

Algoritmy počítačové grafiky III

(Algorithms for Computer Graphics III)

Prof.Ing.Václav Skala, CSc.

<http://www.VaclavSkala.eu>

Abstrakt

Algoritmy počítačové grafiky a jejich implementace je nedílnou součástí jak grafických systémů, tak i CAD/CAM systémů. V této práci jsou uvedeny základní algoritmy, metody a postupy, které se používají v různých modifikacích i v dnešních systémech

Abstract

Algorithms for computer graphics and their implementation is a part of graphical systems and CAD/CAM systems. Fundamental algorithms, methods and approaches used in today's systems are explained.

Algorithmy počítačové grafiky III
Algorithms for Computer Graphics III
Published & printed by: Vaclav Skala – UNION Agency
Na Mazinach 9
CZ 322 00 Plzen
Czech Republic
Year 2011
ISBN 978-80-86943-21-3

Electronic version <http://www.VaclavSkala.eu> <http://Graphics.zcu.cz>

Profil autora

Prof. Ing. Václav Skala, CSc. se zabývá počítačovou grafikou od r.1975, kdy na Vysoké škole strojní a elektrotechnické v Plzni zavedl a následně vyučoval předmět Počítačová grafika a umělá inteligence. V roce 1981 obhájil disertační práci na témate relačních databází se specializací na reprezentaci relací a optimalizaci dotazů uživatele. V současné době se odborně věnuje především algoritmům počítačové grafiky a vizualizaci dat a algoritmům včetně matematických aspektů. Je vedoucím Centra počítačové grafiky a vizualizací (<http://Graphics.zcu.cz>) při Katedře informatiky a výpočetní techniky na Fakultě aplikovaných věd Západočeské univerzity v Plzni.



V letech 1984-1989 působil na Brunel University v Londýně a později přednášel na NATO Advanced Study Institute v zahraničí. V roce 1996 se stal profesorem na Západočeské univerzitě, odborně působil na Bath University v U.K., Gavle University ve Švédsku a na dalších zahraničních odborných pracovištích.

Prof. Skala je řešitelem zahraničních i národních odborných výzkumných projektů, členem několika redakčních rad prestižních zahraničních odborných časopisů, členem programových výborů mezinárodních odborných konferencí. Od r.1992 je organizátorem mezinárodních odborných konferencí WSCG zaměřených na počítačovou grafiku, vizualizaci dat a počítačové vidění (<http://www.WSCG.eu>).

V současné době je prof. Skala profesorem na Západočeské univerzitě v Plzni a VŠB-Technické univerzitě v Ostravě. V roce 2010 odborně působil též na Přírodovědné fakultě Ostravské univerzity v Ostravě (<http://AlgVis.osu.cz>).

V roce 2010 získal významné ocenění mezinárodní asociace pro počítačovou grafiku

„Fellow of the Eurographics Association“

za dlouhodobé odborné výsledky a organizační aktivity v oblasti počítačové grafiky.

Poznámky pro laskavého čtenáře

Toto je rekonstrukce učebního textu Algoritmy počítačové grafiky I– III, který vznikl pro studenty Západočeské univerzity v Plzni v roce 1991. Tato publikace vznikla rozšířením publikace Počítačová grafika I - II, která vznikala v roce 1989 pro potřeby studentů Vysoké školy strojní a elektrotechnické a která byla vydána v roce 1990.

Jde tedy o text zohledňující algoritmy, metody a stav technologie před více než před 20 lety. Je nutné si uvědomit, že obor počítačové grafiky je poměrně velmi mladý, bouřlivě se rozvíjející a poměrně hodně závislý na dostupných technologiích. Nicméně většina metod, postupů a algoritmů je používána dodnes.

Pochopitelně je oblast počítačové grafiky dnes podstatně širší a zahrnující mnohé jiné oblasti, které jsou odborně rozvíjeny kolegy v **Centru počítačové grafiky a vizualizací** při Katedře informatiky a výpočetní techniky Fakulty aplikovaných věd, viz <http://Graphics.zcu.cz>

Rád bych proto požádal čtenáře, aby uvedený text a kvalitu tisku posoudil v kontextu té doby, kdy:

- nebyl prakticky přístup k zahraniční literatuře a publikace se velmi obtížně získávaly přes mezinárodní výpůjčky, resp. přes osobní kontakty
- nebyl k dispozici Internet, e-mail, WEB a digitální knihovny
- byly k dispozici první 8mi bitové PC systémy s „fantastickou“ kapacitou paměti 512 KB
- byly k dispozici první jehličkové tiskárny (9 tiskových bodů na textovou řádku) a „unikátní“ textové procesory T602 a výborný graficko-textový procesor Chi-Writer (<http://en.wikipedia.org/wiki/ChiWriter>), ve kterém byl text napsán.

Je nutné si uvědomit, že před rokem 1989 bylo velmi obtížné získávat odborné publikace a publikovat odborné výsledky v mezinárodně uznávaných časopisech a na mezinárodních konferencích.

Již v roce 1975 na Vysoké škole strojní a elektrotechnické, Fakultě elektrotechnické vzniká předmět Počítačová grafika a umělá inteligence

Nové odborné výsledky, algoritmy a metody laskavý zájemce nalezne na URL:

<http://www.VaclavSkala.eu>.

Také bych rád zájemce o počítačovou grafiku odkázal na publikace z mezinárodních konferencí WSCG, na časopis Journal of WSCG a GraVisMa workshopy:

- **WSCG** – International Conferences on Computer Graphics, Visualization and Computer Vision
<http://www.WSCG.eu>
- **GraVisMa** – Computer Graphics, Vision and Mathematics <http://www.GraVisMa.eu>

Tento text, který vznikl rekonstrukcí z archivu autora, byl doplněn o seznam publikací tak, aby tvořil ucelenou publikaci. Předpokládá se, že další části původní publikace budou rekonstruovány následně.

Je mi milou povinností poděkovat všem, kteří mi stimulovali moje odborné aktivity a které jsem měl tu čest nejen potkat na Brunel University a na NATO Advanced Study Institutes, ale i s mnohými odborně spolupracovat. Patří k nim zejména:

Prof. M.L.V. Pitteway

Brunel University,
U.K.



Prof. Jack E. Bresenham

IBM Corp., USA
a
Winthrop University,
USA



Prof. F.R.A. Hopgood

Rutherford Appleton
Laboratory, U.K.
a
Oxford Brookes
University, U.K.



Prof. David F. Rogers

U.S. Naval Academy,
USA



Předmluva

Je mou milou povinností poděkovat všem, kteří přispěli ke vzniku této publikace, ať už vytvořením podmínek nebo stimulací k vlastní práci. Chtěl bych zejména poděkovat ing.I.Kolingerové za její konkrétní, podnětné a kritické připomínky, bývalým i současným studentům Západočeské univerzity v Plzni se zaměřením Počítačová grafika a CAD systémy, kteří byli neocenitelnými pomocníky při ověřování algoritmů a tvorbě demonstračních a výukových programů.

1. Úvod

V publikaci jsem se pokusil zachytit princip a rozvoj vybraných algoritmů počítačové grafiky z oblasti použití světla, barev a barevných systémů. Vzhledem k prudkému rozvoji použití barev je tato publikace pojata jako přehledová, přičemž podrobnosti lze nalézt v literatuře. Některé pasáže bylo nutno zredukovat na téměř informativní úroveň z důvodů přílišné teoretické složitosti, složitosti algoritmů, případně pro jejich nedostupnost, neboť firmy skutečně použité metody nepublikují.

Počítačová grafika se dnes používá v nejrůznějších odvětvích lidské činnosti. Je poněkud paradoxní, že ačkoliv byla tato oblast rozvinuta především z hlediska možných technických aplikací, je jednou z nejvíce finančně výhodných oblastí právě oblast ryze netechnická, a tou je animace filmů. V dnešní době existují specializované společnosti, které nejenže vytvářejí speciální programové vybavení, ale též i příslušné systémy. Je zřejmé, že tato oblast vyžaduje zcela odlišné prostředky než jsou technické aplikace. Mezi prvními filmy, které byly vytvořeny za pomoci animace, je známý film **TRON**.

Je nezbytné též upozornit na ty aplikace, které nejsou sice tak "efektní", nicméně stejně důležité. Jednou z nich je použití počítačové grafiky pro komunikaci s člověkem v systémech řízení technologických procesů a ve výrobě vůbec. Z hlediska samotné počítačové grafiky jde o "nezajímavý" problém, neboť jde o výstupy téměř statické a jednoduché, které zobrazují např. schéma zapojení v rozvodně elektrického proudu, schéma potrubního hospodářství v petrochemickém provozu apod. Do těchto základních "statických" výstupů se pak promítají změny, např. přepnutí rozvaděčů v rozvodně s barevným vyjádřením těch částí, které jsou pod napětím apod.

Kromě uvedených aplikací je a bude těžiště aplikací počítačové grafiky především v technických aplikacích, kdy půjde zejména o podporu konstrukčních prací. Takové systémy (někdy též nazývané CAD systémy) však neobsahují jen prostředky počítačové grafiky, které musejí být z technického hlediska velmi výkonné, ale musejí též obsahovat části pro práci s rozsáhlými databázemi a vazbu na vlastní přípravu výroby. Tyto systémy, které se připravují "na míru" určité aplikace, doznaly asi největšího použití v automobilovém, leteckém a loďařském průmyslu.



Obr. 1.1

Další oblastí aplikace počítačové grafiky je oblast systémů usnadňujících publikační činnost. Jednodušší systémy se nazývají textovými procesory, resp. WP systémy (Word Processor Systems),

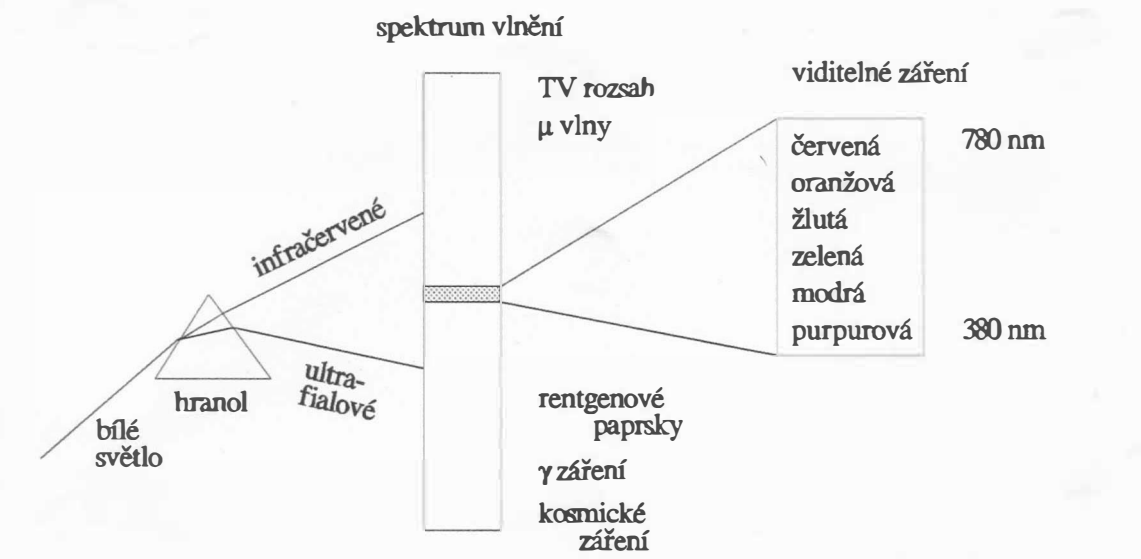
mezi které lze zařadit WordPerfect, Chi-Writer apod. Tato třída systémů je určena pro psaní rozsahem malých publikací. Systémy typu DTP systémy (Desk Top Publishing Systems) jsou určeny k usnadnění činností nutných k vydávání rozsáhlých publikací, které mají být tištěny v kvalitě knižního tisku. Do této třídy lze zahrnout systém VENTURA, resp. TEX. Obě třídy dnes velmi uspokojivě řeší problematiku práce pouze s černobílými obrázky.

Ve všech výše uvedených aplikacích počítačové grafiky je použito barev k získání výsledného žádaného vjemu. Barva je novou dimenzí, která vstupuje do oblasti vizualizace dat.

U všech systémů a periferních zařízení je zcela jasný postup stále častějšího využívání barev, přičemž rozlišovací schopnost a poskytovaná paleta barev je stále větší. U některých zařízení, např. u laserových nebo inkoustových tiskáren bude však vždy omezen počet tzv. základních barev na určitý počet. Je proto nutné nejen pochopit některé základní principy použití barev v počítačové grafice, ale i porozumět principům jejich míchání a speciálním technikám, které umožňují, aby pozorovatel získal vjem bohaté palety barev na výsledném obrázku, i když skutečný počet barev poskytovaný zařízením je velmi omezený. Jako příklad je možné uvést výstup z inkoustové tiskárny firmy Iris Graphics, Inc. (U.S.A.), která používá pouze čtyř barev a rozlišení 1500 dpi, viz obr.1.1 (bohužel vzhledem k možnostem reprodukční techniky lze předložit pouze černobílou kopii).

2. Barva

Rozvoj prostředků počítačové grafiky umožnil, aby se barva a různé úrovně jasu, resp. šedi, staly nedílnou součástí počítačové grafiky. Ačkoliv proces vnímání barvy lidským mozkiem je předmětem psychologického a fyziologického zkoumání a není dosud zcela vysvětlen, může být fyzikální podstata vyjádřena na základě formálního aparátu ověřeného teoretickými a experimentálními výsledky. V roce 1666 objevil Isaac Newton, že sluneční paprsek procházející skleněným hranolem není na výstupní straně bílý, ale obsahuje spojité spektrum barev od barvy fialové na jednom konci k barvě červené na konci druhém. Toto barevné spektrum může být rozděleno do šesti oblastí, a to purpurové (magenta), modré (blue), modrozelené (cyan), zelené (green), žluté (yellow) a červené (red), přičemž přechod z jedné oblasti do druhé je pozvolný, viz obr.2.1.



Obr. 2.1

Rozsah barev je dán rozsahem viditelného vlnění, které je pouze malou částí širokého spektra elektromagnetického vlnění, jež od nejkratších vlnových délek, tj. od kosmického záření a γ -záření, přechází v ultrafialové záření a do oblasti vlnění, které je naše oko schopno vnímat. Oblast viditelného vlnění ohraničuje infračervené záření. Se zvětšující se vlnovou délkou pak následuje mikrovlnná oblast a oblast pro TV a radiové signály atd. Barva objektu, kterou vnímáme, je dána povahou

(přesněji řečeno spektrálním složením) světla přicházejícího od pozorovaného objektu. Ta pak záleží na povaze (přesněji spektrálním složení) zdroje světla a na povaze (přesně spektrální odraznosti či propustnosti) pozorovaného tělesa. Některé materiály světlo nejen odrážejí, ale částečně i propouštějí, např. sklo, celofán apod. Materiály lze tedy dělit na průhledné, průsvitné a neprůsvitné. Je-li použit např. červený zdroj světla k osvětlení odrážejícího povrchu, který odráží pouze modré světlo, pak se povrch bude jevit jako černý. Tento povrch nemůže odrazit světlo modré barvy, je-li použit k osvětlení zdroj červeného světla.

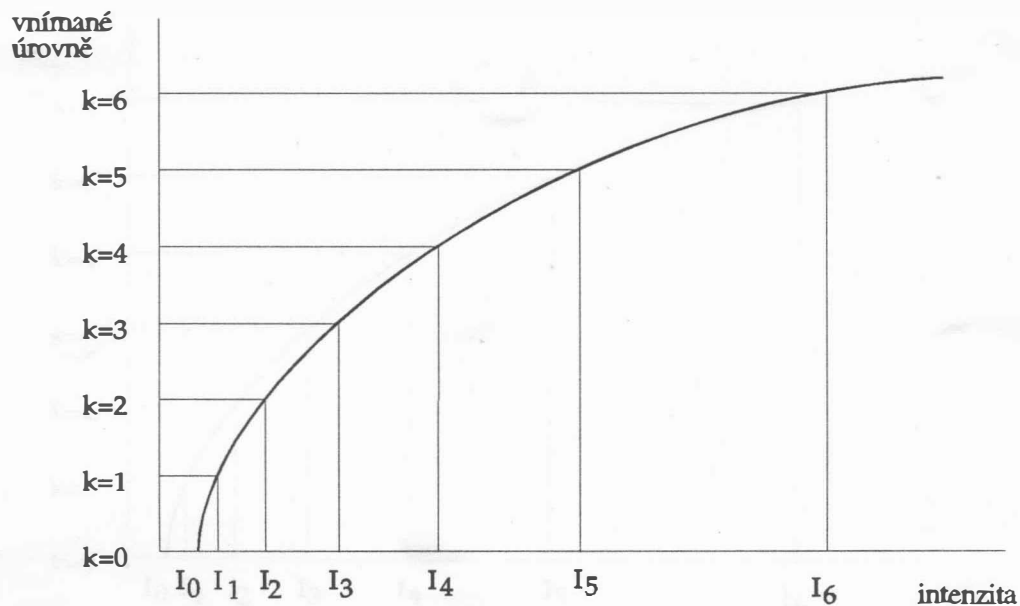
Obsahuje-li světlo všechny vlnové délky o přibližně stejné intenzitě, je zdroj světla, resp. objekt, achromatický, neboli nepestrý. Achromatický zdroj světla se jeví bílý. Je-li odražené nebo průpuštěné světlo (u transparentního objektu) achromatické, jeví se jako bílé, černé anebo šedé o určité úrovni intenzity. Z experimentů vyplývá, že objekty odrážející achromaticky více než 80% světla ze zdroje bílého světla se jeví jako bílé. Jestliže objekty odrážejí méně než 3% světla, jeví se jako černé. Intenzitu odraženého světla je vhodné uvažovat v rozsahu $\langle 0, 1 \rangle$, kde černé barvě odpovídá hodnota 0 a bílé barvě odpovídá hodnota 1.

