

# ZOBRAZOVÁNÍ BAREV PŘI OMEZENÉ PALETĚ

Jiří Ž á r a

Katedra počítačů FEL ČVUT, Karlovo nám. 13, 121 35 Praha 2

zara@cs.felk.cvut.cs

## Úvod

Současná počítačová grafika umožňuje vytvářet obrázky prostorových scén, které díky metodám sledování paprsku a radiálním metodám působí velice přirozeným a realistickým dojmem. Výpočty barevných odstínů jsou prováděny s přesností, která však u běžných barevných monitorů přesahuje technické možnosti při zobrazení spočítaných jemných barevných odstínů. V počítačové grafice se totiž používá mnoha zařízení, která mají jen omezenou škálu zobrazitelných odstínů. Běžné obrazové adaptéry osobních počítačů dokáží zobrazit jen 16 barevných odstínů, v případě adaptérů VGA 256 odstínů. Zobrazované odstíny je sice možno vybrat z palety několika tisíc možných barev, na obrazovce jich však současně může být jen omezený počet. Podobná situace nastává i u levnějších a středních modelů grafických pracovních stanic. Úplným extrémem jsou pak černobílé tiskárny, kde je paleta barev tvořena pouze dvěma zástupci — bílou a černou.

K překlenutí rozdílu mezi vypočítanými a zobrazovanými barevnými odstíny slouží tzv. rozptylovací metody, jejichž rozboru a popisu je věnován tento příspěvek.

Dále popsané postupy budou použitelné jak pro vykreslování monochromatického obrazu (v odstínech šedi), tak pro zpracování barevných informací na různých obrazových adaptérech osobních počítačů. Z praktického hlediska budou algoritmy zaměřeny na nejčastěji používaný adaptér VGA.

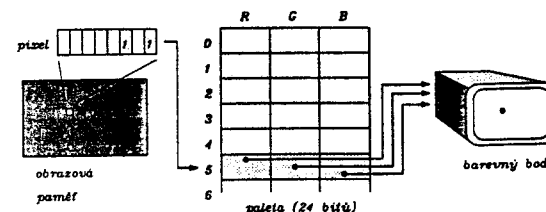
## 1 Technické zázemí

Běžné obrazové adaptéry nabízejí standardně jen 16 různých barev, v nestandardním režimu adaptéru VGA lze potom docílit 256 různých barevných kombinací v rozlišení, které závisí na velikosti obrazové paměti (640 × 400 pixelů pro 256 kB, 800 × 600 pixelů pro 512 kB a 1024 × 768 pixelů pro 1 MB obrazové RAM).

Ačkoliv se 256 barevných odstínů jeví ve srovnání s dřívějšími šestnácti jako dostačující množství, pro tvorbu realistických obrázků je to stále ještě málo. Lidské oko, přes všechny své nedokonalosti, dokáže rozlišit několik tisíc barevných odstínů a je citlivé zejména na postupnou změnu barev, tak typickou pro obrázky osvětlených oblých povrchů. Za standard, postačující pro kvalitní barevnou rastrovou grafiku, je proto ve světě považováno kódování barev pomocí 24-bitové informace, která je složena ze 3 bytů pro jednotlivé složky [R, G, B]. V tomto způsobu kódování je tedy možné definovat 16 milionů barevných kombinací, pro něž je používán název true colors. Větší množství bitů se pro kódování barev nepoužívá, uvedených 256 stupňů intenzity každé barevné složky poskytuje dostatečně jemnou diskretizaci jasů na monitoru počítače.

Při použití 24 bitů na barvu musí být obrazový adaptér vybaven velkou obrazovou pamětí. Pro poměrně běžné rozlišení 1024 × 768 pixelů je potřeba 3 MB RAM, což je pro oblast

osobních počítačů poněkud vysoká hodnota. Na obranu osobních počítačů budíž řečeno, že ani 32-bitové grafické pracovní stanice (MikroVAX, SUN, HP) nejsou v běžných konfiguracích vybavovány adaptérem, který používá 24 bitů na pixel. Použití takto kvalitních zařízení je většinou spojováno s přítomností grafického akcelerátoru a je tedy také patřičně drahé. Zajímavá je však skutečnost, že adaptér s 24 bity na pixel se objevuje v nabídce konfigurací počítačů Apple Macintosh.



