

KÓDOVANIE OBRÁZU

Hovorí sa, že raz vidieť je lepšie ako sto krát počuť. Preto je grafická informácia niekedy dôležitejšia ako text. Čo sa týka spôsobu, akým uchovať obrazovú informáciu v počítači, opäť môžeme povedať, že neexistuje univerzálne riešenie tohto problému. Dá sa však povedať, že existujú dva rôzne pohľady na tento problém. Prvým z nich je pozeranie sa na obrázok ako na sieť (**raster**) veľmi malých štvorcov - **pixelov**. Uloženie grafickej informácie pomocou takéhoto prístupu voláme **rastrová grafika**. Druhý prístup je pozeranie sa na obrázok ako na zoskupenie objektov (alebo ich častí), ktoré sa dajú nakresliť pomocou matematických vzorcov a funkcií. Tieto objekty majú svoje vlastnosti ako polohu na obrázku, veľkosť, farbu, priehľadnosť povrchu, lesklosť povrchu... Tieto vlastnosti sú vstupnými parametrami (**vektormi**) matematických vzorcov a funkcií, pomocou ktorých sa objekty nakreslia. Uloženie grafickej informácie pomocou takéhoto prístupu sa nazýva **vektorová grafika**.

Rastrová grafika

Rastrové obrázky sú rozdelené na sieť myslených štvorčekov - **pixelov**. Rozmer každého obrázka je pre počítač počet pixelov na šírku x počet pixelov na výšku. Pre každý pixel (štvorček) je nutné okrem polohy (riadok a stĺpec) zakódovať aj farbu, resp. ďalšie parametre.

Ak je obrázok monochromatický (čierna a biela farba), kódovanie je jednoduché (1 - rozsvietený (biely) bod, 0 - nerozsvietený (čierny) bod), napr.

```
100111001
011010110
101101101
110111011
111010111
111101111
```

A teda zápis v riadku vyzerá : 011000110 100101001 010010010 001000100 000101000 000010000, čo je 54 bitov.

Pri niektorých obrázkoch je však potrebné zaznamenať i odtiene, preto dve farby - čierna a biela - nestačia. Výhodné je zakódovať odtiene sivej farby pomocou ôsmich bitov tak, aby informácia o každom bode zaberala 1 B, t.j. jedno pamäťové miesto počítača. Čím bude bod svetlejší, tým väčšia hodnota sa do pamäte uloží. Preto bod čiernej farby bude uložený ako 0 a bod bielej farby ako maximálna možná hodnota - 255. Obrázok teda bude kódovaný nasledovne:

```
192  0  0 255 255 192  0  0 255
 0 255 192  0 192  0 255 192  0
192  0 255 192  0 255 192  0 255
255 192  0 255 255 192  0 255 255
255 255 192  0 192  0 255 255 255
255 255 255 192  0 255 255 255 255
```

Obrázok bude v pamäti počítača zaberat' 54 Bajtov (pamäťových miest).

Vektorová grafika

Vektorové obrázky sú tvorené nie skupinou štvorčekov, ale základnými objektmi, ktorých algoritmy zostrojenia sú vopred známe, len treba zaznamenať (zakódovať) ich základné charakteristiky (napr. pre kružnicu súradnice stredu, polomer, príp. farba čiary napr. : K40Ž200150 - kružnica s polomerom 40 žltej farby a so súradnicami stredu 200 150).