2.1 Achromatické světlo

Achromatické světlo je vlastně to, co může vidět např. uživatel černobílého televizního přijímače. Jediným atributem achromatického světla je jeho intenzita. Počet úrovní intenzity je závislý na konkrétním zařízení. Na rozdíl od TV přijímače, kde je možné zobrazit několik úrovní šedi, zařízení typu tiskárna, zapisovač, monochromatický display mohou zobrazit většinou pouze dvě úrovně.

Je přirozenou otázkou, kolik úrovní šedi jsme schopni rozlišit, resp. kolik je jich zapotřebí. V případě použití šestnácti a méně úrovní šedi deteguje oko čáry ohraničující jednotlivé úrovně. Pro 64 úrovní již téměř nelze detekovat jednotlivé přechody úrovní šedi. Pro reprezentaci plynulého přechodu jednotlivých úrovní šedi je postačujících 256 úrovní. Nicméně pro speciální rastrové zobrazovací jednotky může být v některých případech zapotřebí i více než 256 úrovní šedi.

Je-li k dispozici např. 256 úrovní intenzity, vzniká otázka, jak budou jednotlivé úrovně šedi přiřazeny k intenzitám světla tak, aby nevznikaly detekovatelné přechody. Z Lambert-Beerova zákona (fyziologické hledisko) vyplývá, že má-li být stupnice úrovní šedi (jasová stupnice) rovnoměrná, musí být intenzita záření odstupňována logaritmicky.



Obr. 2.1.1

Označíme-li jednotlivé intenzity I_k , kde I_0 je nejmenší dosažitelná intenzita, a je-li počet rozlišitelných intenzit např. 256, pak lze psát (I_0 je nenulová prahová hodnota):

$$I_1 = r \cdot I_0 \qquad I_2 = r \cdot I_1$$

$$I_{255} = r^{255} \cdot I_0 = 1 \qquad I_k = r \cdot I_{k-1} = r^k \cdot I_0$$

kde r je poměr mezi intenzitami, a tedy:

$$r = \left(\frac{1}{I_0} \right)^{1/255}$$

$$I_j = I_0^{(255-j)/255} \qquad 0 \leq j \leq 255$$

Nicméně zobrazení jednotlivých intenzit na stínítku, resp. na filmu není tak jednoduchou záležitostí vzhledem k existujícím nelinearitám. Z tohoto důvodu je nutné použít tzv. gama korekci, která je založena na tom, že intenzita světelného bodu na luminoforu je úměrná napětí U na řídicí elektrodě obrazovky:

$$I = c \cdot U^d$$

kde c, d jsou konstanty.

Pro intenzitu I je tedy nutné určit napětí U , a tedy:

$$U = \left(\frac{I}{c} \right)^{1/d}$$

Nyní pro požadovanou intenzitu I určíme nejbližší intenzitu I_j , a to buď prohledáním tabulky dostupných intenzit nebo určením indexu ze vztahu:

$$j = \text{ROUND} \left(\log_r (I/I_0) \right)$$

kde $\text{ROUND} (x)$ určí nejbližší celou hodnotu k hodnotě x .

Nyní:

$$I_j = r^j \cdot I_0$$

a tedy:

$$U_j = \left(\frac{I_j}{c} \right)^{1/d}$$

Vzhledem k tomu, že hodnoty I_0, c, d jsou závislé na typu obrazovky, je obvyklé zapsat příslušné hodnoty do tabulky příslušné zobrazovací jednotky; podrobně viz [24] (Experimentálně $1 \leq d \leq 4$; $-2,2 \leq c \leq 2,3$).

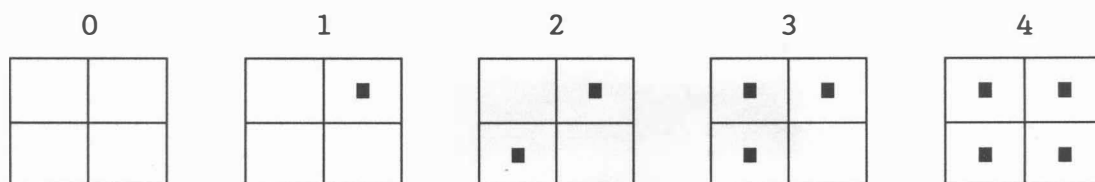
2.2 Půltónování

V některých případech je nutné zobrazit obraz s více úrovněmi šedi, resp. jasu, než je na daném zařízení možné. Techniky řešící tuto problematiku se nazývají půltónování. Byly vyvinuty zejména pro reprodukci fotografií na zařízeních se dvěma úrovněmi šedi (černé a bílé), zejména pro účely tisku novin a levných periodik. Následkem omezené rozlišitelnosti oka, dané mozaikovou strukturou sítnice, integruje oko intenzity světla v oblastech menších než je mez rozlišení. Základní myšlenkou je použití vzorů používajících černých a bílých bodů nakupených tak, aby vytvářely dojem příslušné intenzity.

Techniky půltónování byly vyvinuty vlastně pro transformaci obrázků s více úrovněmi šedi na obrázky se dvěma úrovněmi šedi, přičemž je po transformaci obrázku požadován přibližně "stejný vjem". Proto se také někdy tyto algoritmy nazývají transformační.

Metoda vzorů

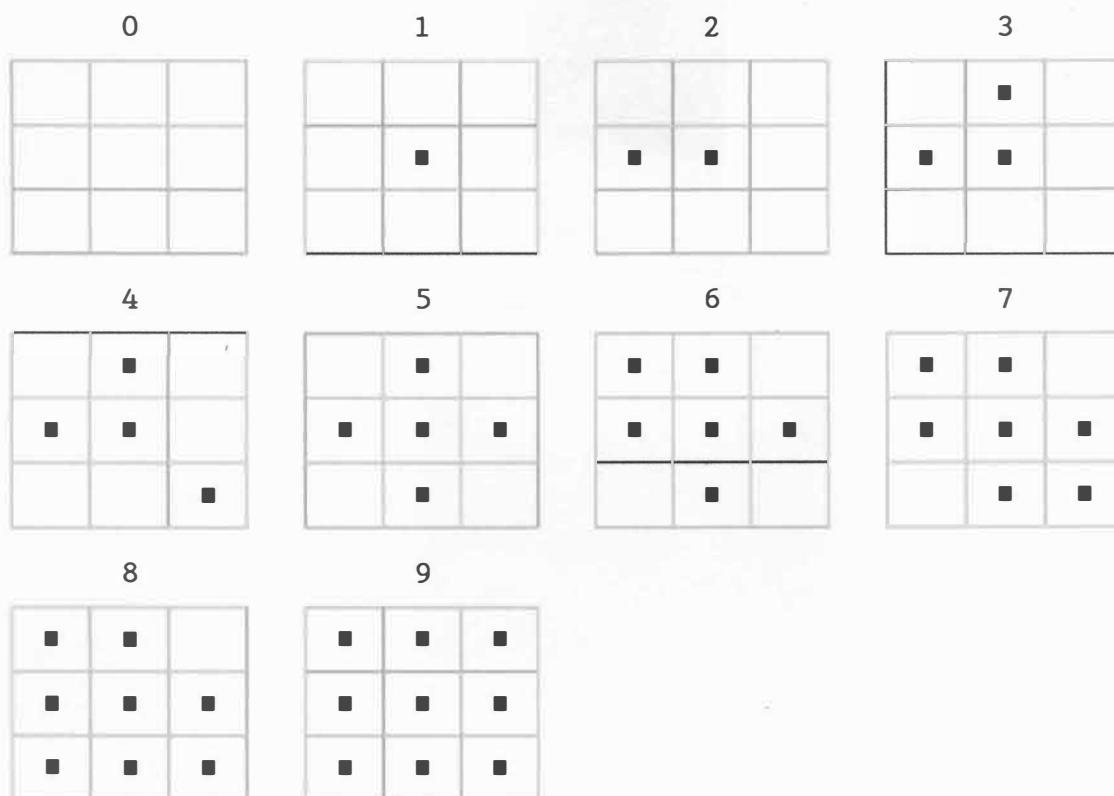
Principem metody vzorů je přímá náhrada pixelu (obrazového elementu), který má určitou úroveň jasu, vzorem rozměru $n \times m$, jehož body (bílé nebo černé) vytváří dojem požadovaného jasu. Pro jednotlivé úrovně je pak nutné stanovit, jak vzor bude definován. Např. pro 5 úrovní šedi je zapotřebí použít vzor 2×2 bodů, tj.:



Vzory typu:



nemohou být použity, neboť vytvářejí ve výsledném obrazu nežádoucí vodorovné nebo svislé čáry. Pro 10 úrovní šedi je nutné použít vzor 3×3 bodů, tj.:

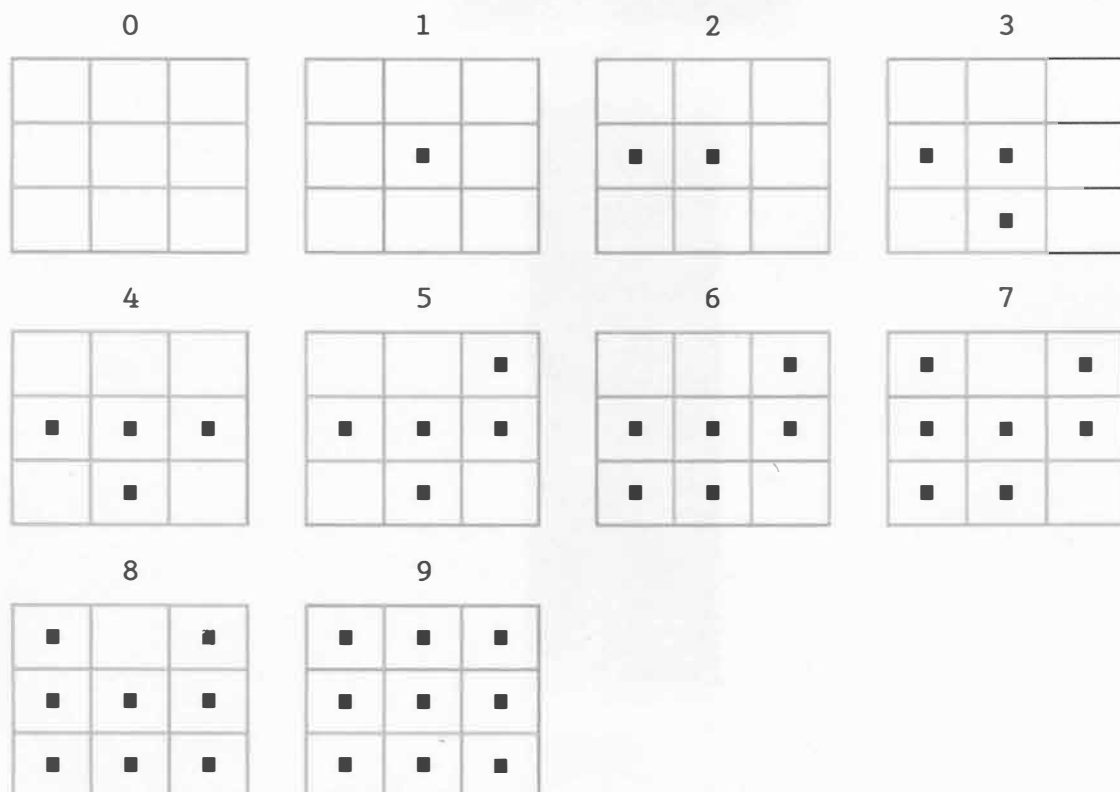


Některé vzory mohou být formálně reprezentovány pomocí matice.

Pro danou intenzitu se aktivují všechny pixely vzoru, jejichž hodnota v matici je menší nebo rovna požadované intenzitě. Např. pro vzor 3 x 3 může být matice určena takto:

$${}^{(3)}\mathbf{T} = \begin{bmatrix} 7 & 9 & 5 \\ 2 & 1 & 4 \\ 6 & 3 & 8 \end{bmatrix}$$

Rozměr matice pak obecně definuje maximální počet úrovní šedi, které lze reprezentovat. Pro danou matici pak dostaneme vzory:



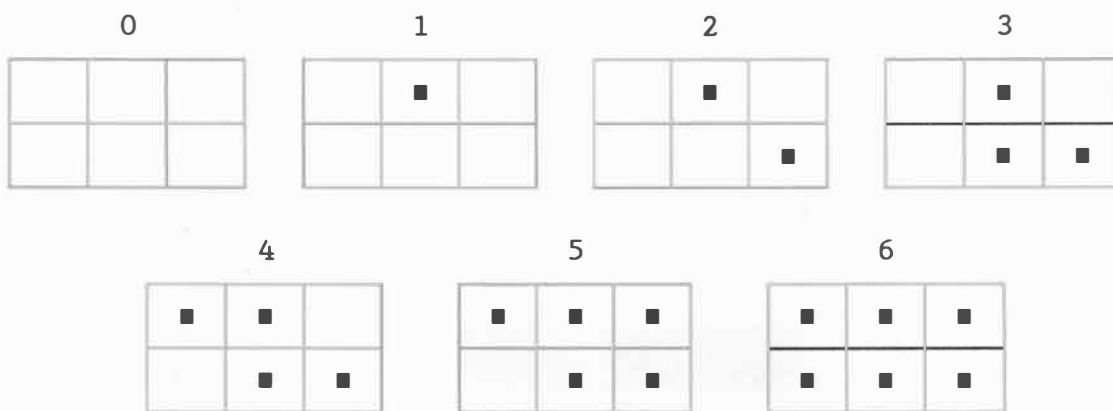
Dříve uvedený vzor 2 x 2 bodů lze pak reprezentovat maticí:

$${}^{(2)}\mathbf{T} = \begin{bmatrix} 3 & 1 \\ 2 & 4 \end{bmatrix}$$

Pro více úrovní je někdy výhodný vzor 4 x 4 bodů, jehož matice je definována:

$${}^{(4)}\mathbf{T} = \begin{bmatrix} 1 & 9 & 3 & 15 \\ 13 & 5 & 14 & 7 \\ 4 & 10 & 2 & 12 \\ 16 & 8 & 11 & 6 \end{bmatrix}$$

V některých případech je nutné nebo vhodné použít vzory, které jsou reprezentovány obdélníkovou maticí, např. pro 7 úrovní šedi

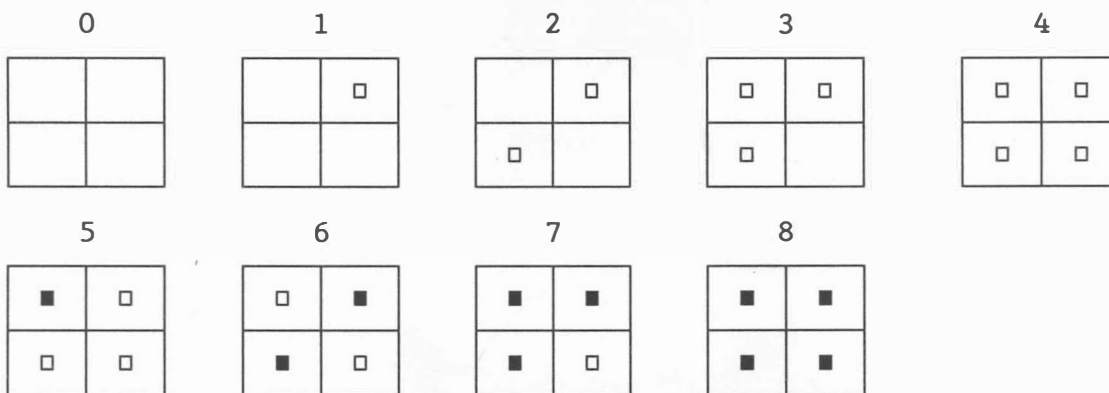


je vzor definován maticí:

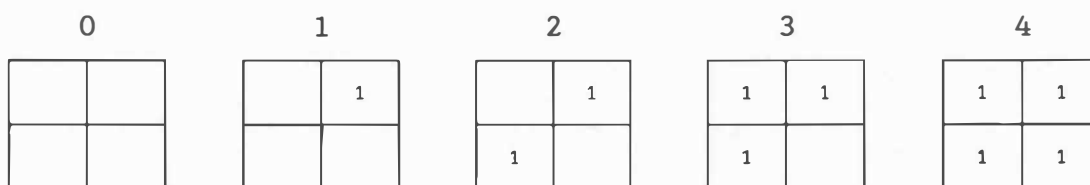
$$(3, 2)_T = \begin{bmatrix} 4 & 1 & 5 \\ 6 & 3 & 2 \end{bmatrix}$$

Použitím obdélníkové matice lze někdy zcela nebo alespoň částečně eliminovat problém různých vzdáleností mezi sousedními body v horizontálním a vertikálním směru u konkrétního zařízení.