Obr 1: Obrazový adaptér s paletou barev

Grafické adaptéry, které nepoužívají plných 24 bitů na pixel, zpracovávají barevnou informaci pomocí palety (obr. 1). Data, uložená v obrazové paměti v typickém rozsahu 8 bitů na pixel, představují index do palety o 256 položkách. Každá položka je definována jako kombinace složek [R, G, B]. Pro složku je v adaptéru VGA vymezeno 6 bitů, v kvalitnějších adaptérech 8 bitů.

## 2 Rozptylovací metody

Metody, které dokáží z několika barev vytvořit iluzi bohaté barevné škály, se nazývají polotónování<sup>1</sup> a rozptylování<sup>2</sup>. Pojem polotónování je používán především u tiskáren, kde je jeden barevný pixel původního obrázku převeden na matici bodů s omezenou velikostí palety. Dochází tedy ke zvětšení obrázku. Rozptylovací metody jsou vhodné pro zobrazení obrázku, který byl např. vypočítán v kvalitě 24 bitů na pixel a který má být vykreslen na obrazovce v nezměněné velikosti. V této kapitole se zaměříme především na techniku rozptylování — polotónování budeme chápat jako její speciální případ.

Obě metody využívají schopnosti lidského oka vytvářet z několika blízkých barevných paprsků vjem jediného paprsku, jehož barva je dána aditivním složením barev paprsků původních. Tato vlastnost je základem všech metod počítačové grafiky pro generování obrázků o mnoha odstínech na zařízeních s omezenou velikostí palety. Ostatně je i principem barevné televizní obrazovky.

Rozptylovací techniky jsou používány pro zpracování jak černobílých, tak barevných obrázků. V dalším textu vysvětlíme nejprve základní principy na černobílých obrázcích, později se zaměříme na v praxi častou situaci zobrazování 24-bitových barevných informací na barevném displeji s 8 bity na pixel a nastavitelnou paletou barev.

<sup>1</sup>angl. halftoning

<sup>2</sup>angl. dithering

Předpokládejme nejprve, že chceme obrázek obsahující 16 ( $= 2^4$ ) odstínů šedi vykreslit na displeji, který zobrazuje pouze černou a bílou. Původní 4 bity na pixel je nutno zredukovat na 1 bit na pixel obrázku tak, aby si lidské oko dokázalo z různých kombinací sousedních bílých a černých bodů vytvořit představu několika odstínů šedi.

## 2.1 Náhodné rozptýlení

Označme  $C_{in}$  vstupní intenzitu jasu (odstínu šedi) jednoho obrazového bodu. Její hodnota se pohybuje v intervalu  $(0, 15)$ . Obdobně označíme výstupní intenzitu  $C_{out} \in (0, 1)$ . Při použití metody náhodného rozptýlení využijeme generátoru náhodných čísel, jehož hodnoty vrací funkce **random**.

Pro určení výsledné intenzity  $C_{out}$  černobílého obrazového bodu použijeme algoritmus:

```
Cout := 0;
if Cin > random(15) then inc(Cout);
```

přičemž předpokládáme, že funkce **random(15)** vrací nezáporné hodnoty menší než 15. Je-li potom vstupní intenzita nulová, je výsledná intenzita také nulová. Má-li vstupní intenzita nejvyšší možnou hodnotu 15, nabude výstupní intenzita vždy své nejvyšší hodnoty 1. V ostatních případech rozhodne o tom, zda výsledný pixel bude rozsvícen nebo ne, výsledek porovnání velikosti vstupní intenzity a náhodně vygenerovaného čísla. To se ve výsledném obrázku projeví tak, že například plocha o vstupní intenzitě 10 bude tvořena nepravidelně se střídajícími bílými a černými pixely. Poměr počtu bílých pixelů k černým bude přitom dvě třetiny, jak to odpovídá poměru 10/15. Tímto způsobem zůstanou zachovány původní jasové poměry v celém obrázku.

Náhodné rozptýlení je rychlá a jednoduchá metoda zachovávající původní velikost obrázku. Dodává větším plochám vzhled drsného povrchu, výsledné obrázky působí dojmem fotografie s velkým zrněním.

## 2.2 Maticové rozptýlení

Maticová rozptylovací metoda používá pro generování různých odstínů šedi pravidelných, předem daných vzorků složených z bílých a černých bodů. Pokud jsou tyto vzorky dobře navrženy, působí výsledný obrázek jako kresba, ve které malíř tužkou vystínoval tmavší, či světlejší místa pravidelnými jemnými šrafkami. Původně byla tato metoda navržena pro takový převod obrázků, při kterém je jeden vstupní pixel nahrazen skupinou (maticí) pixelů výsledných. Dojde tedy ke zvětšení obrázku. Tento způsob se nazývá polotónování.

