Počítačová grafika, 3. díl

EDUARD BARTL

Přírodovědecká fakulta UP, Olomouc

Třetí díl série věnované počítačové grafice přímo navazuje na díl druhý. Zabývat se totiž budeme dalšími úpravami digitálního obrazu. Ukážeme si, jak změnit světlost a kontrast, podrobně se pobavíme o požití histogramu při úpravě obrazu.

V celém tomto dílu budeme pracovat s šedotónovým obrázkem s barevnou hloubkou k. Pixely tohoto obrázku tedy mohou nabývat celkem $K = 2^k$ různých hodnot (stupňů šedi). Dále budeme předpokládat, že má tento obrázek rozměry $m \times n$ pixelů. Hodnoty pixelů jsou popsány obrazovou funkcí $o\colon \{0, 1, \ldots, m-1\} \times \{0, 1, \ldots, n-1\} \to \{0, 1, \ldots, K-1\}$.

1. Úprava kontrastu

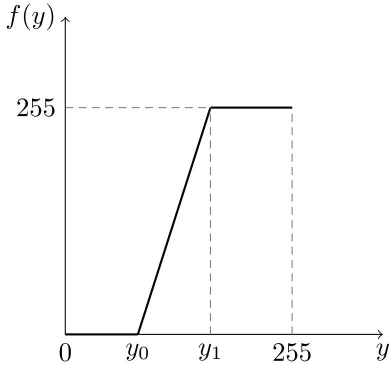
Ze zkušenosti víme, jak dopadne fotografie pořízená proti silnému světelnému zdroji – téměř každý pixel bude buď velmi světlý nebo velmi tmavý; jinými slovy, obrázek bude příliš *kontrastní*. Bude-li naopak snímek pořízen za nedostatečného osvětlení, bude působit šedivě, protože v něm budou chybět velmi tmavé a velmi světlé stupně šedi. Takový snímek bude nekontrastní. Typickou úpravou digitálního obrazu v prostorové doméně je tedy zvětšení nebo zmenšení kontrastu. Provádět ji budeme, stejně jako v případě prahování, aplikací vhodně zvolené funkce f na všechny pixely vstupního šedotónového obrázku.

Nejprve se věnujme zvětšení kontrastu. Jak jsme již řekli, v nekontrastním obrázku absentují velmi tmavé a velmi světlé stupně šedi. Lapidárně řečeno, nejtmavější stupně přítomné v obrázku jsou příliš světlé a naopak nejsvětlejší stupně jsou příliš tmavé.

Tento problém je řešitelný tak, že nejtmavější stupně ještě více ztmavíme a nejsvětlejší stupně ještě více zesvětlíme. Stupně šedi, které se nacházejí uprostřed, by neměly být naší úpravou výrazně dotčeny. Ze slovního popisu by mohlo být zřejmé, jak bude funkce f definována:

$$f(y) = \begin{cases} 0 & \text{pro } y < y_0; \\ 255 \cdot \frac{y - y_0}{y_1 - y_0} & \text{pro } y_0 \le y \le y_1; \\ 255 & \text{pro } y > y_1, \end{cases}$$

kde y_0 a y_1 jsou předem dané hodnoty stupňů šedi. Graf této funkce je ukázán na obr. 1. Jako obvykle je kvůli přehlednosti graf funkce f zakreslen pomocí spojité čáry, přestože je ve skutečnosti tvořen izolovanými body.



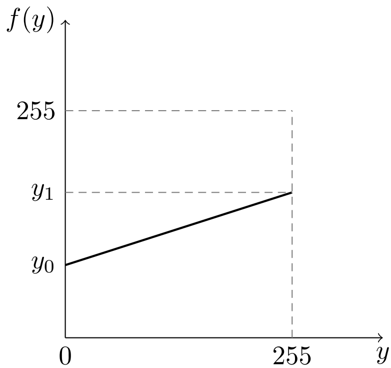
Obr. 1 Graf funkce realizující zvětšení kontrastu

Volba hodnot y_0 a y_1 se určuje právě podle hodnoty nejtmavějšího a nejsvětlejšího stupně šedi přítomného ve vstupním obrázku. Je velmi jednoduché je určit z *histogramu*, o kterém se budeme bavit v jedné z následujících sekcí.