V případě některých zařízení je možné zobrazit různě velké body (□ malý, ■ velký bod). Pak lze použít např. vzor 2 x 2 bodů k reprezentaci 9 úrovní šedi:



Jsou-li k dispozici např. 2 bity na pixel (lze případně realizovat i přetiskem), pak vzory mohou mít tvar (čísla označují intenzity - nula se neuvádí):



	5	6	7	8	9																				
	<table border="1" style="border-collapse: collapse; text-align: center;"><tr><td>1</td><td>2</td></tr><tr><td>1</td><td>1</td></tr></table>	1	2	1	1	<table border="1" style="border-collapse: collapse; text-align: center;"><tr><td>1</td><td>2</td></tr><tr><td>2</td><td>1</td></tr></table>	1	2	2	1	<table border="1" style="border-collapse: collapse; text-align: center;"><tr><td>2</td><td>2</td></tr><tr><td>2</td><td>1</td></tr></table>	2	2	2	1	<table border="1" style="border-collapse: collapse; text-align: center;"><tr><td>2</td><td>2</td></tr><tr><td>2</td><td>2</td></tr></table>	2	2	2	2	<table border="1" style="border-collapse: collapse; text-align: center;"><tr><td>2</td><td>3</td></tr><tr><td>2</td><td>2</td></tr></table>	2	3	2	2
1	2																								
1	1																								
1	2																								
2	1																								
2	2																								
2	1																								
2	2																								
2	2																								
2	3																								
2	2																								
	10	11	12																						
	<table border="1" style="border-collapse: collapse; text-align: center;"><tr><td>2</td><td>3</td></tr><tr><td>3</td><td>2</td></tr></table>	2	3	3	2	<table border="1" style="border-collapse: collapse; text-align: center;"><tr><td>3</td><td>3</td></tr><tr><td>3</td><td>2</td></tr></table>	3	3	3	2	<table border="1" style="border-collapse: collapse; text-align: center;"><tr><td>3</td><td>3</td></tr><tr><td>3</td><td>3</td></tr></table>	3	3	3	3										
2	3																								
3	2																								
3	3																								
3	2																								
3	3																								
3	3																								

Více bitů na pixel nebo větší vzory vedou k odpovídajícímu zvýšení dostupných úrovní intenzit.

Používání vzorů vede ke zvětšení výsledného obrazu. Nedochozí ke ztrátě obrazové informace, což např. u zařízení s nastavitelnou hustotou zobrazovaných bodů nevadí a technika poskytuje dobré výsledky. Při omezené velikosti výsledného obrazu však vlastně dochází ke zmenšení rozlišovací schopnosti, což je u některých aplikací nežádoucí. Proto byly vyvinuty speciální techniky pro zlepšení vizuálního rozlišení.

Metoda konstantního prahu

Metoda konstantního prahu je nejjednodušší technikou pŕltónování. Přesáhne-li intenzita pixelu určitou pevnou prahovou hodnotu, pixel se zobrazí jako bílý, v opačném případě jako černý.

```
{ White odpovídá barvě bílé, Black barvě černé }
for y := ymax downto ymin do
  for x := xmin to xmax do
    if I(x,y) > T then PutPixel(x,y,White)
      else PutPixel(x,y,Black);
```

kde $I(x,y)$ je požadovaná intenzita pixelu (x,y)
 $\langle x_{min}, x_{max} \rangle$ je rozlišovací schopnost ve směru osy x
 $\langle y_{min}, y_{max} \rangle$ je rozlišovací schopnost ve směru osy y

Algoritmus 2.2.1

Prahová hodnota T je nastavena na vhodnou hodnotu, např. přibližně na střední hodnotu

$$T = \frac{I_{\max} - I_{\min}}{2}$$

kde I_{\max} a I_{\min} jsou maximální a minimální hodnota jasu v daném obrázku.

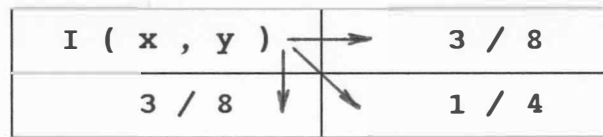
Použití prahových funkcí vede však při skutečných aplikacích k příliš velké ztrátě obrazové informace viz obr.2.2.1, neboť dochází k poměrně velkým chybám při zobrazování jednotlivých pixelů a je prakticky nepoužitelná.



Obr. 2. 2. 1

Floyd-Steinbergova metoda

Floyd a Steinberg [43] navrhli adaptivní metodu, která chyby částečně distribuuje do sousedních pixelů, a to tak, že chyba je distribuována ve směru vpravo a dolů podle pravidla znázorněného na obr.2.2.2.



Obr. 2.2.2

```

const xmin = 0; xmax = 1023; { rozsah rastru pro směr x }
      ymin = 0; ymax = 1023; { rozsah rastru pro směr y }
      Black_gray = 0.0;   White_gray = 1.0;
var T, Error: real;
    x, y: integer;
begin T := (I_max + I_min) / 2;
      for y := ymax downto ymin do
        for x := xmin to xmax do
          begin { určení hodnoty pixelu pro práh T }
            { a chyby Error }
            if I(x,y) < T then
              begin PutPixel(x,y,Black);
                    Error := I(x,y) - Black_gray
              end
            else
              begin PutPixel(x,y,White);
                    Error := I(x,y) - White_gray
              end;
            if x < xmax then I(x+1,y) := I(x+1,y) + 3*Error/8;
            if y > ymin then I(x,y-1) := I(x,y-1) + 3*Error/8;
            if (x < xmax) and (y > ymin)
              then I(x+1,y-1) := I(x+1,y-1) + Error/4
          end
        end
      end
end

```

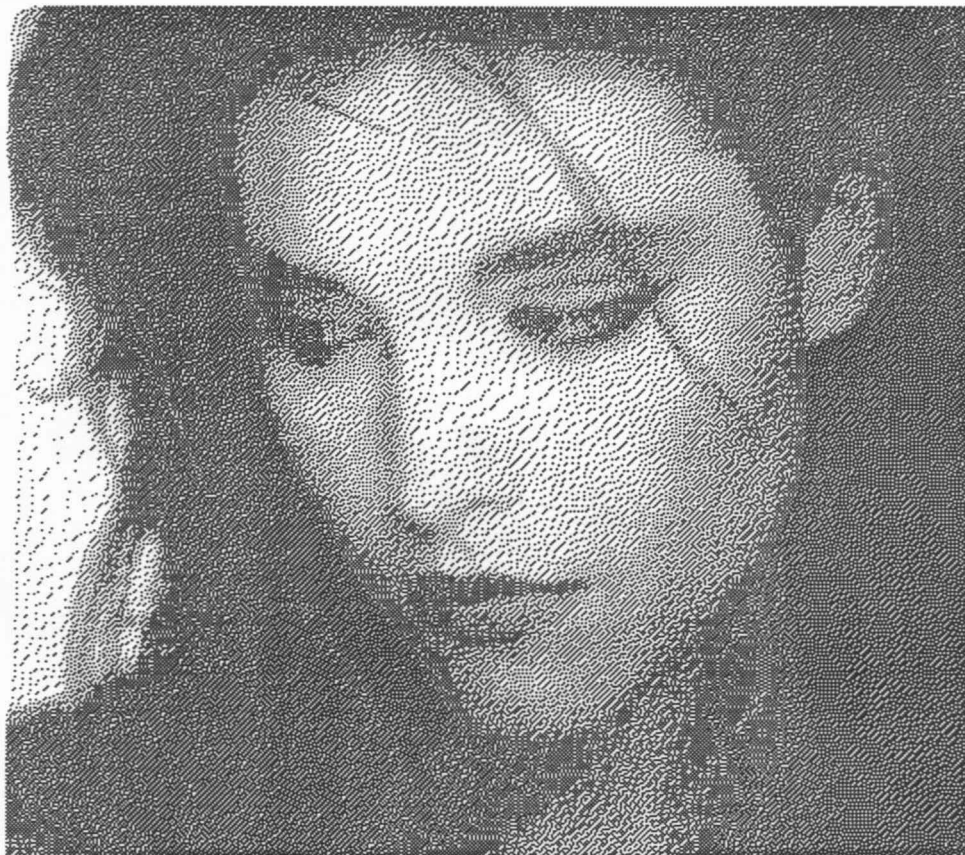
Algoritmus 2.2.2

Pak Floyd-Steinbergův algoritmus pro prahovou hodnotu

$$T = \frac{I_{\max} - I_{\min}}{2}$$

popisuje algoritmus 2.2.2.

Obraz zpracovaný uvedeným algoritmem vykazuje podstatně lepší zobrazení v detailech, viz obr.2.2.3, oproti jednoduchému použití prahové funkce při zobrazování.



Obr. 2.2.3

Tato metoda navíc umožňuje zvýraznit změny kontrastu [21],[76],[172], jsou-li hodnoty $I(x,y)$ upraveny před vlastním zpracováním Floyd-Steinbergovým algoritmem takto

$$I'(x,y) = \frac{I(x,y) - \overset{\wedge}{\alpha} \hat{I}(x,y)}{1 - \alpha}$$

kde: $\hat{I}(x,y)$ je průměrná hodnota intenzit okolí bodu (x,y) .

Pak pro $\alpha > 0$ dostáváme větších rozdílů hodnot mezi sousedními body, tj. zvětšuje se kontrast, přičemž průměrná hodnota intenzit zůstává zachována. Např pro okolí o velikosti 3×3 a $\alpha = 0,9$ dostáváme:

$$I'(x,y) = 9 I(x,y) - \sum_{u=x-1}^{x+1} \sum_{v=y-1}^{y+1} I(x,y)$$

Vzhledem k tomu, že takto získané hodnoty mohou nabývat hodnot mimo interval $I(x,y) \in \langle 0, I_{\max} \rangle$, je nutné vypočtené hodnoty $I'(x,y)$ vhodně upravit, např. pomocí vztahu

$$I''(x,y) = \min \{ I_{\max}, \max \{ 0, I'(x,y) \} \}$$

přičemž I_{\max} je maximální realizovatelná intenzita.

Takto získané hodnoty $I''(x,y)$ se pak zpracují pomocí již dříve uvedeného Floyd-Steinbergova algoritmu.

Zobrazují-li se motivy s homogenním pozadím, pak se doporučuje superponovat obraz náhodným šumem k zamezení pravidelné textury, která by vznikala na pozadí. Ke generaci šumu lze např. použít generátor náhodných čísel tak, že

$$\text{noise} \in \langle \pm 0,05 \frac{I_{\max} - I_{\min}}{2} \rangle$$

Algoritmus 2.2.2 je pak nutno modifikovat takto:

```
if I(x,y) + noise < T then .....
```

Případné zvětšení konstanty 0,05 vede k celkovému zhoršení kvality obrázku.

Metoda dvourozměrného rozptylu chyb

Metoda dvourozměrného rozptylu chyb je rozšířením metody konstantního prahu a Floyd-Steinbergova algoritmu. Ke každé poloze (x,y) obrazu I se přičítá hodnota přenášené chyby, která vyplývá ze statistického stanovení průměru již dříve spočítaných chyb. Rozsah úrovní šedi v obraze zahrnuje hodnoty od I_{\min} do I_{\max} . Přenos chyb je zajištěn pomocí váhové funkce Weight. Váhové koeficienty je vhodné volit jako 2^n pro zlepšení výpočetní účinnosti, např. viz [101].

Pixel je aktivován tehdy, jestliže hodnota odstínu šedi v bodě (x,y) obrazu I včetně přenášené chyby je menší nebo rovna prahové hodnotě, tj.

$$T = \frac{I_{\max} - I_{\min}}{2}$$

Pak příslušné podmínky mají tvar

```
if (I(x,y) + ErrorCarry(x,y)) > T then PutPixel(x,y,White)
```

```
if (I(x,y) + ErrorCarry(x,y)) ≤ T then PutPixel(x,y,Black)
```

Hodnota přenášené chyby je výsledkem ze statistického průměru z dříve spočítaných chyb

$$\text{ErrorCarry}(x,y) = \frac{\sum_{(i,j) \in \text{Area}} \text{Error}(x+i,y+j) * \text{Weight}(i,j)}{\sum_{(i,j) \in \text{Area}} \text{Error}(i,j)}$$

Statistický průměr se počítá pomocí funkce Weight a pole Area, které jsou definovány takto

Weight	Area
.	
. 1 2 4 2 1 .	(-2, -2), (-2, -1), (-2, 0), (-2, 1), (-2, 2)
. 2 4 8 4 2 .	(-1, -2), (-1, -1), (-2, 0), (-2, 1), (-1, 2)
. 4 8 X . . .	(0, -2), (0, -1), X

kde X je aktuální poloha v obraze.

Funkce Carry je závislá na funkci Error, která je závislá na tom, zda byl či nebyl určitý pixel aktivován. V poloze (x,y) se chceme přiblížit úrovni šedi, která je dána hodnotou I(x,y) + přenášená chyba. Je-li pixel v této poloze zobrazen jako černý bod, pak obraz je příliš tmavý a funkce Error je rovna I(x,y) + přenášená chyba - I_{min}. Pokud je pixel v této poloze pixel zobrazen jako bod bílý, pak obraz je příliš světlý a funkce Error je rovna I(x,y) + přenášená chyba - I_{max}.

Funkce Error je pak definována takto:

```
if GetPixel(x,y) = black then
    Error(x,y) := I(x,y) + ErrorCarry(x,y) - Imin
if GetPixel(x,y) = white then
    Error(x,y) := I(x,y) + ErrorCarry(x,y) - Imax
```

Výsledek metody dvourozměrného rozptylu chyb je uveden na obr.2.2.4. Pokud je homogenní pozadí, lze opět superponovat šum k odstranění pravidelné textury na pozadí.



Obr. 2. 2. 4

Dithering

Dithering (někdy též rozmývání) je jinou technikou, která také umožňuje zobrazení při nezměněné rozlišovací schopnosti. Je založena na principu vložení náhodných chyb do obrazu. Chyba je do obrazu vkládána na základě pozorování prahové funkce a hodnoty intenzity pixelu. Přidávání zcela náhodných chyb nevede k požadovanému výsledku, nicméně existuje adaptivní vzor, který vede k uspokojivým výsledkům. Rozhodnutí, zda aktivovat

daný pixel, je závislé nejen na hodnotě příslušné intenzity $I(x,y)$, ale též na matici ${}^{(n)}D$ rozměru $n \times n$, která obsahuje hodnoty od 0 do $n^2 - 1$.

Je-li x, y poloha pixelu a $I(x,y)$ jeho intenzita, pak po určení odpovídajícího prvku ve vzoru o souřadnicích i, j , kde:

$$j = x \bmod n + 1 \qquad i = y \bmod n + 1$$

je daný pixel aktivován jen tehdy, je-li

$${}^{(n)}D_{i,j} < I(x,y)$$

Nejmenší vzor podle [76] je reprezentován maticí 2×2 , a to:

$${}^{(2)}D = \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix}$$

Větší vzory mohou pak být získány pomocí rekurentních vztahů takto:

$${}^{(2n)}D = \left[\begin{array}{c|c} 4 \cdot {}^{(n)}U & 4 \cdot {}^{(n)}D + 2 \cdot {}^{(n)}U \\ \hline 4 \cdot {}^{(n)}D + 3 \cdot {}^{(n)}U & 4 \cdot {}^{(n)}D + {}^{(n)}U \end{array} \right]$$

kde

$${}^{(n)}U = \begin{bmatrix} 1 & \dots & 1 \\ \vdots & & \vdots \\ 1 & \dots & 1 \end{bmatrix}$$

a n je rozměr matice.