Kódovanie farieb

Aby bol obrázok farebný, musíme každému pixelu obrázka priradiť určitú farbu. Spôsobov, akými sa to dá urobiť je opäť niekoľko. Na to, aby sme to vedeli urobiť, musíme vedieť niečo o ľudskom oku a spôsobe videnia farieb. Farby vnímame vďaka svetlu, ktoré vstupuje do nášho oka a dopadá na bunky - čapíky, ktoré sú citlivé na farbu. Svetlo je vlastne elektromagnetické vlnenie s rôznou vlnovou dĺžkou. Ľudské oko dokáže vnímať iba určitý rozsah vlnových dĺžok svetla. Svetlo s najväčšou vlnovou dĺžkou, ktoré ľudské oko vníma, je červenej farby (780 nm) a svetlo s najnižšou vlnovou dĺžkou je svetlo fialovej farby (380nm). V tomto spektre farieb je ľudské oko schopné bez problémov rozlíšiť vyše 1,5 milióna rôznych farieb. Nie však každej farbe, ktorú rozlíšime pripadá iná vlnová dĺžka, pretože podľa intenzity dopadajúceho svetla, ľudské oko rozlišuje sýtosť a jas daného odtieňa farby. Takže iná vlnová dĺžka prislúcha iba rôznym odtieňom farieb. Prvé riešenie problému kódovania farieb, ktoré sa ponúka, je zobrať celý rozsah vlnových dĺžok a rozdeliť ho na niekoľko častí a do pamäte počítača uložiť poradové číslo farby, ktorej prislúcha určitá vlnová dĺžka, ďalej jej sýtosť a jas. Kódovanie farieb takýmto spôsobom využívajú **farebné modely HSB, HLS**. Oba modely uchovávajú informáciu o odtieni (**Hue**) a sýtosti (**Saturation**). Tieto modely sa odlišujú iba v tom, že prvý model používa jas (**Brightness**) a druhý používa svetlosť (**Lightness**). V prvom modeli dostaneme čiernu farbu tým, že nastavíme jas na nulu a bielu tak, že jas nastavíme na maximálnu hodnotu (nezávisle od sýtosti). V druhom modeli dosiahneme čiernu, ak je svetlosť aj sýtosť 0 a bielu, ak je svetlosť aj sýtosť maximálna. Hlavným nedostatkom tohto problému je náročnosť výroby svetla so zadanou vlnovou dĺžkou. Našťastie existujú aj iné riešenia problému kódovania farieb.

Farebný model RGB je najčastejšie využívané kódovanie farby bodu obrázka. Empiricky sa zistilo, že takmer všetky farby sa dajú vytvoriť zmiešaním ľubovoľných troch nezávislých farieb (t.j. že pomocou zmiešania dvoch nedostaneme tretiu). Najvýhodnejšie pre výrobu svetelných lúčov bolo použitie farieb červená (**Red**), zelená (**Green**), modrá (**Blue**). Aby sme vedeli vytvoriť 1,5 milióna farieb stačí, ak každú z týchto farieb rozdelíme na 115 odtieňov, ktorých zmiešaním v rôznych pomeroch vzniknú všetky farby. Kvôli uchovaniu v pamäti počítača, je však výhodnejšie použiť až 256 odtieňov každej farby (8 bitov), čo nám umožní vytvoriť až 16 777 216 rôznych farieb. Pri takomto kódovaní je každá farba zakódovaná 24 bitmi, čo sú tri pamäťové miesta počítača. Pričom 255 0 0 je sýta červená farba, 0 255 0 je sýta zelená farba, 0 0 255 je sýta modrá farba, 0 0 0 je čierna farba a 255 255 255 je biela farba. Obrázok srdca môžeme týmto spôsobom zakódovať napríklad aj takto (kvôli kratšiemu zápisu použijeme šestnástkovú sústavu):

```
C0 C0 C0 FF 00 00 FF 00 00 FF FF FF FF FF FF FF C0 C0 C0 FF 00 00 FF 00 00 FF FF FF
FF 00 00 FF 00 00 FF 00 00 FF 00 00 C0 C0 C0 FF 00 00 FF 00 00 FF 00 00 FF 00 00
C0 C0 C0 FF 00 00 FF 00 00 FF 00 00 FF 00 00 FF 00 00 FF 00 00 FF 00 00 FF 00 00
FF FF FF C0 C0 C0 FF 00 00 FF 00 00 FF 00 00 FF 00 00 FF 00 00 FF 00 00 FF 00 00
FF FF FF FF FF FF C0 C0 C0 FF 00 00 FF 00 00 FF 00 00 FF 00 00 FF 00 00 FF 00 00
FF FF FF FF FF FF FF FF FF C0 C0 C0 FF 00 00 FF 00 00 FF 00 00 FF 00 00 FF 00 00
```

Obrázok bude v pamäti počítača zaberat' 162 Bajtov (pamäťových miest).