Funkce f realizující zmenšení kontrastu může být definována například takto:

$$f(y) = \frac{y_1 - y_0}{255} \cdot y + y_0.$$

Graf této funkce je ukázán na obr. 2.



Obr. 2 Graf funkce realizující zmenšení kontrastu obrázku

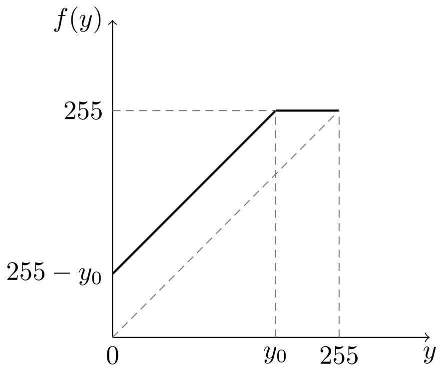
2. Úprava světlosti

Při fotografování se často dostáváme do situace, kdy je získaný snímek příliš světlý nebo naopak tmavý. Úprava světlosti se dá taktéž řešit volbou vhodné funkce f. Hodnoty pixelů vstupního obrázku pouze zvýšíme nebo naopak snížíme o konstantní hodnotu, která určuje míru zesvětlení nebo ztmavení.

Funkce f realizující zesvětlení obrázku je skutečně velmi jednoduchá:

$$f(y) = \begin{cases} 255 - y_0 + y & \text{pro } y < y_0; \\ 255 & \text{pro } y \geq y_0, \end{cases}$$

kde y_0 je parametr určující, jak moc bude obrázek zesvětlen. Hodnotu parametru y_0 můžeme nastavit na libovolné číslo z množiny $\{0, 1, \ldots, 255\}$ – čím menší číslo, tím více bude obrázek zesvětlen. Je celkem jasné, že pro nastavení $y_0 = 255$ získáme funkci $f(y) = y$, která hodnotu libovolného pixelu zobrazí na sebe samu – obrázek tedy zůstane nezměněn. Pokud však parametr y_0 snížíme, obrázek se stane světlejším. Je však nutné si uvědomit, že každý pixel mající hodnotu alespoň y_0 se aplikováním funkce f stane zcela bílým. Všechny tyto pixely tedy splynou – použitím funkce f pro $y_0 < 255$ tedy může dojít ke ztrátě informace obsažené v obrázku. Graf funkce realizující zesvětlení vznikne posunutím grafu funkce $f(y) = y$ po svislé ose o hodnotu $255 - y_0$. Tento graf je ukázán na obr. 3.

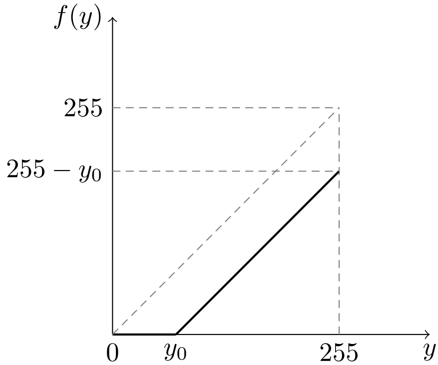


Obr. 3 Graf funkce realizující zesvětlení obrázku

Funkce f realizující ztmavení obrázku je velmi podobná, vše se však odehrává naopak – místo zvětšování hodnot pixelů dochází k jejich zmenšování:

$$f(y) = \begin{cases} 0 & \text{pro } y < y_0; \\ y - y_0 & \text{pro } y \geq y_0. \end{cases}$$

Graf této funkce je ukázán na obr. 4.