Na obr. 2 vidíme příklad vzorků, které by byly použity v případě, že vstupní intenzita se pohybuje v rozsahu  $(0, 4)$ . Pro  $C_{in} = 0$  použijeme levý vzorek, pro  $C_{in} = 4$  vzorek nejvíce vpravo. Pro větší škálu vstupních intenzit je nutno použít matice vyšších řádů.

Vždy je třeba dodržet zásadu, že každá následující matice vznikla z předchozí přidáním jedničky na jednu novou pozici. Je tedy nutno vyvarovat se permutací uvnitř matice, které by v případě některých kombinací vstupních odstínů vedly k nesprávnému zesvětlení, či ztmavení obrázku. Při uvedené zásadě nemusíme ani udržovat v paměti všechny matice pro různé hodnoty  $C_{in}$  a můžeme použít matici jedinou (příklady jsou na obr. 3). Hodnoty v takové

$$\begin{array}{c|c|c|c|c} \left| \begin{array}{cc} 0 & 0 \\ 0 & 0 \end{array} \right| & \left| \begin{array}{cc} 0 & 0 \\ 0 & 1 \end{array} \right| & \left| \begin{array}{cc} 0 & 0 \\ 1 & 1 \end{array} \right| & \left| \begin{array}{cc} 0 & 1 \\ 1 & 1 \end{array} \right| & \left| \begin{array}{cc} 1 & 1 \\ 1 & 1 \end{array} \right| \\ C_{in} = 0 & C_{in} = 1 & C_{in} = 2 & C_{in} = 3 & C_{in} = 4 \end{array}$$

Obr 2: Příklady rozptylovacích matic

$$M_d = \begin{bmatrix} 0 & 12 & 3 & 15 \\ 8 & 4 & 11 & 7 \\ 2 & 14 & 1 & 13 \\ 10 & 6 & 9 & 5 \end{bmatrix}, M_p = \begin{bmatrix} 1 & 5 & 9 & 2 \\ 8 & 12 & 13 & 6 \\ 4 & 15 & 14 & 10 \\ 0 & 11 & 7 & 3 \end{bmatrix}$$

Obr 3: Různé rozptylovací matice  $4 \times 4$  pro  $C_{in} \in (0, 15)$

rozptylovací matici<sup>3</sup> určují, při jaké velikosti  $C_{in}$  se na výstupu objeví v odpovídajícím místě jednička. Pokud je  $C_{in}$  větší než daná hodnota v matici, bude výstupní pixel rozsvícen.

Je zřejmé, že jedničky a nuly lze v rozptylovacích maticích uspořádat mnoha způsoby. To je samozřejmě možné a pro různé typy zobrazovacích zařízení se skutečně používají různá uspořádání matic ([Holl80]). V příkladu na obr. 3 vidíme matice  $M_d$  vhodnou pro displeje, která vytváří „křížkový“ stínovací vzor. Jiné uspořádání je vhodné pro tiskárny, kde je často žádoucí „puntíkový“ vzor, připomínající fotografie z novin, složené z větších a menších teček (matice  $M_p$  na obr. 3).

V některých případech potřebujeme rozptylovací matice vyšších řádů. Ty lze generovat algoritmicky v případě, že řád rozptylovací matice  $M$  je mocnina 2:

$$\text{Nechť } M_{(2,2)} = \begin{bmatrix} 0 & 3 \\ 2 & 1 \end{bmatrix}$$

$$\text{Pak } M_{(4,4)} = \begin{bmatrix} M_0 & M_3 \\ M_2 & M_1 \end{bmatrix}, \text{ kde } M_i = 4 * M_{(2,2)} + i.$$

Metodu maticového rozptýlení lze úspěšně použít i v případě, kdy nezvětšujeme výsledný obrázek. Tehdy použijeme místo celé matice pouze jeden její prvek. Pro výběr prvku poslouží souřadnice  $(x, y)$  zobrazovaného pixelu, které po použití funkce **mod** určí indexy v matici. Vzorec pro určení, zda v daném místě bude mít bod bílou nebo černou barvu, bude (v případě indexování rozptylovací matice od nuly) potom:

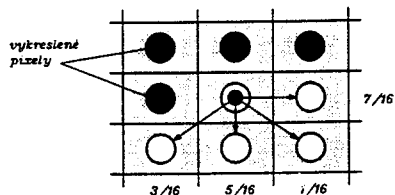
```
Cout := 0;
if Cin > M(x mod 4, y mod 4) then inc(Cout);
```

<sup>3</sup>V uvedeném případě dochází k nesouhlasu mezi rozsahem vstupní intenzity a počtem vzorků generovaných rozptylovací maticí. Pomocí matice  $4 \times 4$  lze totiž vygenerovat 17 různých vzorků. V praxi se tato situace řeší například zopakováním čísla 8 v rozptylovací matici, čímž maximální hodnota v ní obsažená klesne na potřebných 14.