Pro $n = 4$ pak dostáváme:

$${}^{(4)}D = \begin{bmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{bmatrix}$$

Celý postup je znázorněn algoritmem 2.2.3.

```

const xmin = 0; xmax = 1023; { rozlišení ve směru x }
      ymin = 0; ymax = 1023; { rozlišení ve směru y }
      n = 4; { velikost matice }
var x, y, i, j: integer;
      D: array [ 0..3 , 0..3 ] of byte
          = ((0, 8, 2, 10), (12, 4, 14, 6), (3, 11, 1, 9), (15, 7, 13, 5));
{ ! matice D se indexuje od 0 - pro zrychlení výpočtu indexu }

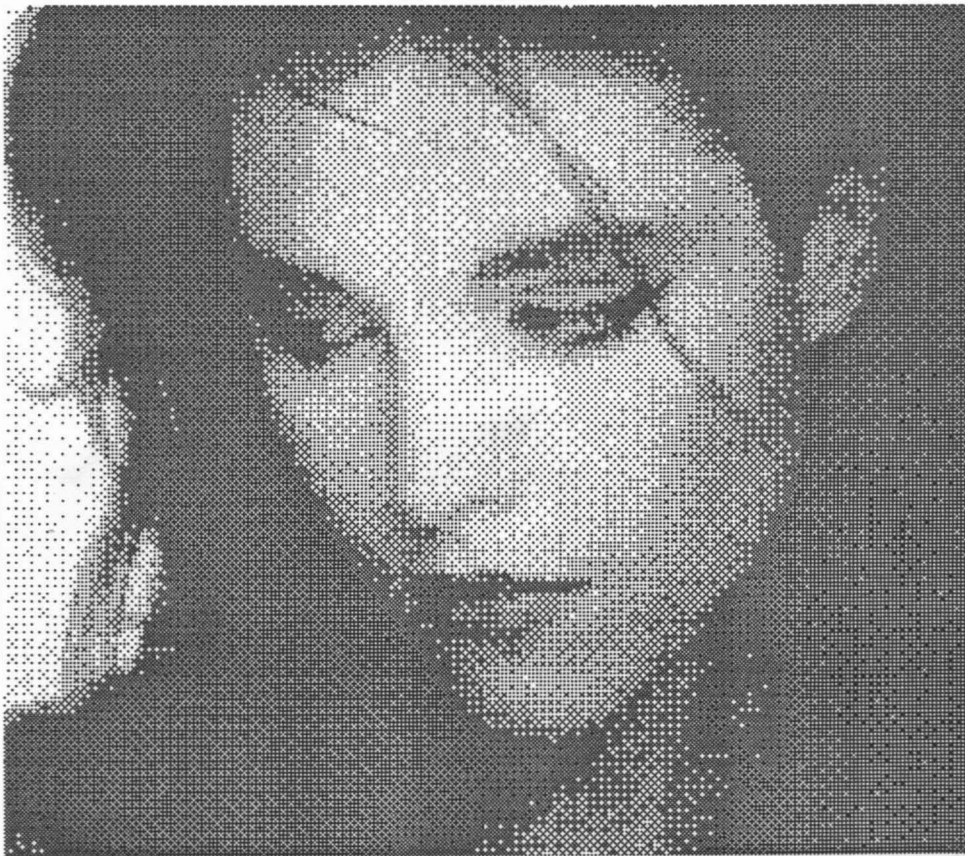
```

```

begin
  for y := ymax downto ymin do
  begin
    i := y mod n;
    for x := xmin to xmax do
    begin { určení polohy (i,j) v matici D }
      j := x mod n;
      if D[i,j] < I(x,y) then PutPixel(x,y,White)
        else PutPixel(x,y,Black);
    end
  end
end
end;

```

Algoritmus 2.2.3



Obr.2.2.5

Výše uvedené algoritmy, které byly původně určeny pro vykreslení daného obrazu pomocí dvou úrovní jasu, je možné modifikovat pro některá výstupní zařízení poskytující více úrovní jasu, např. pro EGA/VGA karty apod. Využitím těchto

možností získáme výstup podstatně lepší kvality. Vzhledem k velké rozšířenosti karty EGA, která poskytuje 4 úrovně šedi, je vhodné na tomto místě uvést modifikaci Floyd-Steinbergova algoritmu.

Předpokládejme, že chceme zobrazit obraz, který má 16 úrovní jasu, pomocí EGA karty, která má k dispozici úrovně Black, Dark, Light a White, viz obr.2.2.6.

```
const Black_gray = 0; Dark_gray = 5;
      Light_gray = 10; White_gray = 15;
      { Definice prahových hodnot }
      { pro ovládání EGA karty pak Black, Gray, Light a White }
var T1,T2,T3: real;
    Error: real;
    x,y: integer;
begin
  T1:= (Dark_gray - Black_gray) / 2;
  T2:= (Light_gray - Dark_gray) / 2;
  T3:= (White_gray - Light_gray) / 2;
  for y := ymax downto ymin do
  for x := xmin to xmax do
  begin { určení hodnoty pixelu pro práh T a chyby Error }
    if I(x,y) < Dark_gray then
      if I(x,y) < T1 then
        begin PutPixel(x,y,Black);
              Error := I(x,y) - Black_gray
        end
      else
        begin PutPixel(x,y,Dark);
              Error := I(x,y) - Dark_gray
        end
    else
      if I(x,y) ≥ Light_gray then
        if I(x,y) > T3 then
          begin PutPixel(x,y,White);
                Error := I(x,y) - White_gray
          end
        else
```

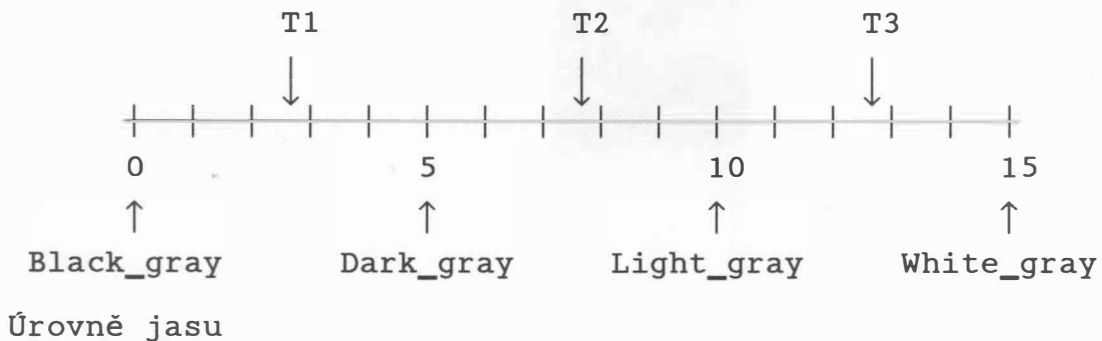
```

begin PutPixel(x,y,Light);
      Error := I(x,y) - Light_gray
end
else { prostřední interval }
  if I(x,y) < T2 then
begin PutPixel(x,y,Dark);
      Error := I(x,y) - Dark_gray
end
  else
begin PutPixel(x,y,Light);
      Error := I(x,y) - Light_gray
end;
  if x < xmax then I(x+1,y) := I(x+1,y) + 3*Error/8;
  if y > ymin then I(x,y-1) := I(x,y-1) + 3*Error/8;
  if (x < xmax) and (y > ymin)
      then I(x+1,y-1) := I(x+1,y-1) + Error/4
end
end
end

```

Algoritmus 2.2.4

Prahové hodnoty



Obr. 2.2.6

Uvedené techniky lze modifikovat i pro barevný výstup na barevné tiskárny, ať inkoustové či laserové. Na obr.1.2.16 je ukázka barevného výstupu při použití modifikované Floyd-Steinbergovy metody. Bohužel zařízení s uvedenými parametry nejsou zatím, pokud je autorovi známo, v ČSFR k dispozici.

Poznámka₁

Matice ${}^{(n)}_D$ a ${}^{(n)}_U$ lze též určit pomocí rekurentních vztahů takto:

$${}^{(1)}_D = [0] \qquad {}^{(1)}_U = [1]$$

pak

$${}^{(2n)}_D = \left[\begin{array}{c|c} {}^4 (n)_D & {}^4 (n)_D + 2 (n)_U \\ \hline {}^4 (n)_D + 3 (n)_U & {}^4 (n)_D + (n)_U \end{array} \right]$$

kde

$${}^{(2n)}_U = \left[\begin{array}{c|c} (n)_U & (n)_U \\ \hline (n)_U & (n)_U \end{array} \right]$$

Poznámka₂

Srovnajte matice ${}^{(4)}_D$ s maticí vzorů ${}^{(4)}_T$ a najděte souvislosti.

Příklad

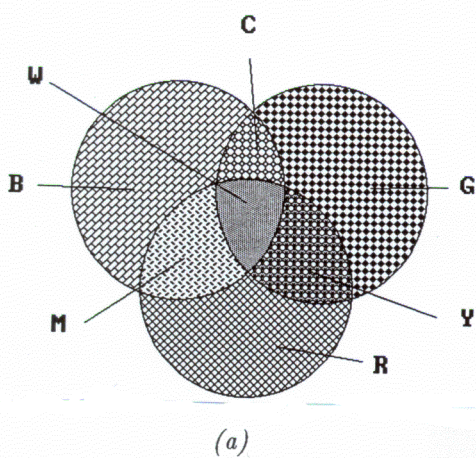
Pro Floyd-Steinbergův algoritmus navrhněte i jiná schémata distribuce chyby, paralelizujte výpočet chyby a experimentálně ukažte jejich vlastnosti.

2.3. Chromatické světlo

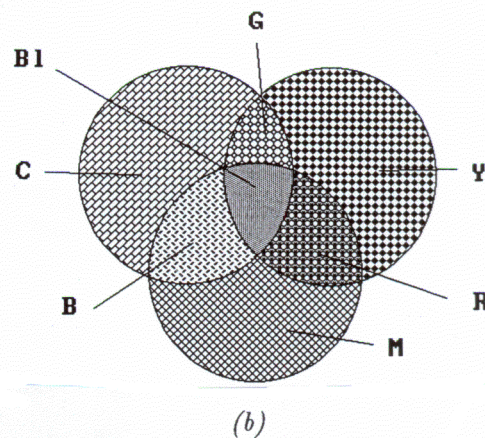
Vidíme-li světlo určité vlnové délky, získáváme určitý vjem barvy. Přirozené zdroje světla neobsahují pouze jednu určitou vlnovou délku, i když vnímáme určitou barvu. Je známo., že většina reálných barev může být tvořena pomocí základních barev, a to:

- červené (R-Red), zelené (G-Green) a modré (B-Blue) v aditivním RGB systému, viz obr.2.3.1.a

- modrozelené (C-Cyan), purpurové (M-Magenta), žluté (Z-Zellow) v subtraktivním barevném systému, viz obr.2.3.1.b



obr.2.3.1.a



obr.2.3.1.b

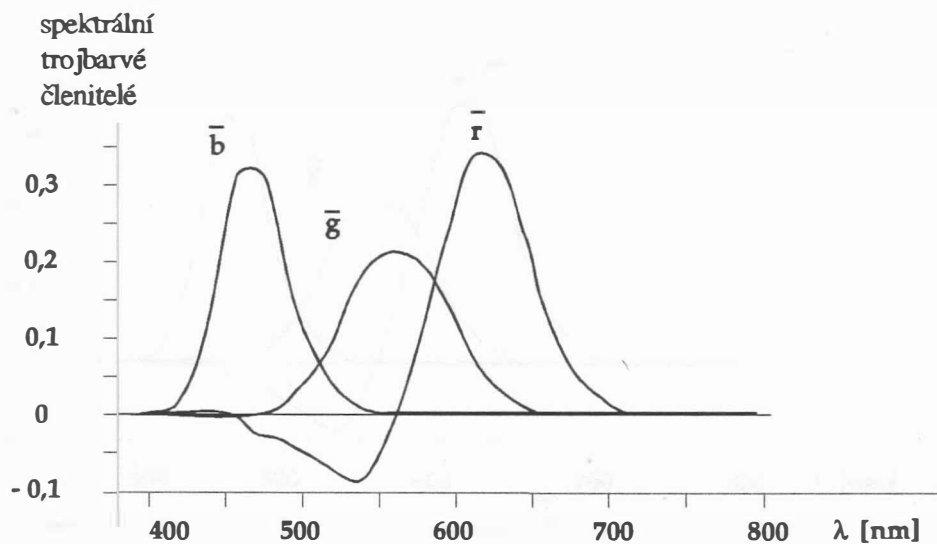
Aditivní systém RGB se nejčastěji používá pro barevné monitory (jde o sčítání světelných zdrojů), zatímco subtraktivní systém CMZ se používá při záznamu např. na filmový materiál nebo na papír pomocí barevných inkoustů (barvy se odečítají od bílého pozadí). I když je možné černou barvu získat kombinací všech tří základních složek (modrozelené, purpurové a žluté), bývá čttná barva k dispozici samostatně.

Mezinárodní komise pro osvětlení CIE (Commission International de l'Eclairage u nás též známá pod zkratkou MKO) stanovila vlnové délky základních barev takto:

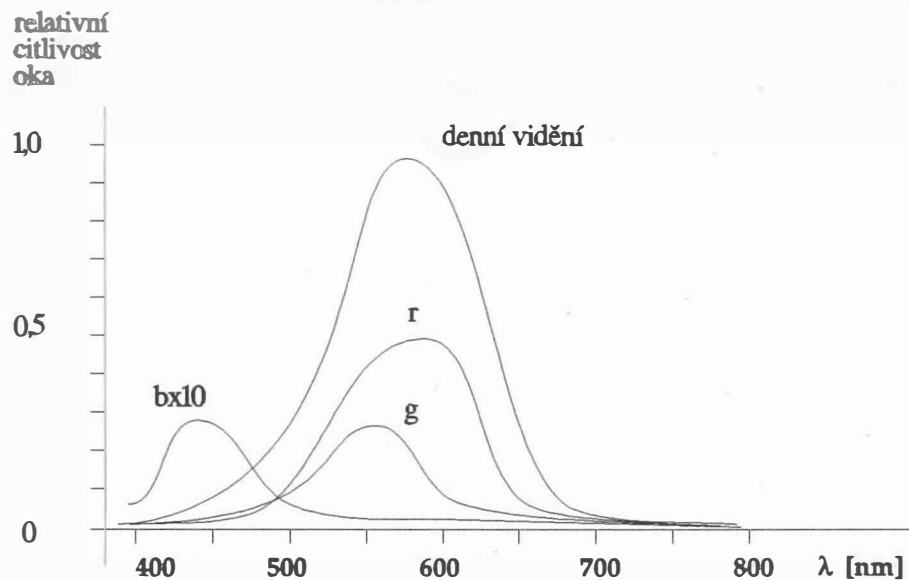
červená (R)	780,0 nm
zelená (G)	546,1 nm
modrá (B)	435,8 nm

System RGB

Jsou-li za základní barvy vybrány barvy červená (R), zelená (G) a modrá (B), pak téměř všechny barvy jsou vyjádřitelné pomocí váhového součtu jednotlivých složek. Vezmeme-li v úvahu viditelné spektrum vlnových délek, pak každé vlnové délce odpovídá jistá barva. Na obr.2.3.2 jsou ukázány průběhy jednotlivých váhových koeficientů r , g , b , které se též nazývají trichromatickými spektrálními členiteli. Jejich hodnoty, získané kolorimetrickými měřeními, jsou určeny tak, aby chom obdrželi příslušné barvy odpovídající jednotlivým vlnovým délkám ve viditelném spektru (380 - 780 nm).



Obr. 2. 3. 2



Obr. 2. 3. 3

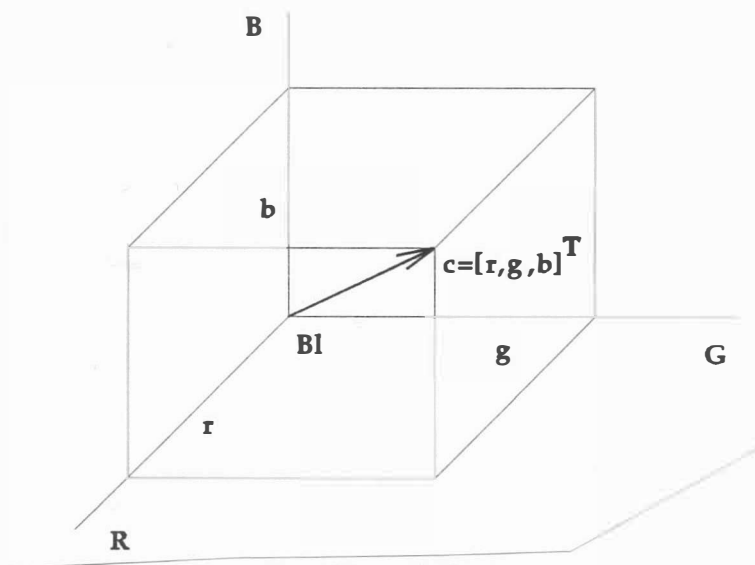
Poznamenejme, že trichromatické spektrální členitele závisejí na daném pozorovateli. Relativní citlivost oka v závislosti na vlnové délce je znázorněna diagramem na obr.2.3.3, přičemž je na místě poznamenat, že lidské oko je schopno rozlišit asi 350 000 různých odstínů barev. Z obr.2.3.2 vyplývá, že ne všechny barvy jsou reprezentovatelné pomocí smísení tří základních barev, neboť pro určitou část vlnových délek viditelného spektra je hodnota koeficientů záporná, zejména pak pro barvu červenou.

V případě, že barva c je tvořena vektorem $[r, g, b]^T$ trichromatických spektrálních členitelů, jehož složka r má zápornou hodnotu, pak barva získaná sloučením barvy c a barvy reprezentované vektorem $[-r, 0, 0]^T$ odpovídá barvě s vektorem $[0, g, b]^T$. Lze tedy říci, že při kladných koeficientech $[r, g, b]^T$ je barva definována jako:

$$c = rR + gG + bB$$

Při záporném koeficientu r je však nutné provádět kolorimetrické srovnání tak, že přimísíme k barvě c ještě červenou složku, a tedy:

$$c + rR = gG + bB$$



Obr. 2. 3. 4

Z uvedených vztahů vyplývá, že smísíme-li dvě barvy

$$c_1 = [r_1, g_1, b_1]^T \quad \text{a} \quad c_2 = [r_2, g_2, b_2]^T$$

pak výsledná barva C je dána součtem jednotlivých složek, tj.:

$$c = [r_1 + r_2 , g_1 + g_2 , b_1 + b_2]^T$$

přičemž výsledný jas je samozřejmě dán součtem jednotlivých jasů. Jestliže každé složce vektoru přiřadíme osu v třírozměrném souřadném systému, které označíme R, G, B, pak lze vektor barvy prostorově znázornit jako tělesovou úhlopříčku kvádru, viz obr. 2.3.4.