Obr. 4 Graf funkce realizující ztmavení obrázku

3. Histogram

Histogram je jednoduchým statistickým nástrojem používaným pro popis rozložení číselných dat. Významné použití nachází také v počítačové grafice, zejména v analýze digitálních obrazů. Histogram popisuje četnosti barev zastoupených v obrázku. Omezíme-li se na šedotónové obrázky, pak můžeme říci, že histogram popisuje četnosti jednotlivých stupňů šedi přítomných v daném obrázku.

Histogram h takového obrázku tedy můžeme definovat jako zobrazení $h\colon \{0, 1, \ldots, K-1\} \to \mathbb{N}$ přiřazující stupni šedi $a \in \{0, 1, \ldots, K-1\}$ počet pixelů, které mají hodnotu a. Matematicky zapsáno,

$$h(a) = \operatorname{card}(\{\langle x, y\rangle \mid o(x, y) = a\}),$$

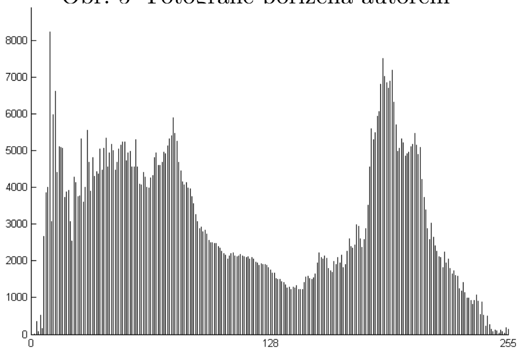
kde card(…) označuje počet prvků množiny uvedené na místě tří teček. Pro histogram h libovolného šedotónového obrázku zjevně platí

$$\sum_{a=0}^{K-1} h(a) = m \cdot n.$$

Jinými slovy, počet pixelů, které mají některou z hodnot $\{0, 1, \ldots, K-1\}$ je roven počtu všech pixelů daného obrázku.

Na obr. 5 je znázorněna šedotónová fotografie s 8 bitovou barevnou hloubkou, počet stupňů šedi je tedy $K = 2^8 = 256$. Histogram této fotografie je ukázán na obr. 6. Vidíme, že je znázorněn pomocí sloupcového grafu; výška sloupečku, který je na vodorovné ose umístěný na pozici a, říká, kolik pixelů má v daném obrázku hodnotu a.

Obr. 5 Fotografie pořízená autorem

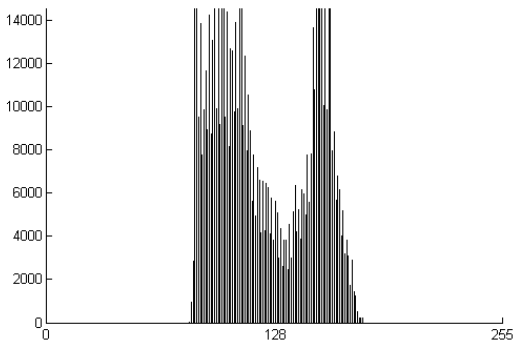


Obr. 6 Histogram obr. 5

Z histogramu můžeme ledacos vyčíst o příslušném obrázku. Například z uvedeného histogramu je dobře vidět, že velké množství pixelů je hodně tmavých (jejich hodnoty se blíží 0) nebo naopak hodně světlých (jejich hodnoty se blíží 255). Oproti tomu poměrně malé množství pixelů má hodnotu uprostřed šedotónové škály (tedy kolem hodnoty 128). Podle toho, jak vypadá tento histogram proto můžeme usoudit, že jemu příslušná fotografie je hodně kontrastní.

Pokud se naopak podíváme na histogram na obr. 8, můžeme okamžitě usoudit, že je jemu příslušný digitální obrázek velmi nekontrastní. Z tohoto histogramu je totiž patrné, že tmavé (téměř černé) a světlé (téměř bílé) pixely nejsou v obrázku vůbec zastoupeny. Obrázek bude patrně působit jako kdyby byl zahalen v mlze. Skutečně tomu tak je, jak se můžeme přesvědčit z obr. 7.