Maticové rozptýlení dává dobré výsledky pro většinu počítačem generovaných obrázků. Pravidelné střídání odstínů nepůsobí rušivě ani na velkých plochách. Při zpracování fotografií však tato pravidelnost dělá dojem umělé vytvořeného obrazu. Právě pro vykreslování obrázků získaných fotografickou cestou byla vyvinuta následující metoda.

### 2.3 Distribuce chyby

Při zachování stejné velikosti obrázku je intenzita vstupního pixelu nahrazena buď jedničkou, nebo nulou na výstupu, čímž nutně dochází ke ztrátě informace. Jeden ze způsobů, jak využít vstupní informaci v co nejvyšší míře, se nazývá **distribuce chyby**<sup>4</sup>. Vstupní intenzita  $C_{in}$  je zde použita nejen k nalezení nejbližší odpovídající intenzity  $C_{out}$ , ale zanedbaná hodnota vzniklá zaokrouhlením  $C_{in}$  z vyššího rozsahu do intervalu  $(0, 1)$  je použita k modifikaci hodnot sousedních pixelů.



Obr 4: Distribuce chyby podle Floyd-Steinberga

Chceme-li například zpracovat pixel o vstupní intenzitě 5, pak výstupnímu pixelu přiřadíme intenzitu 0 a zanedbanou hodnotu 5 přičteme k nejbližším sousedním pixelům. Obdobně při vstupu intenzity 11 vykreslíme výstupní pixel intenzitou 1 a chybu 4 ( $= 15 - 11$ ) odečteme od okolních pixelů. Distribuovaná chyba je pochopitelně mezi sousední pixely pravidelně rozdělena. Jedna z možností je uvedena na obr. 4 a nazývá se Floyd-Steinbergova metoda distribuce chyby. Pravidla pro rozdělení chyby do okolí pixelu byla vytvořena empiricky a jsou známy asi 4 různé způsoby distribuce chyby.

V tomto případě jsou na vstup dány omezující požadavky. Je nutno, aby pixely byly zpracovávány po řádcích (či po sloupcích), tedy nikoliv v libovolném pořadí, jak to dovoluvaly předchozí metody. Pro jeden řádek je třeba vymezit jednorozměrné pole — **paměť chyby**. Před vykreslením každého pixelu je ke vstupní hodnotě  $C_{in}$  přičtena dosud získaná chyba z okolních (dříve zpracovaných) pixelů a teprve výsledná hodnota je převedena do výstupního rozsahu. Zanedbaná hodnota se opět rozdělí a přičte do paměti chyby.

Koeficienty udávající podíl chyby distribuovaný do okolních pixelů (resp. do dočasné paměti chyby) jsou pochopitelně neceločíselné. Vzhledem k celočíselnému charakteru zpracování intenzit je nutno dbát na to, aby nebyla celočíselným dělením zanedbána žádná část. Znamená to, že při rozdělování chyby z jednoho pixelu do jeho sousedů je třeba poslední díl distribuované chyby určit nikoliv uvedeným vzorcem, nýbrž jako rozdíl celé chyby a součtu všech dříve vypočítaných dílů této chyby.

<sup>4</sup>V angl. literatuře se objevují pojmy *error diffusion* nebo *error distribution*.

Floyd-Steinbergova metoda dovoluje při výpočtech využít bitového posunu vpravo, a tím zjednodušit výpočty. Vedle tohoto způsobu distribuce chyby jsou uváděny ještě další metody (Burkes, Stucki), spojené se skutečným aritmetickým výpočtem distribuovaných dílů chyby, a tím i s většími časovými nároky ([Ulic87], [Knut87]).



a) náhodné rozptýlení



b) rozptylovací matice typu  $M_4$

Obr 5: Fotografie Sarah Bernhardtové zpracovaná různými rozptylovacími metodami

c) rozptylovací matice typu  $M_p$

d) distribuce chyby



Na obrázcích 5a-d jsou ukázány různé rozptylovací metody použité na dobovou fotografii Sarah Bernhardtové. Vstupní digitalizovaný obraz měl 256 stupňů šedi.