Farebný model CMY je model, ktorý používa doplnkové farby. V modeli RGB platilo, že ak zmiešame všetky tri základné farby s maximálnou sýtosťou, dostaneme bielu farbu. Takýto spôsob je výhodný pri obrazovkách monitorov, pretože tienidlo je čierne. Pri tlačiarňach sa však tlačí na biely papier, preto potrebujeme vziať také farby, pri ktorých, ak zmiešame ich najsytejšie odtiene, dostaneme čiernu farbu. Takéto farby dostaneme, keď zoberieme doplnkové farby k farbám červená, zelená a modrá. Týmto farbami sú azúrová (**Cyan**), purpurová (**Magenta**) a žltá (**Yellow**). Kvôli tomu, že je lacnejšie vyrobiť čierny atrament ako ho miešať pomocou týchto troch farieb, sa k týmto farbám pridáva i samostatná čierna farba a tento model sa označuje tiež CMYK, kde posledné písmeno je odvodené od black - čierna. Výhodou tohto formátu je tá, že výsledná farba CMY sa dá veľmi jednoducho získať z modelu RGB pomocou vzorcov:

$$C = (255 - R); M = (255 - G); Y = (255 - B)$$

Farebný model YUV slúži na zachovanie čiernobielej informácie pri televíznom vysielaní. Po vzniku farebného filmu nastal problém, ako zakódovať farbu tak, aby farebné filmy mohli pozerat' i ľudia s čiernobielymi prijímačmi. Bolo potrebné zachovať pôvodnú čiernobielu informáciu a doplniť ju tak, aby vznikol farebný obraz. Farba je kódovaná tak, že k čiernobielej zložke Y, ktorú tiež nazývame svietivosť (luminance), pridáme dve farebné zložky UV farebnosti (chrominance), pričom zložka U udáva odtieň medzi modrou a žltou a zložka V udáva odtieň medzi červenou a žltou farbou. Takže čiernobiely prijímač berie do úvahy iba zložku Y, farebné prijímače za pomoci zložiek YUV získajú RGB kód farby pomocou jednoduchej transformácie:

$$R = Y + 1,403V; G = Y - 0,344U - 0,714V; B = Y + 1,770U$$

miniaplikácia na kódy farieb: <http://www.dgx.cz/tools/colormixer/stripe.php>

Grafické formáty

Všetky spomenuté farebné modely kódujú farbu troma nezávislými hodnotami. Najvýhodnejšie je teda každý pixel obrazu zakódovať pomocou troch pamäťových miest počítača - 3 Bajtov. Obrázok s rozmermi 1024x768 pixelov tak v pamäti grafickej karty zaberie 2 359 296 Bajtov. V minulosti kvôli cene pamätí sa na grafické karty montovali pamäte menších rozmerov, teda na kódovanie farieb sa použil menší počet Bajtov. Najstaršie počítače používali iba 16 farieb, to znamená, že každý bod bol zakódovaný 4 bitmi, neskôr sa začali vyrábať 256-farebné grafické karty (VGA), ktoré mali každý bod kódované 8 bitmi. Po zlacnení počítačových pamätí už bolo možné vyrábať karty SVGA, ktoré kodovali farby pomocou 16 bitov (dve pamäťové miesta) v režime **High Color** (vysoká farebnosť). V súčasnosti už grafické karty majú toľko pamäte, že bez problémov môžu kódovať každý bod 24 bitmi (tri pamäťové miesta) v režime **True Color** (pravá farebnosť). Popísaný spôsob uloženia obrázka (keď je každý bod kódovaný pomocou niekoľkých bitov - 4, 8, 16 alebo 24) sa používa formát, ktorý sa volá **bitová mapa (BitMaP)**. Obrázky v takomto formáte sú v počítači uložené v súboroch s príponou BMP.