Obr. 7 Fotografie pořízená autorem



Obr. 8 Histogram šedotónového obrázku

Je důležité si uvědomit, že histogram nic neříká o umístění pixelů v daném obrázku – nese pouze kvantitativní informaci o jednotlivých stupních šedi. Je proto snadné najít několik obrázků se stejným histogramem; stačí pouze změnit umístění některých pixelů, aniž bychom měnili hodnoty těchto pixelů.

Zamysleme se nyní nad otázkou, jakým způsobem histogram vypočítat. Je to jednoduché. Stačí pouze projít každý pixel obrázku,[1)] pokud hodnota daného pixelu bude rovna a, pak v histogramu zvětšíme o jedna výšku sloupečku, který je umístěný na vodorovné ose na pozici a. Výpočet histogramu je popsán algoritmem 1.

Algoritmus 1 Výpočet histogramu

Vstup: šedotónový obrázek o rozměrech $m \times n$ určený obrazovou funkcí o
Výstup: histogram h
  for $y = 0, 1, \ldots, n-1$ **do**
    for $x = 0, 1, \ldots, m-1$ **do**
      $a = o(x, y)$
      $h(a) = h(a) + 1$
    end for
  end for

Rychlost algoritmu 1 závisí především na počtu pixelů vstupního obrázku. Vzhledem k tomu, že je nutné projít všechny pixely, není možné najít jiný algoritmus pro výpočet histogramu, který by byl výrazným způsobem rychlejší.

4. Binning histogram

Pro obrázky s barevnou hloubkou výrazně větší než je 8 bitů se zpravidla používají takzvané *binning histogramy*.[2)] Myšlenka je snadná: celou škálu stupňů šedi rovnoměrně rozdělíme do přihrádek, stupně patřící do jedné přihrádky pak reprezentujeme v binning histogramu jediným sloupečkem.

Předpokládejme tedy, že je celá škála stupňů šedi obsahujících K hodnot rovnoměrně rozdělena hodnotami $a_0, a_1, \ldots, a_{B-1}$. Celkem tedy máme B přihrádek, i-tá přihrádka je ohraničena hodnotami a_i a a_{i+1}. Binning histogram h pak můžeme definovat jako zobrazení

$$h\colon \{0, 1, \ldots, B-1\} \to \mathbb{N},$$

[1)] Často se pixely procházejí po jednotlivých řádcích, počínaje řádkem, který je zcela nahoře. Na způsobu průchodu však v tomto případě nezáleží, důležité je pouze projít je všechny.

[2)] Tento název je odvozen z anglického slova *bin*, což znamená *popelnice*. V tomto případě však budeme toto slovo překládat jako *přihrádka*.

které přiřazuje indexu i přihrádky počet pixelů, které mají hodnotu ležící v této přihrádce. Matematicky to můžeme vyjádřit následovně:

$$h(i) = \mathrm{card}\{(x, y) \mid a_i \leq o(x, y) < a_{i+1}\},$$

kde $i \in \{0, 1, \ldots B - 1\}$. Algoritmus 2 ukazuje, jak binning histogram vypočítat. Můžeme si všimnout, že je tento algoritmus velmi podobný algoritmu 1, pouze se v něm navíc řeší příslušnost stupně šedi do příslušné přihrádky. Funkce floor použitá v tomto algoritmu ořezává desetinnou část daného čísla.