### 3 Barevné palety

Pro výklad rozptylovacích metod v barevném prostoru použijeme velmi časté situace, kdy vstupní barevné informace jsou vypočítávány (nebo načítány) v kvalitě 24 bitů na pixel a mají být zobrazeny na zařízení s 8 bity na pixel. Předpokládejme dále, že oněch 8 bitů na pixel je indexem do palety, ve které je možné nastavit libovolných 256 odstínů z 16 miliónů možných barevných kombinací.

Kromě volby rozptylovací metody se nyní objevuje i otázka vhodného nastavení barevné palety. Je zřejmé, že barevná paleta by měla obsahovat ty barevné odstíny, které se v daném obrázku objevují nejčastěji. Každému obrázku by potom příslušela vlastní paleta barev.

Pokud jsou vstupní data postupně vytvářena nějakým algoritmem a není předem známo barevné složení celého obrázku, je nutno zapsat do palety pevně dané barevné odstíny, které budou později využívány pro kresbu. Jedno z nejjednodušších řešení představuje naplnění palety 256 odstíny šedi. Barevné odstíny na stupně šedi můžeme převést velmi přesně podle vzorce:

$$I = 0.299 r + 0.587 g + 0.114 b, \quad (1)$$

v němž  $r$ ,  $g$  a  $b$  jsou základní barevné složky vstupní barvy a hodnota  $I$  je výsledná intenzita bílé. Obrázek takto vykreslíme průběžně pouze černobíle, úplné barevné informace můžeme uložit na disk a po důkladnějším rozboru barevného složení obrázku nastavíme barevnou paletu pro jeho závěrečné vykreslení v barvách.

Ukládání úplných barevných informací na disk ovšem zabírá značný diskový prostor a navíc musíme na barevný výsledek čekat až do konečného zpracování všech pixelů. Proto byly vyvíjeny metody, které po pevném (univerzálním) nastavení barevné palety umožní pomocí rozptylování barev dosáhnout kvalitního zobrazení již v průběhu generování obrázku.

#### 3.1 Barevná paleta 3-3-2

Paleta, která má být použitelná pro obrázky s různým barevným složením, musí zřejmě obsahovat pravidelně vybrané zástupce z barevné krychle RGB. Krychli však nemůžeme rozdělit ve všech třech osách stejnoměrně, protože pro zakódování jednotlivých barevných zástupců máme jen 8 bitů, které nelze rozdělit na 3 stejné díly. Proto rozdělíme RGB krychli na  $8 (= 2^3)$  řezů v ose R, 8 řezů v ose G a na  $4 (= 2^2)$  řezů v ose B. Počet bitů pro kódování modré bude tedy nižší než pro ostatní dvě základní barvy. Toto omezení není příliš významné – lidské oko je na odstíny modré málo citlivé (viz vzorec 1), a tak redukce stupnice modré z 256 možných na 4 odstíny je pro většinu případů ještě snesitelná.

Vrcholy takto vytvořené sítě řezů v krychli RGB představují 256 zástupců do palety, která se standardně nazývá paleta 3-3-2. Mezi 8-bitovým obsahem pixelu a barevným složením položky palety je jednoznačný vztah. Má-li pixel hodnotu  $i$ , pak  $i$ -tá položka palety obsahuje následující velikosti složek  $r$ ,  $g$  a  $b$ :

$$\begin{aligned} r &= (((i \text{ shr } 5) \quad ) * 255) \text{ div } 7, \\ g &= (((i \text{ shr } 2) \text{ and } 7) * 255) \text{ div } 7, \\ b &= (((i \quad ) \text{ and } 3) * 255) \text{ div } 3. \end{aligned} \quad (2)$$

Prvé tři bity každého pixelu obsahují kódy 0-7 pro 8 stupňů červené. Jim v paletě 3-3-2 odpovídají velikosti červené složky v osmi stupních, avšak v rozsahu 0-255.