Veľká pamäťová náročnosť už teda nie je problém grafickej karty počítača, stále je však problém pri posielaní takýchto obrázkov prostredníctvom internetu, pretože obrázok s rozmermi 1024x768 pixelov v režime true color sa nezmesť ani na disketu. Preto sa na zníženie pamäťových nárokov používa **paleta farieb a kompresia** (stlačenie).

Paleta využíva skutočnosť, že na kreslených obrázkoch väčšinou nie je použitých viac ako 256 farieb. Zníženie pamäťových nárokov spočíva v tom, že očísľujeme všetky použité farby v obrázku číslami od 0 do 255, a potom kódujeme každý bod tak, že uvedieme poradové číslo farby v palete. Tým miesto troch pamäťových miest, každý bod zakódujeme len pomocou jedného pamäťového miesta. Naše "srdce" teda zakódujeme nasledovne:

Paleta:

00 00 00 00 , **01 C0 C0 C0**, **02 FF 00 00**, **03 00 FF 00**, **04 00 00 FF**, **05 FF FF FF**

```
01 02 02 05 05 01 02 02 05
02 02 02 02 01 02 02 02 02
01 02 02 02 02 02 02 02 00
05 01 02 02 02 02 02 00 04
05 05 01 02 02 02 00 04 03
05 05 05 01 02 00 04 03 03
```

Ak spočítame počet pamäťových miest, dostaneme 24 B použitých v palete a 54 B použitých na obrázok, čo je spolu 78 B (bez použitia palety to bolo 162 B).

Ďalší spôsob ako znížiť pamäťovú náročnosť obrázka, je použitie **kompresie**. Princíp kompresie spočíva v tom, že ak sa pixel s rovnakou farbou vyskytuje viac krát za sebou, do pamäte neukladáme jednotlivé pixely, ale uložíme koľko krát sa pixel danej farby vyskytol. Nap. 8 x červená, 3 x modrá ... (toto je veľmi zjednodušene povedané, v skutočnosti sa to robí oveľa rafinovanejšie, ale princíp je rovnaký). Naše srdce by sme takýmto spôsobom mohli zakódovať takto:

```
01 01 02 02 02 05 01 01 02 02 01 05
04 02 01 01 04 02
01 01 07 02 01 00
01 05 01 01 05 02 01 00 01 04
02 05 01 01 03 02 01 00 01 04 01 03
03 05 01 01 01 02 01 00 01 04 02 03
```

Všimnite si, že v druhom a treťom riadku sme ušetrili po 3B, naopak v ostatných riadkoch tento spôsob nebol výhodný, pretože sme dostali nejaké bajty navyše. Takže v konečnom dôsledku by sme zabrali ešte viac pamäte ako pôvodný obrázok. To, či sa má obrázok do pamäte uložiť skomprimovane alebo nie, sa rozhodneme pomocou tzv. Kompresného pomeru (veľkosť po komprimovaní / pôvodná veľkosť). Ak je kompresný pomer < 1, obrázok uložíme komprimovaný, v opačnom prípade ho uložíme v normálnej podobe.

Oba popísané spôsoby zníženia pamäťovej náročnosti používa **formát GIF**. Okrem týchto výhod, formát gif umožňuje označiť jednu z variieb na obrázku za priehľadnú.

256 farebný obrázok	vo formáte BMP	Rovnaký obrázok	vo formáte GIF
32 230 B		13 396 B	

Ďalšou obrovskou výhodou tohto formátu je to, že dokáže uložiť animácie obrázkov.

Naopak nevýhodou tohto formátu je obmedzenie 256 farbami.

Kompresiu využíva i **formát JPEG** (súbory s príponou JPG), tentokrát sa však jedná o tzv. **stratovú kompresiu**. Táto kompresia je založená na vynechávaní, niektorých málo viditeľných detailov. V praxi to znamená, že ak je niekde napríklad jedna svetložltá bodka uprostred veľkého bieleho poľa, jednoducho sa vymaže. Ďalej ak je niekde tenká čiara medzi dvoma plochami, tak sa farba tejto čiary upraví tak, aby sa jej farba dala vypočítať zložením farieb plôch, ktoré obklopuje.