Algoritmus 2 Výpočet binning histogramu

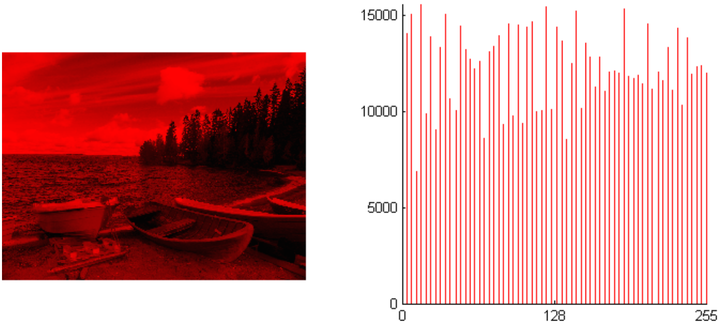
Vstup: šedotónový obrázek o rozměrech $m \times n$ určený obrazovou funkcí o; počet přihrádek B
Výstup: binning histogram h
  for $y = 0, 1, \ldots, n - 1$ **do**
    for $x = 0, 1, \ldots, m - 1$ **do**
      $a = o(x, y)$
      $i = \mathrm{floor}(a \cdot \frac{B}{K})$
      $H(i) = H(i) + 1$
    end for
  end for

5. Histogram barevných obrázků

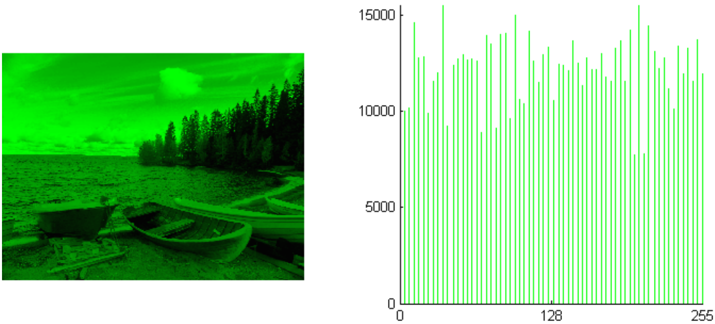
Zastavme se krátce u výpočtu histogramů barevných obrázků. K tomuto výpočtu je možné přistoupit velmi jednoduše. Barevný obrázek rozložíme na jednotlivé barevné složky. Získáme tak tři obrázky stejných rozměrů jako byl původní barevný obrázek – první z nich (v případě použití modelu RGB) nese informaci o zastoupení červené barvy, druhý z nich nese informaci o zastoupení zelené barvy a poslední z nich informaci o zastoupení modré barvy. Na každý z těchto obrázků tak můžeme v principu nahlížet jako na šedotónový obrázek (hodnota každého pixelu je jednotlivé číslo, nikoliv trojice, jak tomu bylo v případě barevného obrázku).

Pomocí algoritmu 1 nebo algoritmu 2 pak vypočteme pro každou barevnou složku samostatný histogram. Tyto histogramy si v souladu s označením jednotlivých barevných složek pojmenujeme jako h_R, h_G a h_B. Na histogram původního barevného obrázku 9 pak nahlížíme jako na trojici histogramů $\langle h_\mathrm{R}, h_\mathrm{G}, h_\mathrm{B}\rangle$. Například pro barevný obr. 9 tak dostaneme histogramy znázorněné na obr. 10, 11 a 12.

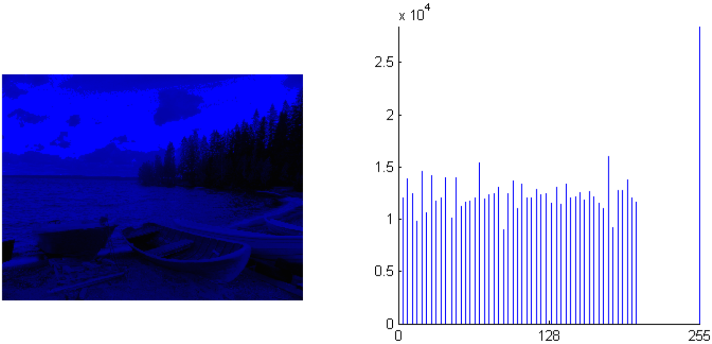
Obr. 9 Fotografie pořízená autorem



Obr. 10 Červená složka barevného obr. 9 a její histogram h_{R}



Obr. 11 Zelená složka barevného obr. 9 a její histogram h_{G}



Obr. 12 Modrá složka barevného obr. 9 a její histogram h_{B}

Právě uvedený přístup k reprezentaci histogramů má však jednu poměrně podstatnou nevýhodu. Předpokládejme, že o jistém barevném obrázku známe tyto hodnoty histogramů:

$$h_{\mathrm{R}}(100) = 250,$$
$$h_{\mathrm{G}}(100) = 250,$$
$$h_{\mathrm{B}}(100) = 250.$$

Kolikrát je v daném obrázku zastoupená barva, která má v RGB krychli souřadnice $\langle 100, 100, 100\rangle$? Je důležité si uvědomit, že barva $\langle 100, 100, 100\rangle$ se v obrázku vyskytuje *nejvýše* 250krát – nemusí se proto v obrázku vyskytovat vůbec. Nevýhoda tak spočívá v tom, že četnost barvy v RGB modelu nemůžeme z dílčích histogramů stanovit přesně.

6. Automatická úprava kontrastu

Na začátku tohoto dílu jsme si ukázali, jak vypadá transformační funkce f realizující zvětšení kontrastu šedotónového obrázku. Zopakujme, že je tato funkce dána předpisem

$$f(a) = \begin{cases} 0 & \text{pro } a < a_0; \\ 255\cdot \frac{a-a_0}{a_1-a_0} & \text{pro } a_0 \leq a \leq a_1; \\ 255 & \text{pro } a > a_1, \end{cases}$$

kde $a \in \{0, 1, \ldots, K-1\}$. Řekli jsme si také, že se hodnoty a_0 a a_1 nastavují na hodnotu nejtmavějšího a nejsvětlejšího stupně šedi přítomného ve vstupním obrázku. Tyto hodnoty je tedy možné snadno odečíst z histogramu – hodnota a_0 je rovna pozici sloupečku s nenulovou výškou, který je v histogramu nejvíce nalevo; naopak a_1 je rovna pozici sloupečku s nenulovou výškou, který je v histogramu nejvíce napravo. Formálně zapsáno:

$$\begin{aligned} a_0 &= \min\{a \mid h(a) \neq 0\}, \\ a_1 &= \max\{a \mid h(a) \neq 0\}. \end{aligned}$$

Nyní máme k dispozici jednoduché vzorce pro výpočet hodnot a_0 a a_1 a úprava kontrastu tak může být provedena bez jakéhokoliv zásahu uživatele. Výše uvedená transformační funkce f je proto často nazývána *automatickou úpravou kontrastu.*

Tento díl seriálu o počítačové grafice se věnoval základním úpravám obrazu v prostorové doméně. Všechny tyto úpravy měly společnou jednu vlastnost: bylo možné je realizovat aplikací transformační funkce f na pixely vstupního obrázku. Obrazová funkce o' výsledného obrázku tak vznikne složením obrazové funkce o vstupního obrázku a transformační funkce f. Platí tedy:

$$o'(x, y) = f(o(x, y))$$

pro všechna $x \in \{0, 1, \ldots, m-1\}$ a $y \in \{0, 1, \ldots, n-1\}$.

Je patrné, že zmíněné úpravy berou v potaz pouze hodnotu upravovaného pixelu (pixelu o souřadnicích x a y) a nijak nezohledňují hodnoty některých jiných pixelů. V praxi je však občas potřebné provést úpravy, které z podstaty věci musí zohlednit hodnoty některý dalších pixelů; většinou se jedná o pixely, které jsou v bezprostředním okolí upravovaného pixelu. Mezi tyto úpravy patří například vyhlazení nebo naopak vyostření obrázku. Zabývat se jimi budeme v dalším pokračování seriálu.

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.