Pro účely rychlého orientačního kreslení obrázku je takto zvolená paleta dostačující již sama o sobě. Vstupní barevnou informaci stačí převést z rozsahu 8-8-8 bitů na příslušné 3-3-2 bity. Tento převod lze provést pomocí celočíselné aritmetiky. Například složku  $r_8$  z 8 bitů převedeme do rozsahu 3 bitů na hodnotu  $r_3$  podle vzorce:

$$r_3 = r_8 * 7 \text{ div } 255 \quad (3)$$

a výsledná barva pixelu  $I$  vznikne bitovým složením hodnot redukovaných barevných složek:

$$I = r_3 \text{ shl } 5 + g_3 \text{ shl } 2 + b_2. \quad (4)$$

Kvalitnějšího vzhledu obrázku dosáhneme ovšem teprve tehdy, vezmeme-li v úvahu i onu zanedbanou hodnotu, která se ztrácí při převodu z 8 bitů na nižší počet. Tuto hodnotu budeme značit  $r_{rest}$ . Na různých způsobech zpracování  $r_{rest}$  jsou založeny následující rozptylovací metody. Jejich společným rysem je to, že každá základní barevná složka je zpracovávána samostatně a výsledná barva vznikne podle vzorce 4.

Všechny rozptylovací metody, které jsme uvedli pro černobílé obrázky, lze použít i pro zpracování barevné informace. Postup lze rozdělit do několika kroků:

1. Pomocí vzorce 3 nejprve nalezneme nejbližší barevnou hodnotu existující v paletě 3-3-2. Ke vstupní intenzitě  $r_8$  tedy najdeme intenzitu  $r_3$ .
2. Zároveň určíme zanedbanou hodnotu (např.  $r_{rest} \in (0, 31)$  pro červenou složku), která vznikla zaokrouhlením vstupní barvy do nižšího výsledného rozsahu.
3. Podle použité rozptylovací metody (náhodné nebo maticové rozptýlení) použijeme zanedbanou hodnotu  $r_{rest}$  k případnému zvýšení výsledné intenzity  $r_3$  o jedničku.

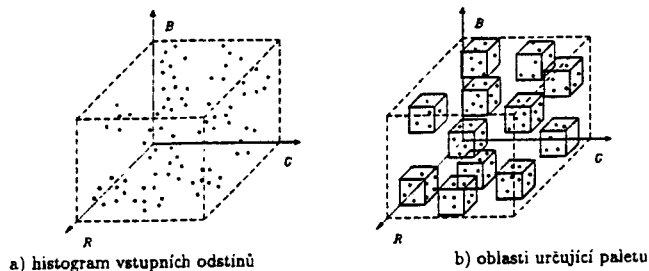
Takto zpracujeme všechny tři barevné složky a z výsledných intenzit složíme hodnotu pixelu podle vztahu 4. U metody distribuce chyby je postup obdobný, využívá se však navíc tři pomocných pamětí chyby pro jednotlivé barevné složky.

#### 3.2 Adaptivně vytvářená barevná paleta

Pevně daná paleta typu 3-3-2 dovoluje použít rozptylovacích technik, aniž by byla předem známa barevná struktura celého obrázku. Při požadavku na co nejvyšší kvalitu obrazu je taková paleta nevhodná, protože v řadě případů obsahuje i takové odstíny, které se na obrázku vůbec nevyskytují, případně jsou zastoupeny v minimálním počtu. Tehdy je na místě vytvořit paletu, která bude odpovídat rozložení barevných odstínů v konkrétním obrázku, a která bude tedy pro každý obrázek definována zvlášť.

Nalezení palety pro daný obrázek je náročné jak časově, tak paměťově. V první fázi je vytvářen histogram četnosti barevných odstínů obsažených v celém obrázku. Pro další výpočty je důležitý nejen fakt, že se určitý odstín v obrázku objevuje, ale i to, jak často je v něm obsažen. Je zřejmé, že vytvoření histogramu představuje zejména na osobních počítačích

pěkný oříšek – představme si např. objem paměti potřebný pro histogram všech  $2^{18}$  barevných kombinací pro adaptér VGA. Přitom již poměrně malý obrázek o rozměrech  $256 \times 256$  pixelů obsahuje okolo 30–40 tisíc různých odstínů. V praxi se našťastí ukazuje, že jednotlivé barvy ve výsledné paletě nebývají těsně vedle sebe. To z hlediska hledání palety znamená, že histogram můžeme vytvořit jen podle 5 nejvýznamnějších bitů každé základní barevné složky.



