti však většina lidí vnímá zelený čtverec jako nejsvětlejší a modrý čtverec jako nejtmaší.



Obr. 3 Zelenou barvu vnímáme jako nejsvětější

Tento problém vyřešíme tak, že místo aritmetického průměru použijeme průměr vážený:

$$y = \frac{v_1 \cdot r + v_2 \cdot g + v_3 \cdot b}{v_1 + v_2 + v_3},$$

přičemž váhy  $v_1$ ,  $v_2$ ,  $v_3$  nastavíme tak, že největší bude  $v_2$ , menší  $v_1$  a nejmenší  $v_3$ . Pokud navíc tyto váhy zvolíme tak, že jejich součet bude roven jedné, jmenovatel zlomku vypadne. Zkušenost ukázala, že lidskému vnímání nejlépe vyhovují následující hodnoty vah:  $v_1 = 0,299$ ,  $v_2 = 0,587$ ,  $v_3 = 0,114$ . Konečný vzorec pro výpočet hodnoty  $y$  pak vypadá takto:

$$y = 0,299 \cdot r + 0,587 \cdot g + 0,114 \cdot b. \quad (2)$$

Opět nezapomeňme, že je nutné provést zaokrouhlení.

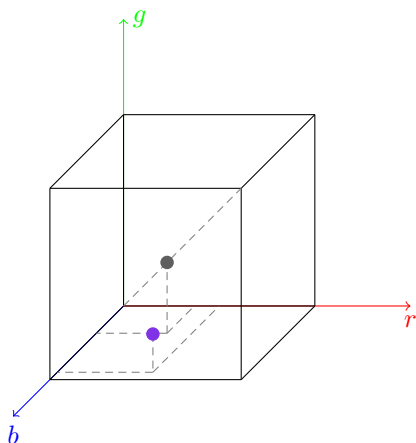
## 2. Desaturace barvy

Dalším zajímavým příkladem úpravy obrázku je *desaturace barev*. Naší snahou bude snížit *sytosť* barevných odstínů v obrázku; v tomto smyslu se proto budeme snažit přiblížit barevný obrázek k šedotónovému. Míra tohoto přiblížení bude dána hodnotou parametru, který bude zadán na vstupu desaturační metody. Na desaturaci barvy tak můžeme pohlížet jako na zobecnění metody popsané v předchozí sekci.

Uvažujme fialovou barvu, ta má v RGB krychli souřadnice  $\langle 128, 51, 230 \rangle$ . Tato barva je znázorněna fialovým bodem na obr. 4. V tomtéž obrázku je dále na tělesové uhlopříčce zobrazen šedý bod, který je šedotónovou variantou dané fialové barvy. Jeho souřadnice  $\langle 94, 94, 94 \rangle$  jsme získali pomocí vzorce (2):

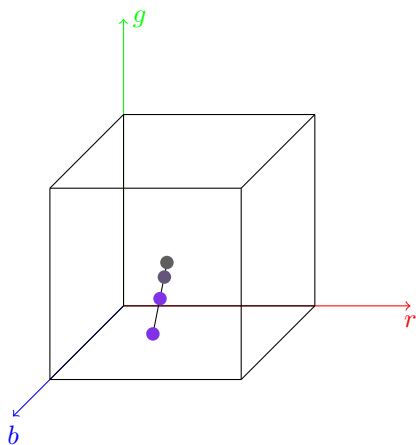
$$y = 0,299 \cdot 128 + 0,587 \cdot 51 + 0,114 \cdot 230 \doteq 94.$$

Geometrický pohled na desaturaci barvy je takový, že se snažíme po přímce posunout z fialového bodu do šedého bodu – čím blíže budeme



Obr. 4 Fialová barva a příslušná šedá barva v RGB krychli

šedému bodu, tím více bude fialová barva desaturovaná. Musíme tedy zavést parametr  $d$  určující, jak moc budeme chtít danou barvu desaturovat; jinými slovy, jak moc se budeme chtít vzdálit od fialové barvy směrem k příslušné šedé barvě. Parametr  $d$  bude proto nabývat hodnot z intervalu  $[0, 1]$ . Čím větší bude jeho hodnota, tím více bude daná barva desaturovaná (pro  $d = 0$  nebude desaturovaná vůbec, pro  $d = 1$  bude desaturovaná zcela, splyne tedy s příslušnou šedou barvou), jak můžeme vidět na obr. 5.