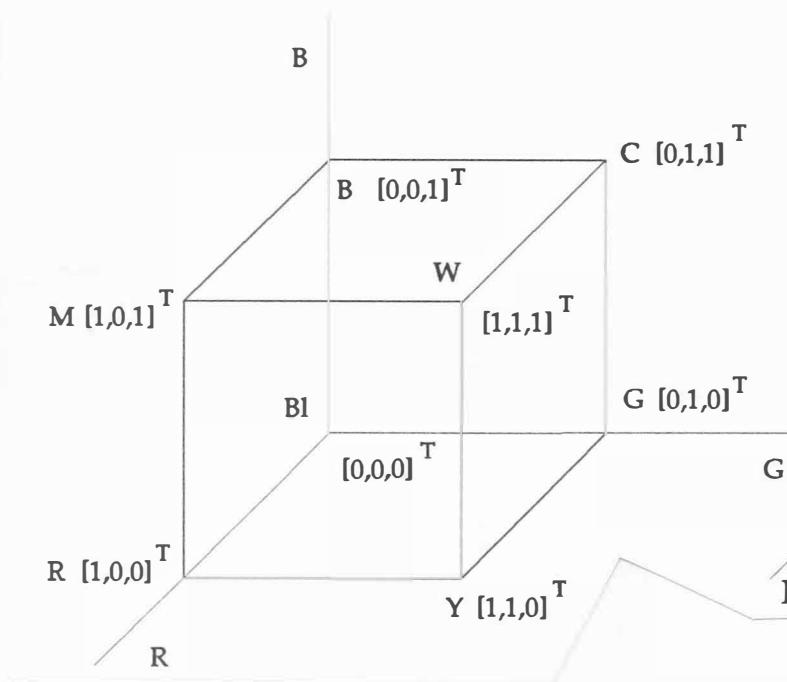
Sníčováním a zvyšováním jasu se vlastně mění jen délka vektoru c , zatímco při změně barvy se pak mění prostorový směr vektoru barvy. Pokud jednotlivé složky r, g, b nabývají hodnot od 0 do 1, lze si barevný prostor představit ve tvaru jednotkové krychle, kde vrchol $[1 , 1 , 1]^T$ odpovídá bílé barvě, viz obr. 2.3.5. Vektor bílé barvy prochází vrcholem krychle a tvoří tělesovou úhlopříčku krychle RGB. Je nutné si uvědomit, že stejně dlouhé vektory různých barev neodpovídají stejným jasům, neboť pro poměr jednotlivých složek

$$r : g : b = 1 : 1 : 1$$

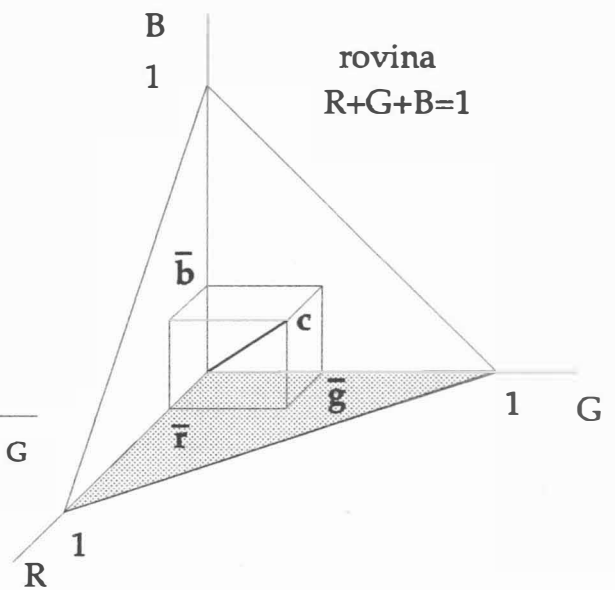
dostáváme poměr odpovídajících jasů:

$$1 : 4,6 : 0,06$$

(jas se udává v cd/m^2), a že jednotková rovina není rovinou konstantního jasu.



Obr. 2.3.5



Obr. 2.3.6

V předchozím textu nabývaly jasové úrovně rozsahu hodnot od 0 do 1. Tuto konvenci je vhodné dodržet. Promítneme-li trichromatické souřadnice na jednotkovou rovinu, viz obr.2.3.6, pak dostáváme

$$\bar{r} = \frac{r}{r + g + b} \quad \bar{g} = \frac{g}{r + g + b} \quad \bar{b} = \frac{b}{r + g + b}$$

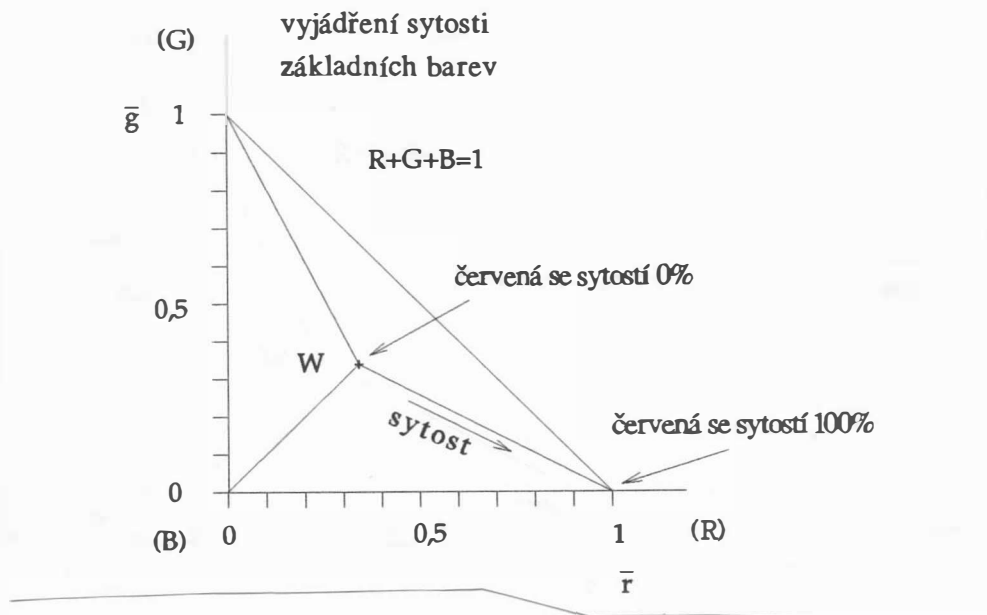
a tedy:

$$\bar{r} + \bar{g} + \bar{b} = 1$$

Nyní je zřejmé, že v normalizovaném tvaru je možné vyjádřit třetí složku ze dvou známých složek. Pro určení druhu barvy, tj. chromatičnosti (tónu a sytosti bez udání jasu), tedy stačí rovinné znázornění barvy. Takovým zobrazením by mohl být rovnoramenný trojúhelník, kterým je např. jednotková rovina, v jehož středu by byla bílá barva a ve vrcholech pak základní barvy RGB. Přehlednějším znázorněním je však pravoúhlý trojúhelník, který vznikne promítnutím jednotkové roviny do roviny R-G. Pak druh barvy je zcela určen koeficienty příslušujícími jednotlivým osám R-G, neboť platí:

$$\bar{b} = 1 - \bar{r} - \bar{g}$$

Pak poloha bodu reprezentujícího barvu bílou je v těžišti pravoúhlého trojúhelníka, viz obr.2.3.7, přičemž sytost barvy je dána vzdáleností bodu reprezentujícího barvu od bodu W reprezentující barvu bílou.



Máme-li dvě barvy, které jsou dány

$$c_1 = [r_1 , g_1 , b_1]^T \quad c_2 = [r_2 , g_2 , b_2]^T$$

pak výsledná barva je dána součtem jednotlivých složek, tj.:

$$c_3 = [r_1 + r_2 , g_1 + g_2 , b_1 + b_2]^T$$

Po promítnutí na jednotkovou rovinu dostáváme

$$\bar{c}_1 = [\bar{r}_1 , \bar{g}_1 , \bar{b}_1]^T \quad \bar{c}_2 = [\bar{r}_2 , \bar{g}_2 , \bar{b}_2]^T$$

$$\bar{c}_3 = [\bar{r}_3 , \bar{g}_3 , \bar{b}_3]^T$$

přičemž:

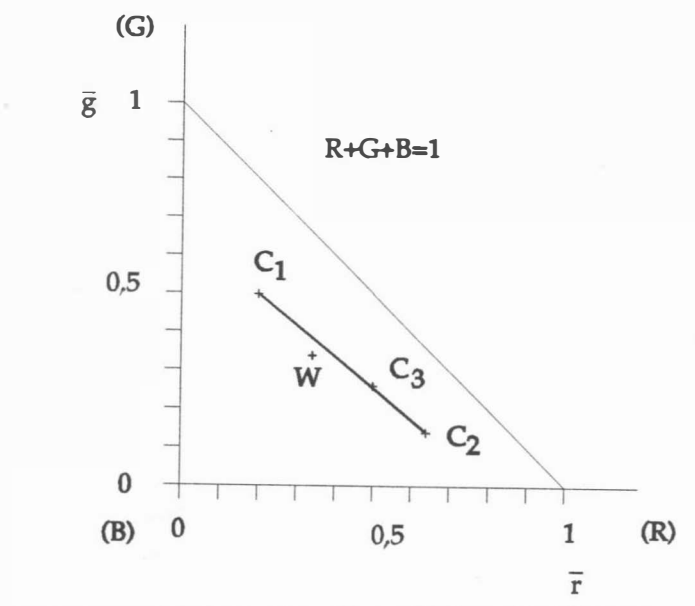
$$\bar{r}_1 = \frac{r_1}{T_1} \quad \bar{r}_2 = \frac{r_2}{T_2} \quad \bar{r}_3 = \frac{r_3}{T_3}$$

$$\bar{g}_1 = \frac{g_1}{T_1} \quad \bar{g}_2 = \frac{g_2}{T_2} \quad \bar{g}_3 = \frac{g_3}{T_3}$$

$$\bar{b}_1 = \frac{b_1}{T_1} \quad \bar{b}_2 = \frac{b_2}{T_2} \quad \bar{b}_3 = \frac{b_3}{T_3}$$

kde: $T_1 = r_1 + g_1 + b_1$ $T_2 = r_2 + g_2 + b_2$

$$T_3 = r_3 + g_3 + b_3$$



Obr. 2. 3. 8

Z výše uvedených rovnic lze odvodit tzv. pákové pravidlo, tj.:

$$T_2 \cdot (\bar{r}_2 - \bar{r}_3) = T_1 \cdot (\bar{r}_3 - \bar{r}_1)$$

$$T_2 \cdot (\bar{g}_2 - \bar{g}_3) = T_1 \cdot (\bar{g}_3 - \bar{g}_1)$$

$$T_2 \cdot (\bar{b}_2 - \bar{b}_3) = T_1 \cdot (\bar{b}_3 - \bar{b}_1)$$

Úpravou dostáváme, že:

$$\bar{r}_3 = \frac{\bar{r}_1 T_1 + \bar{r}_2 T_2}{T_1 + T_2} \quad \bar{g}_3 = \frac{\bar{g}_1 T_1 + \bar{g}_2 T_2}{T_1 + T_2}$$

$$\bar{b}_3 = \frac{\bar{b}_1 T_1 + \bar{b}_2 T_2}{T_1 + T_2}$$

Lze tedy říci, že výsledná barva je středem dvoujramenné páky, kde se součin síly (hodnota T_2 , resp. T_1) a délky páky ($\bar{r}_2 - \bar{r}_3$, resp. $\bar{r}_3 - \bar{r}_1$ pro osu červené) sobě rovnají.

Příklad

Smícháme-li dvě barvy, které jsou určeny vektory:

$$c_1 = [4 , 10 , 6]^T \quad c_2 = [26 , 6 , 8]^T$$

pak výsledná barva je určena vektorem:

$$c_3 = [30 , 16 , 14]^T$$

Po promítnutí na jednotkovou rovinu dostáváme pro jednotlivé barvy vektory:

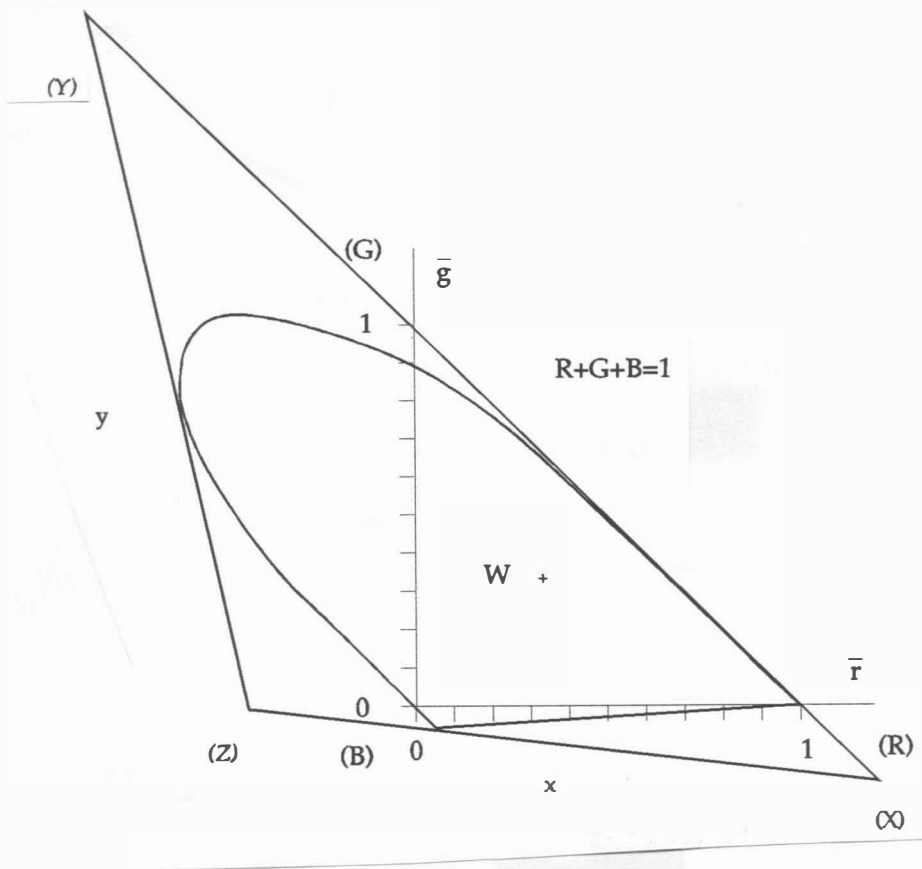
$$\bar{c}_1 = [0,2 , 0,5 , 0,3]^T \quad \bar{c}_2 = [0,65 , 0,15 , 0,2]^T$$

$$\bar{c}_3 = [0,5 , 0,27 , 0,23]^T$$

Poloha jednotlivých bodů v R-G rovině je znázorněna na obr.2.3.8.

Systémy XYZ, CIE-xy, CIE-uv

Nevýhodou barevného systému RGB jsou zejména záporné hodnoty trichromatického členitele pro spektrální barvy od vlnové délky 436 nm do 546 nm, tj. pro určité zelené a modré barvy (viz obr.2.3.9.b), přičemž jas je nutné řešit výpočtem.