Obr 6: Vytváření adaptivní palety

V další části adaptivního vytváření palety je potřeba na základě histogramu nalézt 256 nejhodnějších zástupců. Zde nám opět poslouží představa barevného prostoru jako krychle RGB (obr. 6a), ve které jsou pomocí teček znázorněny použité odstíny. Naším úkolem je nalézt 256 oblastí (obr. 6b), které mají pokud možno stejné vlastnosti ([Heck82], [Wan88]). Požadavky kladené na tyto oblasti mohou být:

- oblasti se tvarem blíží krychli a mají všechny stejnou velikost
- oblasti obsahují stejný počet odstínů

Není jednoznačně dáno, kterou z podmínek vybrat. Algoritmicky jednodušší je první varianta. Při ní se nejprve uzavřou všechny odstíny obsažené v obrázku do omezujícího kvádra a poté se opakovaně provádí dvojice operací:

1. Nalezení oblasti s největším rozměrem v jedné z os R, G, B
2. Rozdělení této oblasti na polovinu v ose s největším rozměrem a vytvoření dvou nových omezujících kvádrů

Tento cyklus se opakuje až do vytvoření 256 oblastí. Každá oblast představuje jednu položku v paletě. Za hodnotu položky palety lze zvolit buď aritmetický průměr barevných odstínů v dané oblasti, nebo ten odstín, který se v oblasti vyskytuje nejčastěji.

Stanovením palety pro daný obrázek ovšem práce nekončí. Vstupní data obrázku je nyní nutno znovu přečíst a zpracovat, tj. pro jejich zobrazení většinou použít rozptylovací techniky. Na rozdíl od výše uvedených metod již nelze zpracovávat barvu pixelu po jednotlivých základních složkách, nýbrž jako jeden celek. Vždy je tedy třeba pro danou vstupní barvu nalézt nejbližší barvu existující v paletě. Nalezenou barvu použijeme k vykreslení pixelu

na obrazovce. Žádáme-li co nejkvalitnější obraz, musíme dále zpracovat rozdíl mezi vstupní barvou a zobrazovanou barvou z palety. Jako jediná možná technika se jeví distribuce chyby.

Nalezení nejbližší existující barvy v paletě je časově náročné. Pokud udržujeme v paměti informace o barevných kvádrech vzniklých dělením barevného prostoru, spočívá hledání jen v porovnání složek R, G, B vstupní barvy s mezními rozměry oblastí. Nemáme-li údaje o dělení barevného prostoru zachovány, prohledává se všech 256 položek palety a hledá se euklidovská nejbližší hodnota.

Samotná kombinace hledání palety pomocí stejně velkých oblastí a následná distribuce chyby nemusí vždy dávat dokonalé výsledky. Zástupci do palety jsou vybráni pomocí aritmetického průměru z každé oblasti a při následném zpracování vstupních odstínů může větší skupina stejných barev z okraje oblasti způsobit kumulaci chyby směrem k další oblasti, opticky i značně odlišné. Již z této úvahy je vidět, že charakter vstupních dat a střídání barevných odstínů má významný vliv na výsledný vzhled obrázku. Určitého zlepšení docílíme zařazením všech osmi vrcholů základní barevné krychle mezi položky palety (přidáme čistou červenou, zelenou, modrou, žlutou, zelenomodrou, fialovou, bílou a černou barvu).

Významným faktorem, který jsme zatím do tvorby adaptivní palety nezařadili, je charakter vnímání barev lidským okem. Výrazného zlepšení dosáhneme, vytváříme-li barevné oblasti nikoliv dělením v polovině, ale dělením, které odpovídá vztahu 1. Znamená to, že cílem dělení jsou kvádry s délkami stran R, G, B v poměru  $3,3 : 1,7 : 10$  (převrácené hodnoty koeficientů ze vztahu 1). V této souvislosti lze také přejít do jiného barevného prostoru, např. HSV nebo HLS a vytvářet oblasti vhodných vlastností v něm ([Fole90]).

Na vzhled výsledného obrázku má vliv ještě řada dalších faktorů spjatých se zpracováním světelných a barevných informací v lidském oku a mozku. Jedním ze známých faktorů je citlivost oka na změnu intenzity (jasu). Intenzita je v displejích odstupňována lineárně. Oko je však při změně z jedné intenzity na druhou citlivé nikoliv na rozdíl jasových úrovní, nýbrž na jejich poměr. Do procesu tvorby barevného obrázku proto můžeme zahrnout tzv. gama korekci, která upraví hodnoty jasových úrovní. Zároveň je také pravda, že jasová stupnice je pro konkrétní displeje různá a že stanovení přesných koeficientů pro gama korekci vyžaduje určitá měření ([Cowa83]). Některé zobrazovací adaptéry již mají gama korekci zahrnutou do procesu převodu číselných údajů o jasu na napěťové hodnoty v displeji.