Po aplikovaní takýchto úprav sa aplikuje klasická kompresia, ktorá je tým pádom oveľa efektívnejšia. Výhodou tohto formátu je to, že používa všetky farby (true color) a dosahuje výborný kompresný pomer (zmenšenie). Je vhodný najmä na uloženie digitálnych fotografií.

napr. BMP formát (160 KB), a ten istý obr. JPEG formát (14,5 KB)

Nevýhodou tohto formátu je to, že sa nedá nastaviť priehľadná farba.

Problém s priehľadnou farbou rieši **formát PNG**, ktorý spája výhody formátov GIF - priehľadná farba, bezstratová kompresia a JPEG - True Color farby, dobrý kompresný pomer.

Nevýhodou tohto formátu je snáď len to, že je málo rozšírený.

Existuje mnoho ďalších formátov, výhodou formátov GIF, JPEG a PNG je však to, že ich podporujú internetové prehliadače, takže ich môžeme vkladať do webových stránok.

Programy na spracovanie grafickej informácie

Programy na spracovanie grafiky delíme do troch skupín na rastrové (niekedy tiež bitmapové) editory, vektorové editory a univerzálne prehliadače.

Medzi **rastrové editory** patria programy na spracovanie fotografií alebo na vytváranie kreslených obrázkov či animácií. Medzi najjednoduchšie rastrové editory patrí napríklad i program Maľovanie (Paint), ktorý je súčasťou operačného systému Windows. Vyspelejším editorom je grafický editor GIMP, ktorý je zadarmo. Z českých grafických editorov je známy editor Photo Studio od firmy Zoner.

Najznámejšie komerčné editory sú PhotoShop od firmy Adobe, Paint Shop Pro od firmy Corel a Fireworks od firmy Macromedia.

Najznámejšie **vektorové editory** sú Corel Draw od firmy Corel, Illustrator od firmy Adobe a z českých je známy Callisto od firmy Zoner, ktorým tiež hovoríme DTP (Desktop Publishing) - programy na tvorbu plagátov, novín a pod.

Medzi vektorové editory, ktoré umožňujú vytvárať animácie, patrí editor Flash od firmy Macromedia a medzi vektorové editory patria tiež **3D modelovacie nástroje**, z ktorých je najznámejšie 3D studio MAX.

Z 3D modelovacích programov, ktoré sú zadarmo, sú známe Blender a Maya.

Univerzálne prehliadače pôvodne slúžili iba na prehliadanie čo najväčšieho počtu grafických formátov, dnes však obsahujú i niektoré základné funkcie grafických editorov ako orezávanie, zmena veľkosti, odstránenie efektu červených očí, zmena farebnosti, jas a kontrastu atď. Možno ich tiež využiť na prevod grafických informácií z jedného formátu do druhého. Najznámejšími prehliadačmi sú IrfanView, ACDsee a XnView.

Úloha 1

Na digitálnom fotoaparáte máme nastavené rozlíšenie, pri ktorom sa fotografie ukladajú ako bitmapové súbory s veľkosťou 600 KB s 24 bitovým farebným kódovaním. Ak prekonvertujeme takúto fotografiu do 256 farieb, akú bude mať približne veľkosť?
(A) 6 000 KB **(B)** 400 KB **(C)** 200 KB **(D)** 150 KB