Obr. 5 Desaturace fialové barvy znázorněná v RGB krychli

Další odvození zkusme provést obecně. Uvažujme barvu s barevnými složkami  $\langle r, g, b \rangle$ . Naším úkolem je stanovit barevné složky desaturované barvy  $\langle r_d, g_d, b_d \rangle$  v závislosti na hodnotě parametru  $d \in [0, 1]$ . Není to nijak složité, využijeme-li základních poznatků z analytické geometrie, dostaneme následující rovnost:

$$\langle r_d, g_d, b_d \rangle = \langle r, g, b \rangle + d \cdot \langle y - r, y - g, y - b \rangle,$$

kde  $y$  je šedá barva vypočtená podle vztahu (2). Provedme ještě jednoduchou úpravu:

$$\begin{aligned} \langle r_d, g_d, b_d \rangle &= \langle r, g, b \rangle + d \cdot \langle y - r, y - g, y - b \rangle = \\ &= \langle r, g, b \rangle + d \cdot \langle y, y, y \rangle - d \cdot \langle r, g, b \rangle = \\ &= (1 - d) \cdot \langle r, g, b \rangle + d \cdot \langle y, y, y \rangle. \end{aligned} \quad (3)$$

Díky této drobné úpravě můžeme pěkně vidět vliv parametru  $d$  na výslednou desaturovanou barvu  $\langle r_d, g_d, b_d \rangle$  – zvětšování hodnoty  $d$  vede k nárůstu vlivu šedé (zcela desaturované) barvy  $\langle y, y, y \rangle$  a současně k poklesu vlivu původní (plně satureované) barvy  $\langle r, g, b \rangle$ .<sup>2)</sup>

Vrátíme se nyní k příkladu s fialovou barvou a právě odvozený vzorec pro desaturaci barvy si vyzkoušíme na konkrétních číslech. Jak jsme si řekli, pokud zvolíme  $d = 0$ , dostaneme plně satureovanou fialovou barvu

$$\langle r_0, g_0, b_0 \rangle = \langle r, g, b \rangle = \langle 128, 51, 230 \rangle,$$

pro  $d = 0,5$  získáme

$$\langle r_{0,5}, g_{0,5}, b_{0,5} \rangle = 0,5 \cdot \langle 128, 51, 230 \rangle + 0,5 \cdot \langle 94, 94, 94 \rangle \doteq \langle 111, 72, 162 \rangle,$$

pro  $d = 0,8$  platí

$$\langle r_{0,8}, g_{0,8}, b_{0,8} \rangle = 0,8 \cdot \langle 128, 51, 230 \rangle + 0,8 \cdot \langle 94, 94, 94 \rangle \doteq \langle 101, 85, 121 \rangle,$$

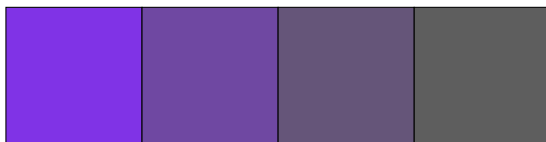
a pro  $d = 1$  na závěr dostaneme

$$\langle r_1, g_1, b_1 \rangle = \langle y, y, y \rangle = \langle 94, 94, 94 \rangle.$$

Všechny tyto barvy jsou znázorněny na obr. 6.

---

<sup>2)</sup>Vztahu (3) se říká *konvexní kombinace* bodů  $\langle y, y, y \rangle$  a  $\langle r, g, b \rangle$ .



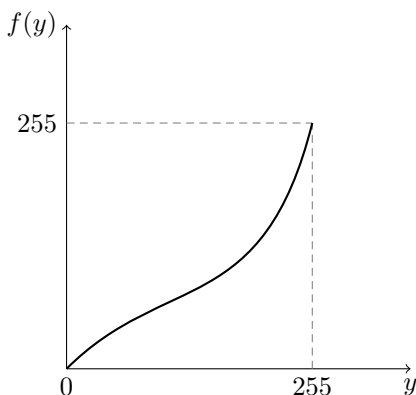
Obr. 6 Desaturovaná fialová barva; parametr  $d$  je pro jednotlivé čtverce (zleva) nastaven na 0, 0,5, 0,8 a 1

### 3. Prahování

Ve zbytku tohoto dílu se omezíme pouze na úpravu šedotónových obrázků. Uvažovat tedy budeme digitální rastrový obraz s barevnou hloubkou například 8 bitů, stupně šedi jednotlivých pixelů vyjádříme čísly z množiny  $\{0, 1, \dots, 255\}$  (čím větší číslo, tím světlejší šedá). Všechny zbývající metody úpravy jsou založeny na transformaci těchto hodnot pixelů. Hledáme tedy funkci

$$f: \{0, 1, \dots, 255\} \rightarrow \{0, 1, \dots, 255\}$$

převádějící hodnotu pixelu  $y \in \{0, 1, \dots, 255\}$  na hodnotu  $f(y) \in \{0, 1, \dots, 255\}$  tak, aby to vyhovovalo požadovanému záměru. Tuto funkci znázorníme pomocí grafu, který může vypadat podobně jako graf na obr. 7.