Obr. 2.3.9

Zavedením nových virtuálních souřadnic XYZ se vyhneme problému záporných koeficientů, avšak některé barvy mají sytost větší než 100%, a jde tedy o barvy neskutečné (nelze je získat rozkladem bílého světla). Komisí CIE byl definován též převod mezi RGB systémem a systémem XYZ virtuálních souřadnic takto:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 2,7689 & 1,7518 & 1,1302 \\ 1,0000 & 4,5907 & 0,0601 \\ 0,0000 & 0,0565 & 5,5943 \end{bmatrix} \cdot \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

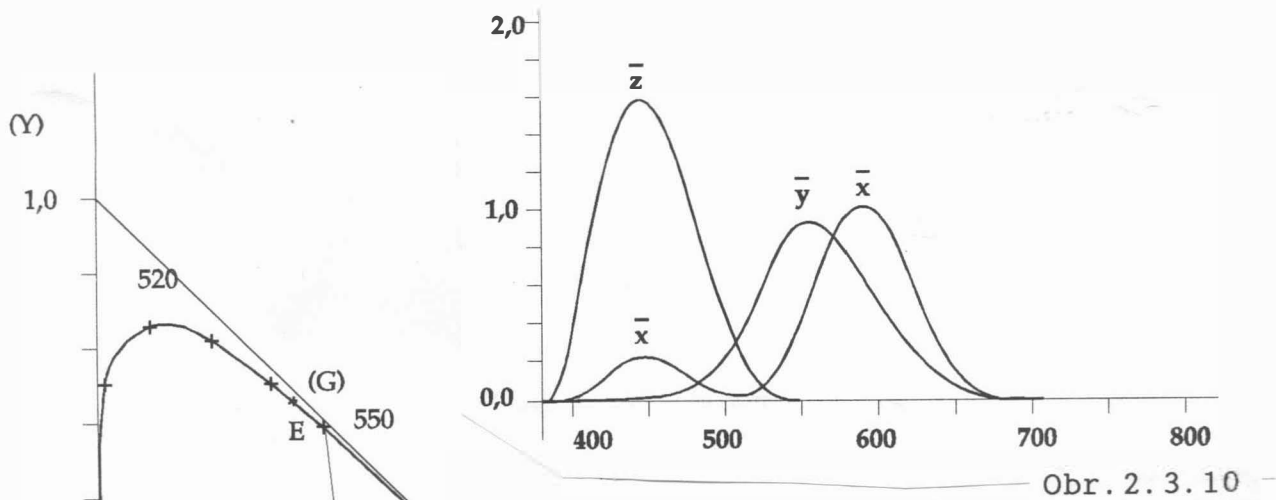
přičemž výsledný jas je určen složkou Y. Promítneme-li opět souřadnice XYZ na jednotkovou rovinu, pak:

$$x = \frac{X}{X + Y + Z} \quad y = \frac{Y}{X + Y + Z} \quad z = \frac{Z}{X + Y + Z}$$

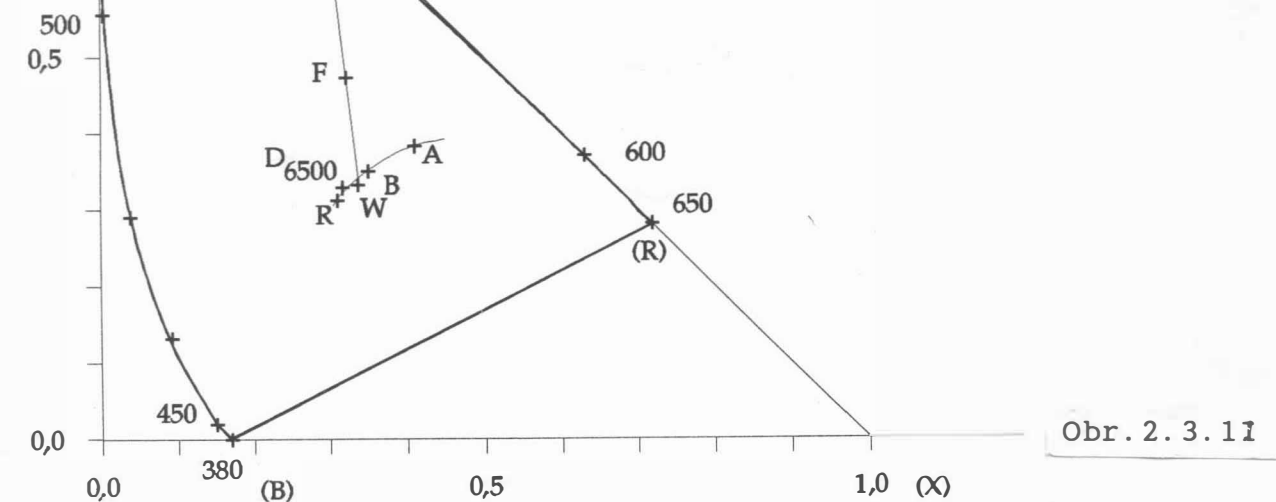
přičemž:

$$x + y + z = 1$$

Souřadnici z není nutné uvádět, neboť $z = 1 - x - y$. Takto získáme CIE-xy souřadný systém.



Obr. 2.3.10



Obr. 2.3.11

Nevýhodou CIE systému je, že neobsahuje informaci o jasu. Proto je každá barva reprezentována vektorem $[x, y, Y]^T$ a zpětný přepočítání je pak dán vztahy:

$$X = x \frac{Y}{Y} \quad Y = y \quad Z = (1 - x - y) \frac{Y}{Y}$$

Přepočtením trichromatických spektrálních členitelů v RGB systému z obr.2.3.2 do systému XYZ dostaneme trichromatické spektrální členitele v XYZ systému, jejichž závislost na vlnové délce je znázorněna na obr.2.3.10. Protože trichromatický členitel Y je nositelem jasu, je jeho průběh totožný s křivkou jasové citlivosti oka, viz obr.2.3.3.

Zobrazíme-li nyní jednotlivé barvy po promítnutí na jednotkovou rovinu pouze v rovině x-y, pak obdržíme standardní CIE-xy barevný diagram, který je zobrazen na obr.2.3.11. Bílé izoenergetické světlo, označené W, je definováno souřadnicemi $x = 1/3$ a $y = 1/3$, zatímco normalizované světlo D_{6500} odpovídající bílému světlu obrazovky má souřadnice $x = 0,313$ a $y = 0,329$. Bílé světlo odpovídající bodu R užívané v NTSC systému (systém barevného televizního vysílání používaný zejména v USA) má souřadnice $x = 0,310$ a $y = 0,316$.

Sytost barvy F je dána poměrem vzdáleností WF/WE, přičemž bod E reprezentuje danou barvu se sytostí 100%. Z toho vyplývá, že mísením barvy E s bílou barvou dostáváme tutéž barvu s menší sytostí, které odpovídá určitý bod na spojnici WE. Tón barvy je určen náhradní vlnovou délkou barvy F, která je určena průsečíkem polopřímky WF a čáry spektrálních barev (vlnovou délku je možné např. odečíst z CIE-xy diagramu).

Každé dvě barvy, které smísením dávají bílé světlo, se nazývají barvami doplňkovými. Pro smísení dvou barev platí tzv. první Grassmanův zákon. Jsou-li dány dvě barvy v systému CIE reprezentované vektory:

$$c_1 = [x_1 , y_1 , Y_1]^T \quad c_2 = [x_2 , y_2 , Y_2]^T$$

pak barva, která vznikne jejich smísením, je dána vektorem:

$$c_3 = [x_3 , y_3 , Y_3]^T$$

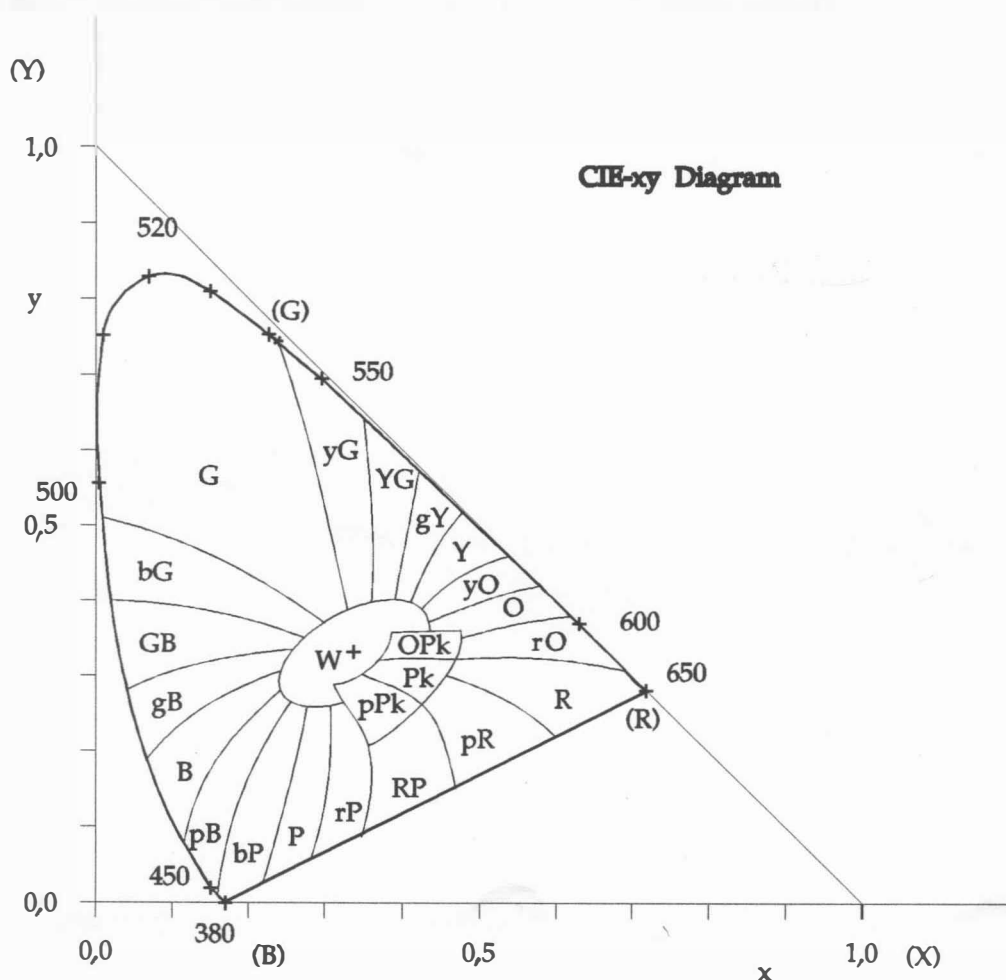
kde

$$x_3 = \frac{x_1 T_1 + x_2 T_2}{T_1 + T_2} \quad y_3 = \frac{y_1 T_1 + y_2 T_2}{T_1 + T_2}$$

přičemž

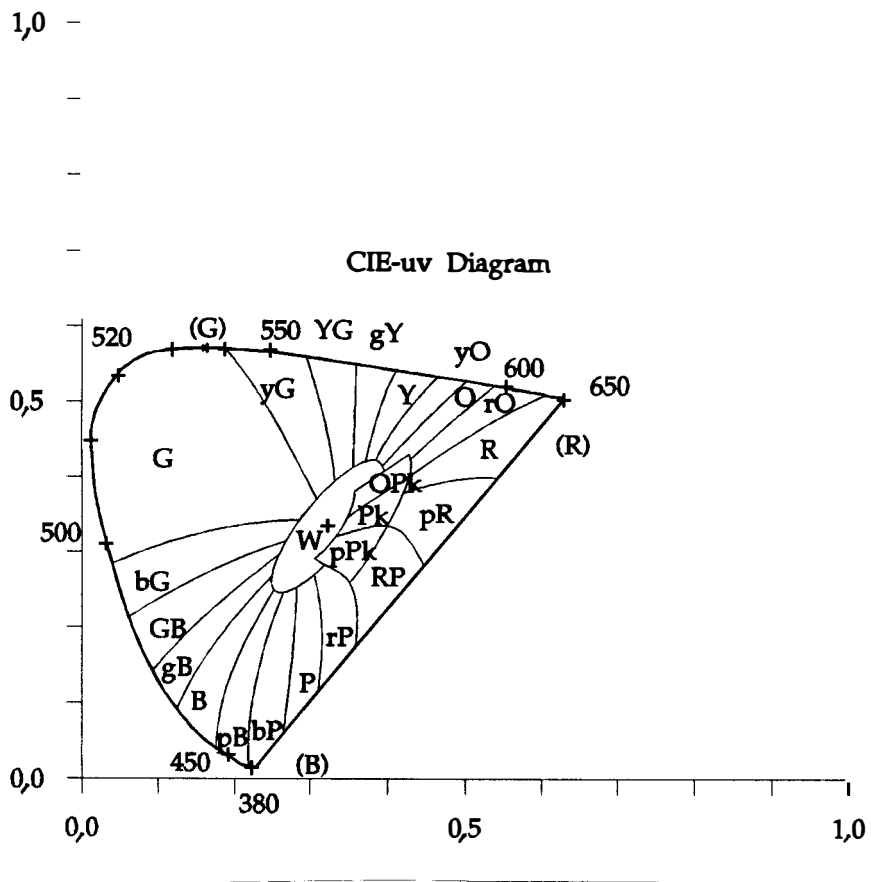
$$T_1 = \frac{Y_2}{Y_1} \quad T_2 = \frac{Y_1}{Y_2} \quad Y_3 = Y_1 + Y_2$$

Při mísení více barev lze postupovat tak, že výše uvedený postup aplikujeme postupně na všechny barvy, ať už původní, či vzniklé smísením ostatních barev.



Obr. 2. 3. 12. a

V systému XYZ i v CIE-xy trojúhelníku, viz obr.2.3.12.a, neodpovídají stejné lineární vzdálenosti v různých místech prostoru nebo trojúhelníku stejným subjektivně vnímaným rozdílům vjemu barvy a naopak. Proto tam, kde je nutné chromatičnost světla nebo předmětů reprezentovat charakteristikami odpovídajícími subjektivním vjemům, doporučuje se používat znázornění pomocí systému stejných barevných rozdílů, tj. systému CIE-uv (1960) nebo systému CIE-Uvw (1964), kde v celém prostoru nebo rovině stejným subjektivně vnímaným rozdílům vjemů barvy odpovídají přibližně stejné vzdálenosti (podrobně viz [89], [215]). Diagram systému CIE-uv (v literatuře označovaný jako CIE Uniform Color Space) je znázorněn na obr.2.3.12.b.



Obr. 2.3.12.b

Pro systém CIE-UVW pak platí převodní vztahy:

$$W = 25 Y^{1/3} - 17 \quad U = 13 W (u' - u'_w)$$

$$V = 13 W (v' - v'_w)$$

kde u'_w , v'_w jsou hodnoty u' , v' pro smluvní bílé světlo.

Pro převod ze systému XYZ byly definovány vztahy:

$$u' = \frac{4X}{X + 15Y + 3Z} \quad v' = \frac{9Y}{X + 15Y + 3Z}$$

Pro převod z trichromatických souřadnic xy byly definovány převodní vztahy:

$$u' = \frac{4x}{-2x + 12y + 3} \quad v' = \frac{9y}{-2x + 12y + 3}$$

Diagram pro systém CIE-uv je definován v souřadnicích u' a v' .

V literatuře lze nalézt i původně definovaný systém, který však již není ČSN uváděn (proměnné jsou značeny bez znaku '),

jehož převodní vztahy byly určeny takto:

$$u = \frac{4x}{-2x + 12y + 3} \quad v = \frac{6y}{-2x + 12y + 3}$$

Příklad

Určete CIE barevné souřadnice barvy vzniklé smísením tří barev reprezentovaných vektory:

$$c_1 = [0,1, 0,3, 10]^T \quad c_2 = [0,35, 0,2, 10]^T$$

$$c_3 = [0,2, 0,05, 10]^T$$

Aplikací prvního Grassmanova zákona dostáváme, že barva vzniklá smísením barev c_1 a c_2 je určena vektorem:

$$c_4 = [x_4, y_4, Y_4]^T$$

příčemž

$$T_1 = \frac{Y_1}{y_1} = \frac{10}{0,3} = 33,33 \quad T_2 = \frac{Y_2}{y_2} = \frac{10}{0,2} = 50$$

pak

$$x_4 = \frac{x_1 T_1 + x_2 T_2}{T_1 + T_2} = 0,25$$

$$y_4 = \frac{y_1 T_1 + y_2 T_2}{T_1 + T_2} = 0,24$$

$$Y_4 = Y_1 + Y_2 = 20$$

tj. $c_4 = [0,25, 0,24, 20]^T$

Nyní je nutné smísit barvu c_3 a c_4 . Pak výsledná barva c_5 je určena vektorem

$$c_5 = [x_5, y_5, Y_5]^T$$

příčemž

$$T_3 = \frac{Y_3}{y_3} = \frac{10}{0,05} = 200$$

$$T_4 = \frac{Y_4}{y_4} = \frac{20}{0,24} = 83,3$$

$$x_5 = \frac{x_3 T_3 + x_4 T_4}{T_3 + T_4} = 0,215$$

$$Y_5 = \frac{Y_3 T_3 + Y_4 T_4}{T_3 + T_4} = 0,106$$

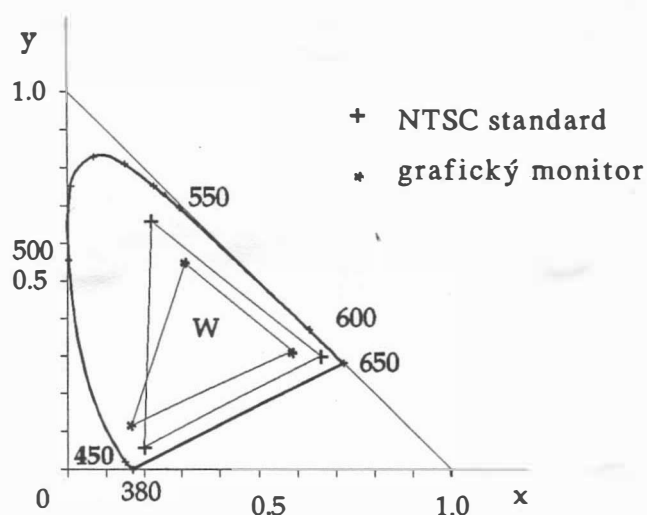
$$Y_5 = Y_3 + Y_4 = 20 + 10 = 30$$

Výsledná barva vzniklá smísením barev c_1 , c_2 a c_3 je dána v CIE systému souřadnicemi:

$$c_5 = [0,215 , 0,106 , 30]^T$$

Rozdělení jednotlivých barev v diagramech CIE-xy a CIE-uv je znázorněno na obr. 2.3.12.