Riešenie

Najprv musíme zistiť koľko bodov sa nachádza v obrázku. Vieme, že jeden bod obrázka je kódovaný 24 b, zaberá teda tri pamäťové miesta teda 3 Bajty ($24:8=3$). Ďalej vieme, že 1 kB = 1024 B, takže obrázok bude zaberat':

$$600 \text{ kB} \times 1024 = 614\,400 \text{ B}$$

Teraz zistíme, koľko bodov sa nachádza v obrázku:

$$\frac{614\,400 \text{ B}}{3 \text{ B}} = 204\,800$$

veľkosť obrázka veľkosť bodu počet bodov

Teraz potrebujeme zistiť, koľkými bitmi sa dá zakódovať jeden bod 256 farebného obrázka, a tiež koľko pamäte takýto bod zaberie. Ak náhodou nevieme z pamäti, že $256 = 2^8$ takže je to 8 bitov, a to zodpovedá 1 B pamäte, môžeme počet bitov vypočítať takto:

$$\log_2 256 = \ln 256 : \ln 2 = 8$$

Veľkosť 256 farebného obrázka teda bude:

$$204\,800 \times 1 \text{ B} = 204\,800 \text{ B}$$

počet bodov veľkosť bodu veľkosť obrázka

Teraz už len prevedieme výsledok na kB:

$$204\,800 \text{ B} : 1024 = 200 \text{ kB}$$

Správna odpoveď je **(C)**.

Úloha 2

Katka má fotografiu triedy uloženú v bitmapovom súbore a chce ju poslať MMS-kou na Petrov mobilný telefón. Fotka má rozmery 256 x 180 pixelov (obrazových bodov) a je v nej použitých 256 farieb. MMS-ka môže mať maximálne 5 KB. Najmenej koľko MMS musí Katka poslať, ak chce poslať celú fotografiu rozloženú do viacerých MMS?

Riešenie

Rozmery fotografie sú 256 x 180 pixelov, teda počet bodov v obrázku bude:

$$256 \times 180 = 46\,080$$

Každý bod 256 farebného obrázka zaberie 1 B pamäte (pozri úlohu 1), takže obrázok bude zaberat':

$$46\,080 \times 1 \text{ B} = 46\,080 \text{ B}$$

počet bodov veľkosť bodu veľkosť obrázka

Jedna MMS môže zaberat' 5 kB čo je v B:

$$5 \text{ kB} \times 1024 = 5\,120 \text{ B}$$

Počet MMS vypočítame takto:

$$\frac{46\,080 \text{ B}}{5\,120 \text{ B}} = 9$$

veľkosť obrázka veľkosť MMS počet MMS

Potrebný počet MMS je 9.

Úloha 3

Na webovej stránke chceme publikovať animovanú fotografiu. Dve skoro rovnaké fotografie sa líšia iba mrknutím oka na jednej z fotografií. Vytvoríme z nich jediný súbor - animovaný obrázok. Ktorý z uvedených formátov použijeme?
(A) gif (B) bmp (C) jpeg (D) ani

Riešenie

Správna odpoveď je (A) (pozri formát GIF).

Úloha 4

Obrázok v pamäti je zložený zo 600 x 800 farebných bodov. Najmenej koľko KB pamäte zaberie tento 6-farebný neskompimovaný obrázok?
(A) 600 × 800 : (8 × 1024) kB
(B) 600 × 800 × 3 : (8 × 1024) kB
(C) 600 × 800 × 6 : (8 × 1024) kB
(D) 600 × 800 : 1024 kB

Riešenie

Obrázok pozostáva z 600 x 800 bodov. V obrázku je použitých 6 farieb, teda na zakódovanie jedného bodu budeme potrebovať nasledovný počet bitov:

$\log_2 6 = \ln 6 : \ln 2 = 2,58$ výsledok presiahol celú hodnotu, musíme teda použiť 3 bity

Ak chceme zistiť, koľko takýto obrázok zaberie pamäte, musíme deliť 8, pretože jedno pamäťové miesto (1 B) pozostáva s 8 bitov:

$$800 \times 600 \times 3 : 8 \text{ B}$$

Výsledok je potrebné ešte previesť na kB, teda vydeliť 1024:

$$800 \times 600 \times 3 : 8 : 1024 \text{ kB} = 800 \times 600 \times 3 : (8 \times 1024) \text{ kB}$$

Správna odpoveď je (B).