Obr. 7 Graf funkce, která transformuje stupně šedi

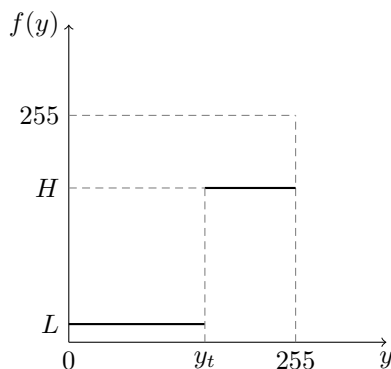
Zdůrazněme však, že tento graf není spojitý, je tvořen oddělenými body.

Jako spojitý vypadá jen kvůli zvolenému měřítku. Označení  $y$  vodorovné osy může působit nezvykle, nezapomeňme však, že na tuto osu vynášíme stupně šedi jednotlivých pixelů vstupního obrázku, které označujeme právě písmenem  $y$ .

První velmi jednoduchou úpravou, kterou budeme řešit volbou vhodné funkce  $f$ , je takzvané *prahování*. Jde pouze o to, že si nejprve zvolíme hodnotu prahu  $y_t$ ; stupně šedi, které tuto hodnotu přesáhnou, zobrazíme na hodnotu  $H$ , ostatní hodnoty zobrazíme na  $L$ . Velmi často se  $L$  nastavuje na 0 (jedná se tedy o černou barvu) a  $H$  na 255 (jde proto o bílou barvu). Funkce  $f$  je tak dána následujícím předpisem:

$$f(y) = \begin{cases} L & \text{pro } y \leq y_t; \\ H & \text{pro } y > y_t. \end{cases}$$

Graf funkce  $f$ , která realizuje prahování, je pak znázorněn na obr. 8.



Obr. 8 Graf funkce realizující prahování

Je zjevné, že prahováním je vstupní šedotónový obrázek převeden na obrázek obsahující pouze dvě různé barvy,<sup>3)</sup> v tomto případě se jedná o dva různé stupně šedi. Navzdory jednoduchosti prahování nachází tato metoda úpravy obrazu v počítačové grafice značné uplatnění. Uveďme například, že je možné volbou vhodné prahové hodnoty  $y_t$  separovat určité objekty v obrazu od pozadí.

<sup>3)</sup>Takovému obrázku se říká *monochromatický*.



## Závěr

V příštím pokračování se budeme věnovat dalším úpravám digitálního obrazu, například změně kontrastu, světlosti a podobně. Postupně se také dostaneme k obtížnějším tématům, jako je kupříkladu histogram a jeho využití v automatických úpravách obrazu.

## Literatura

- [1] *Gonzalez, R. C., Woods, R. E.*: Digital Image Processing. 3. vydání. Pearson Prentice Hall, 2008.
- [2] *Huges, J. F. a kol.*: Computer Graphics. Principles and Practice. 3. vydání. Addison-Wesley, 2014.
- [3] *Felkel, P., Sochor, J., Žára, J., Beneš, B.*: Moderní počítačová grafika. 2. vydání. Computer Press, 2005.
- [4] *Martíšek, D.*: Matematické principy grafických systémů. Littera, 2002.
- [5] Webové stránky Katedry informatiky PřF UP v Olomouci, sekce věnovaná spolupráci se středními školami <https://www.inf.upol.cz/setkani-ucitelu-informatiky>

# Programování bez proměnných

JAN LAŠTOVIČKA

Přírodovědecká fakulta UP, Olomouc

Proměnné ve většině programovacích jazyků hrají zásadní roli: vkládají hodnoty na určitá místa výrazů, uchovávají dočasné výsledky a v neposlední řadě označují argumenty funkce. Článek představuje jazyk Joy, kde pojem proměnné neexistuje.

Joy je navíc čistě funkcionální jazyk. Přestože funkcionální jazyky mají bohatou historii, jejich prvky prosakují do moderních jazyků až poslední dobou. Jazyk Joy tak může poskytnout netradiční úvod do stále více populárnějších technik funkcionálního programování.