- | | |
|-----------------------|-----------------------------|
| 1 R červená | 12 gB zelenavě modrá |
| 2 rO oranžově červená | 13 B modrá |
| 3 O oranžová | 14 pB purpurově modrá |
| 4 yO žlutooranžová | 15 bP modře purpurová |
| 5 Y žlutá | 16 P purpurová |
| 6 gY nazelenale žlutá | 17 rP načervenalé purpurová |
| 7 YG žlutozelená | 18 RP červeně purpurová |
| 8 yG nažloutle zelená | 19 pR purpurově červená |
| 9 G zelená | 20 pPk růžově purpurová |
| 10 bG modravě zelená | 21 Pk růžová |
| 11 BG modrozelená | 22 OPk oranžově růžová |
| 23 iluminační oblast | |



Obr. 2.3.13

I když by bylo možné volit tři základní barvy tak, aby obsáhly co největší plochu CIE-xy diagramu, viz tab.2.3.1, jsou zvoleny tak, aby luminofory byly vyrobitelné a energetické nároky nebyly příliš vysoké. Zakreslením bodů reprezentujících jednotlivé barvy do CIE-xy diagramu dostáváme palety barev zobrazitelných na daném zařízení, viz obr.2.3.13.

	CIE		NTSC		bar. monitor	
	x	y	x	y	x	y
R	0,735	0,265	0,670	0,330	0,628	0,346
G	0,274	0,717	0,210	0,710	0,268	0,588
B	0,167	0,009	0,140	0,080	0,150	0,070
W	0,448	0,408	0,310	0,316	0,313	0,329

tab.2.3.1

Převod mezi RGB a CIE systémem je obecně dán vztahem:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix} \cdot \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

kde $[X_r , Y_r , Z_r]^T$ reprezentuje bod v diagramu CIE, který odpovídá bodu $[r , 0 , 0]^T$ v systému RGB. Jsou-li známy hodnoty x, y pro jednotlivé RGB základní barvy, pak:

$$x_r = \frac{X_r}{X_r + Y_r + Z_r} = \frac{X_r}{T_r}$$

$$y_r = \frac{Y_r}{X_r + Y_r + Z_r} = \frac{Y_r}{T_r}$$

$$z_r = 1 - x_r - y_r = \frac{Z_r}{X_r + Y_r + Z_r}$$

Analogicky pro X_g, Y_g, Z_g a X_b, Y_b, Z_b s tím, že:

$$T_r = X_r + Y_r + Z_r$$

$$T_g = X_g + Y_g + Z_g$$

$$T_b = X_b + Y_b + Z_b$$

Lze tedy psát:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x_r T_r & x_g T_g & x_b T_b \\ Y_r T_r & Y_g T_g & Y_b T_b \\ (1-x_r-Y_r) \cdot T_r & (1-x_g-Y_g) \cdot T_g & (1-x_b-Y_b) \cdot T_b \end{bmatrix} \cdot \begin{bmatrix} r \\ g \\ b \end{bmatrix} \quad (\xi)$$

V maticovém tvaru lze psát:

$$\mathbf{x} = \mathbf{T} \cdot \mathbf{r}$$

Aby transformace byla zcela definována, je nutné určit hodnoty T_r , T_g a T_b . Jsou-li známy složky odpovídající jasům Y_r , Y_g a Y_b , pak platí:

$$T_r = \frac{Y_r}{Y_r} \quad T_g = \frac{Y_g}{Y_g} \quad T_b = \frac{Y_b}{Y_b}$$

Jestliže jsou známy souřadnice $[X_w, Y_w, Z_w]^T$ pro bílou barvu, pak řešením rovnice (ξ), kde

$$\mathbf{x} = [X_w, Y_w, Z_w]^T \quad \text{a} \quad \mathbf{r} = [T_r, T_g, T_b]^T$$

dostaneme požadované hodnoty. Jsou-li však k dispozici chromatické souřadnice, tj. vektor $[x_w, y_w, z_w]^T$, pak platí:

$$T_r = \frac{Y_w}{Y_w} \cdot [x_w(Y_g - Y_b) - Y_w(x_g - x_b) + x_g Y_b - x_b Y_g] / D$$

$$T_g = \frac{Y_w}{Y_w} \cdot [x_w(Y_b - Y_r) - Y_w(x_b - x_r) - x_r Y_b + x_b Y_r] / D$$

$$T_b = \frac{Y_w}{Y_w} \cdot [x_w(Y_r - Y_g) - Y_w(x_r - x_g) + x_r Y_g - x_g Y_r] / D$$

kde:

$$D = x_r \cdot (Y_g - Y_b) + x_g \cdot (Y_b - Y_r) + x_b \cdot (Y_r - Y_g)$$

Inverzní transformace z XYZ souřadného systému do systému RGB je dána transformací:

$$\mathbf{r} = \mathbf{T}^{-1} \cdot \mathbf{x} = \mathbf{W} \cdot \mathbf{x}$$

kde:

$$w_{11} = [(Y_g - Y_b) - x_b \cdot Y_g + Y_b \cdot x_g] / (T_r \cdot D)$$

$$w_{12} = [(x_b - x_g) - x_b \cdot Y_g + x_g \cdot Y_b] / (T_r \cdot D)$$

$$w_{13} = [x_g \cdot Y_b - x_b \cdot Y_g] / (T_r \cdot D)$$

$$w_{21} = [(Y_b - Y_r) - Y_b \cdot x_r + Y_r \cdot x_b] / (T_g \cdot D)$$

$$w_{22} = [(x_r - x_b) - x_r \cdot Y_b + x_b \cdot Y_r] / (T_g \cdot D)$$

$$w_{23} = [x_b \cdot Y_r - x_r \cdot Y_b] / (T_g \cdot D)$$

$$w_{31} = [(Y_r - Y_g) - Y_r \cdot x_g + Y_g \cdot x_r] / (T_b \cdot D)$$

$$w_{32} = [(x_g - x_r) - x_g \cdot Y_r + x_r \cdot Y_g] / (T_b \cdot D)$$

$$w_{33} = [x_r \cdot Y_g - x_g \cdot Y_r] / (T_b \cdot D)$$

Z výše uvedeného je zřejmé, že matice převodu závisí nejen na souřadnicích základních barev, ale též na poloze bodu odpovídajícímu bílé barvě v CIE diagramu.

Poznámka

D je dvojnásobek plochy trojúhelníka vymezeného souřadnicemi základních barev v CIE-xy diagramu.

Příklad

Určete CIE-xy chromatické souřadnice barvy dané vektorem $[255 , 0 , 0]^T$ v RGB systému, má-li monitor níže uvedené souřadnice pro trichromatické členitele [109]

$$x_r = 0,628 \quad x_g = 0,268 \quad x_b = 0,150$$

$$Y_r = 0,346 \quad Y_g = 0,588 \quad Y_b = 0,070$$

a jsou-li souřadnice pro bílou barvu:

$$x_w = 0,313 \quad Y_w = 0,329 \quad Y_w = 1,0$$

Pak platí

$$D = x_r \cdot (Y_g - Y_b) + x_g \cdot (Y_b - Y_r) + x_b \cdot (Y_r - Y_g) = 0,215$$

$$D \cdot T_r / (Y_w / Y_w) = x_w \cdot (Y_g - Y_b) - Y_w \cdot (x_g - x_b) + x_g \cdot Y_b - x_b \cdot Y_g \\ = 0,0539$$

a tedy:

$$T_r = 0,762$$

Analogicky

$$T_g = 1,114 \quad T_b = 1,164$$

Dosazením do rovnice

$$\mathbf{x} = \mathbf{T} \cdot \mathbf{r}$$

dostáváme:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0,478 & 0,299 & 0,175 \\ 0,263 & 0,655 & 0,081 \\ 0,020 & 0,160 & 0,908 \end{bmatrix} \cdot \begin{bmatrix} 255 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 121,94 \\ 67,19 \\ 5,05 \end{bmatrix}$$

Matice převodu je poněkud odlišná od dříve uvedené matice vlivem různosti chromatických souřadnic pro trichromatické členitele a barvu bílou. CIE chromatické souřadnice dané barvy pak jsou

$$x = \frac{\cancel{x} X}{\cancel{x} + \cancel{y} + \cancel{z}} = \frac{121,94}{121,94 + 67,19 + 5,05} = \frac{121,94}{194,18}$$

$$x = 0,628$$

$$y = \frac{\cancel{x} + \cancel{y} + \cancel{z}}{X + Y + Z} = \frac{67,19}{194,18} = 0,346 \quad Y = 67,19$$

což jsou souřadnice červené barvy uvedené v tab.2.3.1.

Příklad

Určete souřadnice barvy v RGB systému, jsou-li CIE chromatické souřadnice dány vektorem $[0,25, 0,2, 10,0]^T$, a jsou-li chromatické souřadnice pro bílou barvu např. rovny $[0,313, 0,329, 1,0]^T$.

Po dosazení pak dostáváme:

$$D = 0,215$$

$$D \cdot T_r / (Y_w / Y_w) = 0,0539$$

Pak platí

$$T_r = 0,0539 / D \left(Y_w / Y_w \right) = 0,762$$

a analogicky:

$$T_g = 1,114$$

$$T_b = 1,164$$

Výpočtem trichromatických členitelů XYZ z chromatických souřadnic CIE dostáváme:

$$X = x \frac{Y}{y} = 0,25 \frac{10}{0,2} = 12,5$$

$$Z = (1 - x - y) \frac{Y}{y}$$

$$= (1 - 0,25 - 0,2) \frac{10}{0,2} = 27,5$$

Transformace do RGB systému je dána vztahem

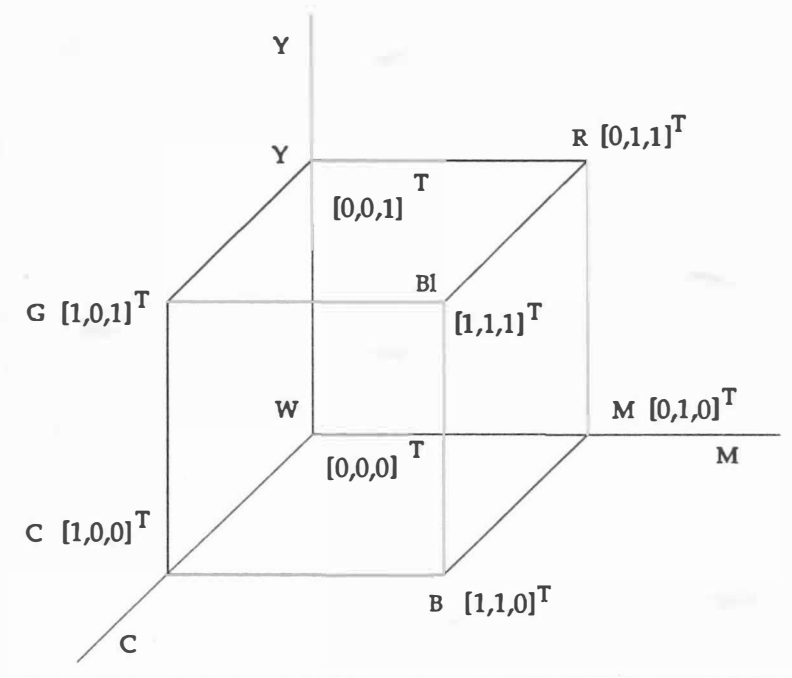
$$\mathbf{r} = \mathbf{T}^{-1} \cdot \mathbf{x}$$

tj.

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 2,739 & -1,143 & -0,424 \\ -1,119 & 2,029 & 0,033 \\ 0,138 & -0,333 & 1,105 \end{bmatrix} \cdot \begin{bmatrix} 12,5 \\ 10,0 \\ 27,5 \end{bmatrix} = \begin{bmatrix} 11,133 \\ 7,209 \\ 28,772 \end{bmatrix}$$

System CMY

Vedle aditivního barevného systému RGB existuje též subtraktivní barevný systém CMY, který je velmi důležitý zejména v případě, že budeme pracovat s nějakým typem zařízení používajícího barevných přetisků. Barevný prostor si lze opět představit ve tvaru krychle se souřadnými osami CMY, viz obr. 2.3.14.



Obr. 2.3.14

Převod mezi CMY a RGB systémy je dán vztahem:

$$\begin{bmatrix} c \\ m \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

kde vektor $[1, 1, 1]^T$ reprezentuje v systému RGB barvu bílou. Opačný převod je definován vztahem:

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} c \\ m \\ y \end{bmatrix}$$

kde vektor $[1, 1, 1]^T$ reprezentuje v systému CMY barvu černou. Poznamenejme, že význam a hodnoty y v CIE, YIQ a CMY systémech jsou rozdílné.

V případě systémů RGB a CMY lze využít technik pultónování tak, že i při použití levných zařízení lze obdržet na výstupu 1024 i více barevných odstínů.

Například vzor 4 x 4 může být definován maticí pro

- černou barvu

$$B = \begin{bmatrix} 1 & 9 & 3 & 15 \\ 13 & 5 & 14 & 7 \\ 4 & 10 & 2 & 12 \\ 16 & 8 & 11 & 6 \end{bmatrix}$$

- purpurovou barvu $M = \hat{B}$

- žlutou barvu $Y = \hat{M}$

- modrozelenou barvu $C = \hat{Y}$,

kde: $\hat{}$ značí "otočení" matice o 90° vpravo.

I když je možné černé barvy docílit pomocí přetisku všech barev, používá se z důvodů rychlosti, zlepšení kontrastu a snížení ceny barva černá jako speciální složka. Tento systém se pak obvykle označuje jako CMYK. Přesnou definici převodních vztahů do RGB, resp. CIE systému lze nalézt v [217].

System YIQ

Pro televizní vysílání je používán systém YIQ zavedený NTSC (National Television Standart Committee) v r. 1953. Tento systém byl použit zejména z důvodu kompatibility barevných a černobílých televizních přijímačů. Hodnota složky Y obsahuje informaci o sytosti a barevném odstínu a je vlastně přijímána černobílým TV přijímačem. Složka I obsahuje informaci o odstínu barvy v rozmezí oranžová-modrozelená, zatímco složka Q obsahuje informaci o odstínu barev zelená-purpurová. Vzhledem k citlivosti je šířka frekvenčního pásma pro I rovna 1.5 MHz, pro Q rovna 0.6 MHz, zatímco pro Y rovna 4 MHz.

Převod systému RGB a YIQ je definován vztahy

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ 0,596 & -0,274 & -0,322 \\ 0,211 & -0,522 & 0,311 \end{bmatrix} \cdot \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

a

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 1,000 & 0,956 & 0,623 \\ 1,000 & -0,272 & -0,648 \\ 1,000 & -1,105 & 1,705 \end{bmatrix} \cdot \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

Uvedené převodní vztahy definují i převod barevných odstínů pro případ výstupu na monochromní display, na kterém lze zobrazit jen různé úrovně šedi, neboť jas je určen složkou Y v systému YIQ, tj.

$$y = [0,299 , 0,587 , 0,114] \cdot [r , g , b]^T$$

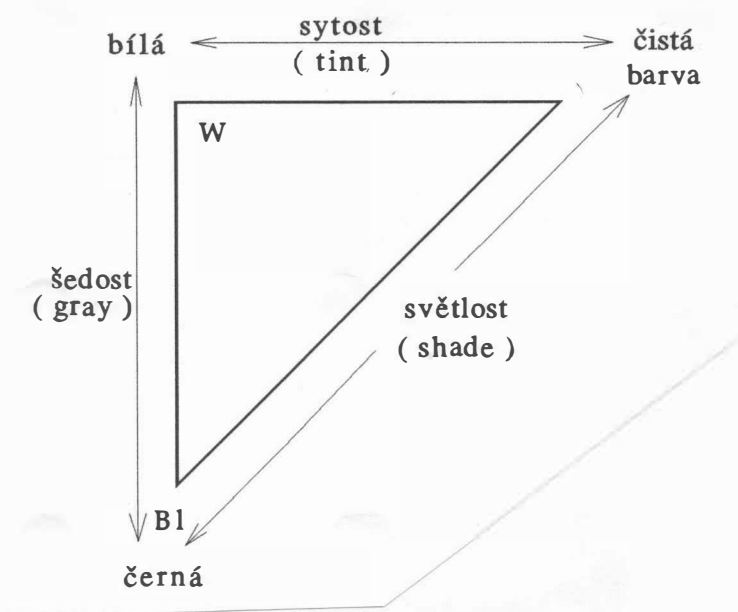
Vzhledem k tomu, že počet úrovní šedi je u zařízení omezen, je nutné v praktických aplikacích řešit problém výběru vhodných subintervalů pro sdružování jasů obsažených v obrázku do skupin tak, aby byl výsledný obrázek co nejvěrnější předloze.