## Závěr

V tomto článku jsme uvedli základní techniky používané při zobrazování mnoha barev na zařízeních s omezenou velikostí palety. Pozornému čtenáři jistě neunikla skutečnost, že neexistuje univerzální metoda pro jakákoliv vstupní data, že barevné složení obrázku je natolik různorodé, že i náročné rozptylovací metody mohou v některých případech zklamat. Zejména tvorbu adaptivní palety můžeme s trochou nadsázky zařadit do oblasti grafické magie.

Na rozptylovacích technikách se také zřetelně projevuje další jev typický nejen pro oblast grafiky, totiž že i malý vzrůst kvality výsledku je spojen s velkým úsilím při jeho tvorbě. Při vykreslování barevných obrázků je navíc celkový dojem značně závislý na individuálním vnímání pozorovatele.

## Literatura

- [Cowa83] Cowan, W.: *An Inexpensive Scheme for Calibration of a Colour Monitor in Terms of CIE Standard Coordinates*. SIGGRAPH 83, 315-321.
- [Fole90] Foley, J., van Dam, A., Feiner, S., Hughes, J.: *Computer Graphics - Principles and Practice*. 2nd ed., Addison-Wesley, Reading, Massachusetts, 1990.
- [Heck82] Heckbert, P.: *Color Image Quantization for Frame Buffer Displays*. SIGGRAPH 82, 297-307.
- [Holl80] Holladay, T. M.: *An Optimum Algorithm for Halftone Generation for Displays and Hard Copies*. Proceedings of the Society for Information Display, 21(2), 1980, 185-192.
- [Knut87] Knut, D.: *Digital Halftones by Dot Diffusion*. ACM TOG, 6(4), October 1987, 245-273.
- [Soch89] Sochor, J., Žára, J.: *Světlo a stín v počítačové grafice*. MOP '89, 1.díl, str.5-93, Vinné, 1989.
- [Ulic87] Ulichney, R.: *Digital Halftoning*. MIT Press, Cambridge, Massachusetts, 1987.
- [Wan88] Wan, S., Wong, K., Prusinkiewicz, P.: *An Algorithm for Multidimensional Data Clustering*. ACM Transactions on Mathematical Software, 14(2), June 1988, 153-162.

## 1) Úvod

V dnešní době, která se vyznačuje složitostí technologických procesů vytvořených z jednodušších celků, může snadno dojít k nouzovému stavu systému nebo jeho havárii. Příčinou je většinou mechanická porucha nebo výpadek určitého zařízení.

Logickým důsledkem takovéto situace je příkaz k opravě nebo výměně poškozeného zařízení a to za provozu ostatní technologie. Aby tato oprava byla možná, je nutné poškozené zařízení izolovat od okolí a zajistit tak bezpečnost pracovníka, který bude opravu provádět. Co všechno je nutné udělat, vymezují předpisy pro konkrétní systém.

## 2) Specifika elektráren

## a) Elektrárenské předpisy

V jaderné energetice platí, z pochopitelných důvodů, zvláště přísné předpisy, které jsou velmi podrobné. Ke každému zařízení podléhajícímu údržbě ( např. soustrojí, potrubní části ...) existuje katalogový list. V něm lze nalézt obchodně-technické údaje, technické parametry, vazby na okolí a jiné důležité informace.

## b) Množství údajů a zařízení

Faktorem ztěžujícím údržbu poškozených zařízení je rozsáhlost systému, který je nutno sledovat a kontrolovat. Například v Elektrárně Dukovany je evidováno asi 700 000 zařízení a příslušná dokumentace pro údržbu zabírá asi 3 vagóny. Z těchto přibližných čísel je vidět, o jak obrovské systémy se jedná. V případě havárie je tento systém údržby pochopitelně velice nepružný.

## c) Obecně o MNT-GRAF

V současnosti již existují počítačem podporované systémy údržby, kdy dokumentace je uložena v databázi a veškerá činnost probíhá na obrazovce. V tomto příspěvku bude popisován systém údržby s grafickou podporou, tzv. MNT-GRAF, což je československé rozšíření systému MNT (Maintenance Management System) firmy Hewlett Packard (HP) o grafickou podporu vytváření tzv. S-příkazů a práci s nimi, s ohledem na okolní technologii a její okamžitý stav.