Systemy HLS a HSV

Nevýhodou dosud uvedených barevných systémů je, že jsou víceméně orientované na vlastnosti zařízení. Je jen velmi obtížné si představit, jak běžný uživatel vyjádří hnědou barvu pomocí jednotlivých složek v systému RGB. Z tohoto důvodu byly odvozeny i jiné barevné systémy, které jsou orientovány uživatelsky a které mají za základ opět tři složky, a to (viz obr.2.3.15):

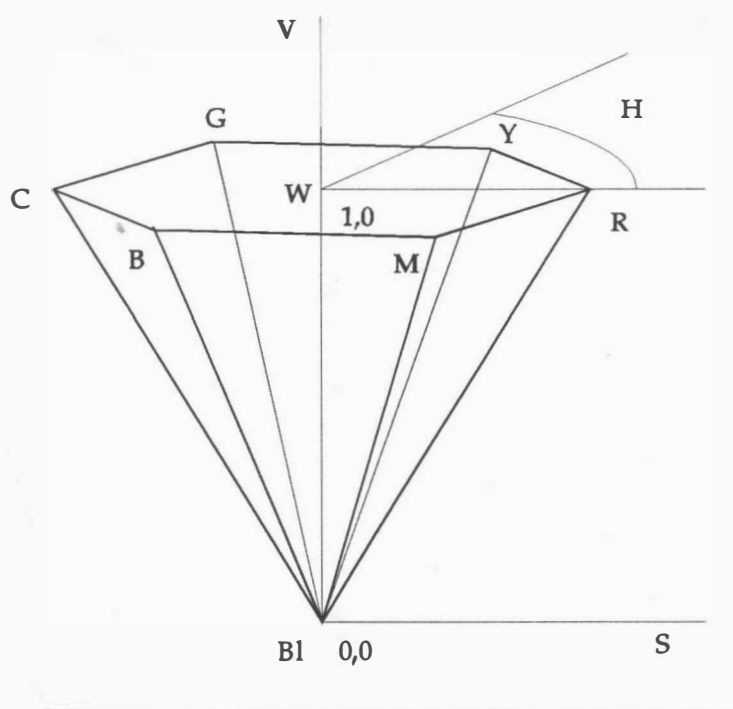
- barevný tón (Hue - označuje se písmenem H)
- sytost barvy (Saturation - označuje se písmenem S)
- jas (někdy označováno jako světlost/tmavost; Value , resp. Lightness - označuje se písmenem V, resp.L).

Odtud jsou pak odvozeny názvy základních systémů, HSV resp. HLS. Na obr.2.3.16 je ukázán systém HSV, zatímco systém HLS je znázorněn na obr.2.3.18.



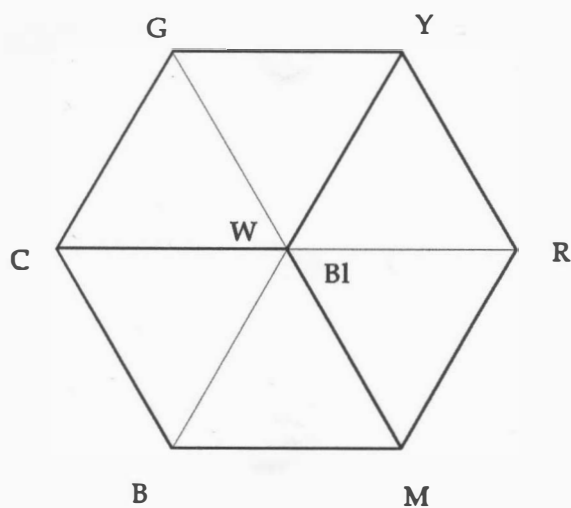
Obr. 2.3.15

Nevýhodou systémů HLS a HSV je, že odstín barvy v těchto systémech není lineárně závislý na hodnotě barevného odstínu v systému RGB a ani jeho změna není spojitá. Např. hrana R-G v krychli RGB se zobrazí jako oblouk R-G v systémech HSV a HLS.



Obr. 2. 3. 16

System HSV (Hue, Saturation, Value) je založen na představě, že hexagon barev vzniká vlastně promítnutím základních barev na rovinu kolmou k tělesové úhlopříčce spojující bílou a černou barvu v RGB krychli, viz obr.2.3.17.



Obr. 2. 3. 17

Převod z RGB do HSV systému a naopak není též realizovatelný jednoduchou transformací. Převod z RGB do HSV systému je popsán algoritmem alg.2.3.1 a převod opačný je popsán algoritmem alg.2.3.2.

```
procedure zRGBdoHSV ( r, g, b: real; var h, s, v: real );
{ vstup: r, g, b ∈ < 0 , 1 > }
{ výstup: h ∈ < 0 , 360 ) ; s , v ∈ < 0 , 1 > }
var q, rc, gc, bc, max, min: real;
begin
  max := MAXIMUM ( r , g , b );
  min := MINIMUM ( r , g , b );
  v := max; { hodnota v }
  if max <> 0 then s := ( max - min ) / max
    else s := 0;
  if s = 0 then h := nedefinováno
  else
  begin q := 1.0 / ( max - min );
    rc := ( max - r ) * q; { vzdálenost od červené }
    gc := ( max - g ) * q; { vzdálenost od zelené }
    bc := ( max - b ) * q; { vzdálenost od modré }
    if r = max then h := bc - gc
      { barva mezi žlutou a purpurovou }
    else if g = max then h := 2 + rc - bc
      { barva mezi modrozelenou a žlutou }
    else if b = max then h := 4 + gc - rc;
      { barva mezi modrozelenou a purpurovou }
    h := h * 60; { konverze do stupňů }
    if h < 0.0 then h := h + 360 { konverze do stupňů }
  end { chromatický případ }
end { zRGBdoHSV };
```

Algoritmus 2.3.1

```

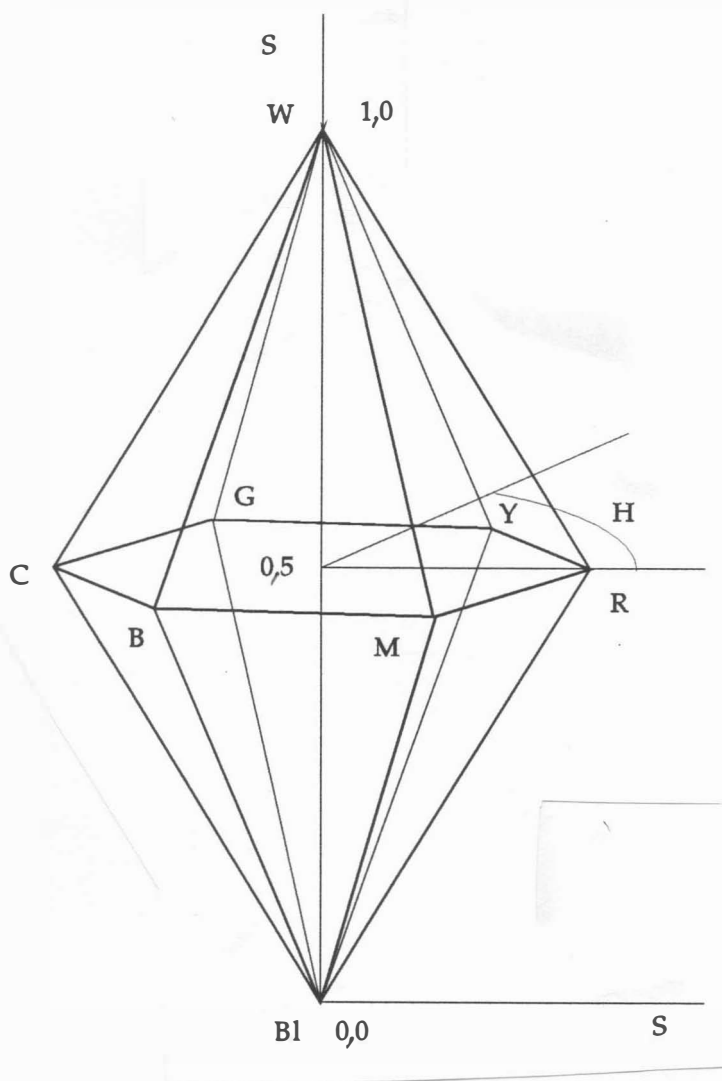
procedure zHSVdoRGB ( var r, g, b: real; h, s, v: real);
{ vstup:  h ∈ < 0 , 360); s, v ∈ < 0 , 1 > }
{ výstup: r, g, b ∈ < 0 , 1 > }
var f, p, q, t: real;
begin
  if s = 0 then { achromatický případ }
    if h = nedefinováno then
      begin r := v;
          g := v;
          b := v
      end
    else ERROR
  else
    begin { chromatický případ }
      if h = 360 then h := 0;
      h := H / 60; { h ∈ < 0 , 6 ) }
      i := TRUNC( h ); { celá část }
      f := h - i; { zbytek }
      p := v * ( 1 - s );
      q := v * ( 1 - s * f );
      t := v * ( 1 - s * ( 1 - f ) );
      case i of
        0:      (r,g,b) := ( v , t , p );
        1:      (r,g,b) := ( q , v , p );
        2:      (r,g,b) := ( p , v , t );
        3:      (r,g,b) := ( p , q , v );
        4:      (r,g,b) := ( t , p , v );
        5:      (r,g,b) := ( v , p , q )
      end { case }
    end { barevný odstín }
  end { zHSVdoRGB };

```

Algoritmus 2.3.2

Rozšířením HSV systému je systém HLS (Hue, Lightness, Saturation). Na obr.2.3.18 je transformovaná RGB krychle do HLS systému. Způsob definování barvy v HLS systému je obdobný způsobu v HSV systému s tím rozdílem, že barevný odstín není definován pro $S = 0$ a $S = 1$, což odráží více realitu, neboť při

mezních hodnotách saturace není barevný rozdíl dobře rozlišitelný.



Obr. 2.3.18

Převod ze systému RGB do systému HLS a naopak je definován algoritmy alg.2.3.3 a alg.2.3.4.

```

procedure zRGBdoHLS ( r, g, b: real; var h, l, s: real );
{ vstup: r, g, b ∈ < 0 , 1 > }
{ výstup: h ∈ < 0 , 360 > ; l, s ∈ < 0,1 > }
var max, min, rc, gc, bc, q: real;

```



```

begin
  max := MAXIMUM ( r, g, b );
  min := MINIMUM ( r, g, b );
  l := ( max + min ) / 2; { světlost }
  { výpočet saturace }
  if max = min then { achromatický případ }
  begin s := 0; h := nedefinováno end
  else
  begin { chromatický případ }
    if l <= 0.5 then S := ( max - min ) / ( max + min )
    else S := ( max - min ) / ( 2 - max + min );
    { výpočet odstínu }
    q := 1 / ( max - min );
    rc := ( max - r ) * q;
    gc := ( max - g ) * q;
    bc := ( max - b ) * q;
    if r = max then h := bc - gc
      { výsledná barva mezi žlutou a purpurovou }
    else if g = max then h := 2 + rc - bc
      { výsledná barva mezi modrozelenou a žlutou }
    else if b = max then h := 4 + gc - rc;
      { výsledná barva mezi purpurovou }
      { a modrozelenou }
    h := h * 60;
    if h < 0.0 then h := h + 360
  end { chromatický případ }
end { zRGBdoHLS };

```

Algoritmus 2.3.3

```

procedure zHLSdoRGB ( var r, g, b: real; h, l, s: real);
{ vstup: h ∈ < 0 , 360 ); l, s ∈ < 0 , 1 > }
{ výstup: r, g, b ∈ < 0 , 1 > }
var M1, M2: real;
function VAL ( n1, n2, h: real): real;
begin
  if h > 360.0 then h := h - 360.0;
  if h < 0.0 then h := h + 360.0;

```

```

if h < 60.0 then VAL := n1 + ( n2 - n1 ) * h / 60.0
else if h < 180 then VAL := n2
      else if h < 240
            then VAL := n1 + ( n2 - n1 ) * ( 240 - h ) / 60.0
            else VAL := n1
end { VAL };
begin
  if l = 0.5 then M2 := l * ( 1 + s )
                else M2 := l + s - l * s;
  M1 := 2 * l - M2;
  if s = 0 then
    begin
      if h = nedefinováno then
        begin r := l; g := l; b := l end
      else ERROR
    end
  else
    begin
      r := VAL ( M1, M2, h + 120 );
      g := VAL ( M1, M2, h );
      b := VAL ( M1, M2, h - 120 )
    end
  end
end { zHLSdoRGB };

```

Algoritmus 2.3.4

System HSI

Mnohé grafické systémy (Tektronics) využívají systému HSI (barevný tón-hue, sytost-saturation, jas-intensity), který bývá zaměňován za systém HLS. Konverze mezi HSI a RGB systémy může být provedena ve dvou krocích, a to zavedením pomocného vektoru $[M_1, M_2, I_1]^T$ a transformací:

$$\begin{bmatrix} M_1 \\ M_2 \\ I_1 \end{bmatrix} = \frac{1}{\sqrt{6}} \begin{bmatrix} 2 & -1 & -1 \\ 0 & \sqrt{3} & -\sqrt{3} \\ \sqrt{2} & \sqrt{2} & \sqrt{2} \end{bmatrix} \cdot \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

pak:

$$H = \arctg (M_1/M_2) \quad s = \frac{\sqrt{3}}{\sqrt{2}} \sqrt{M_1^2 + M_2^2} \quad I = \frac{\sqrt{3}}{3} I_1$$

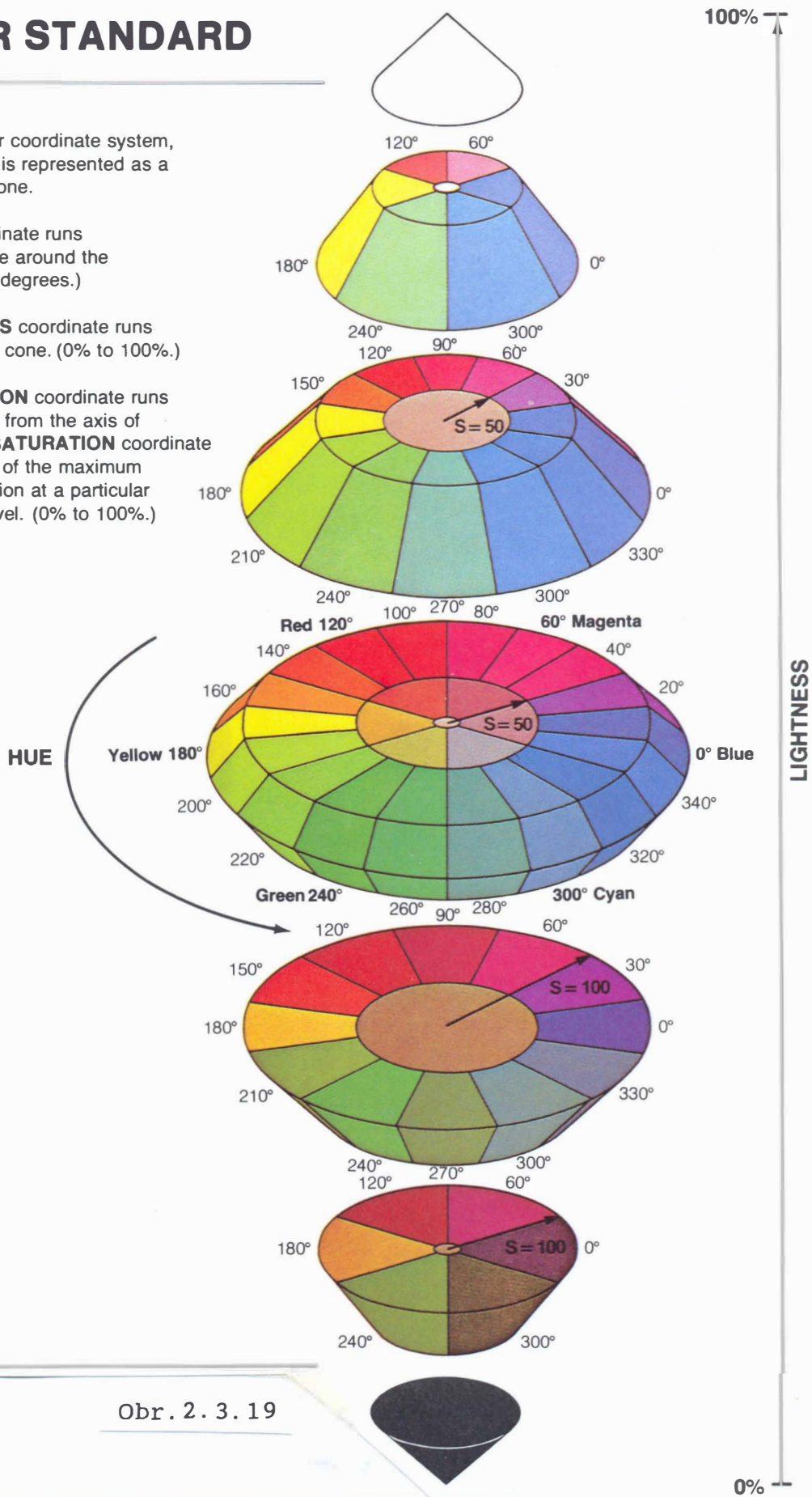
COLOR STANDARD

In the **HLS** color coordinate system, the color space is represented as a double-ended cone.

The **HUE** coordinate runs counterclockwise around the cone. (0 to 360 degrees.)

The **LIGHTNESS** coordinate runs vertically up the cone. (0% to 100%.)

The **SATURATION** coordinate runs radially outward from the axis of the cone. The **SATURATION** coordinate is a percentage of the maximum possible saturation at a particular **LIGHTNESS** level. (0% to 100%.)



Obr. 2.3.19

Pro zpětnou transformaci pak platí, že inverzní matice je rovna matici transponované, přičemž:

$$M_1 = \frac{\sqrt{2}}{\sqrt{3}} S \sin(H)$$

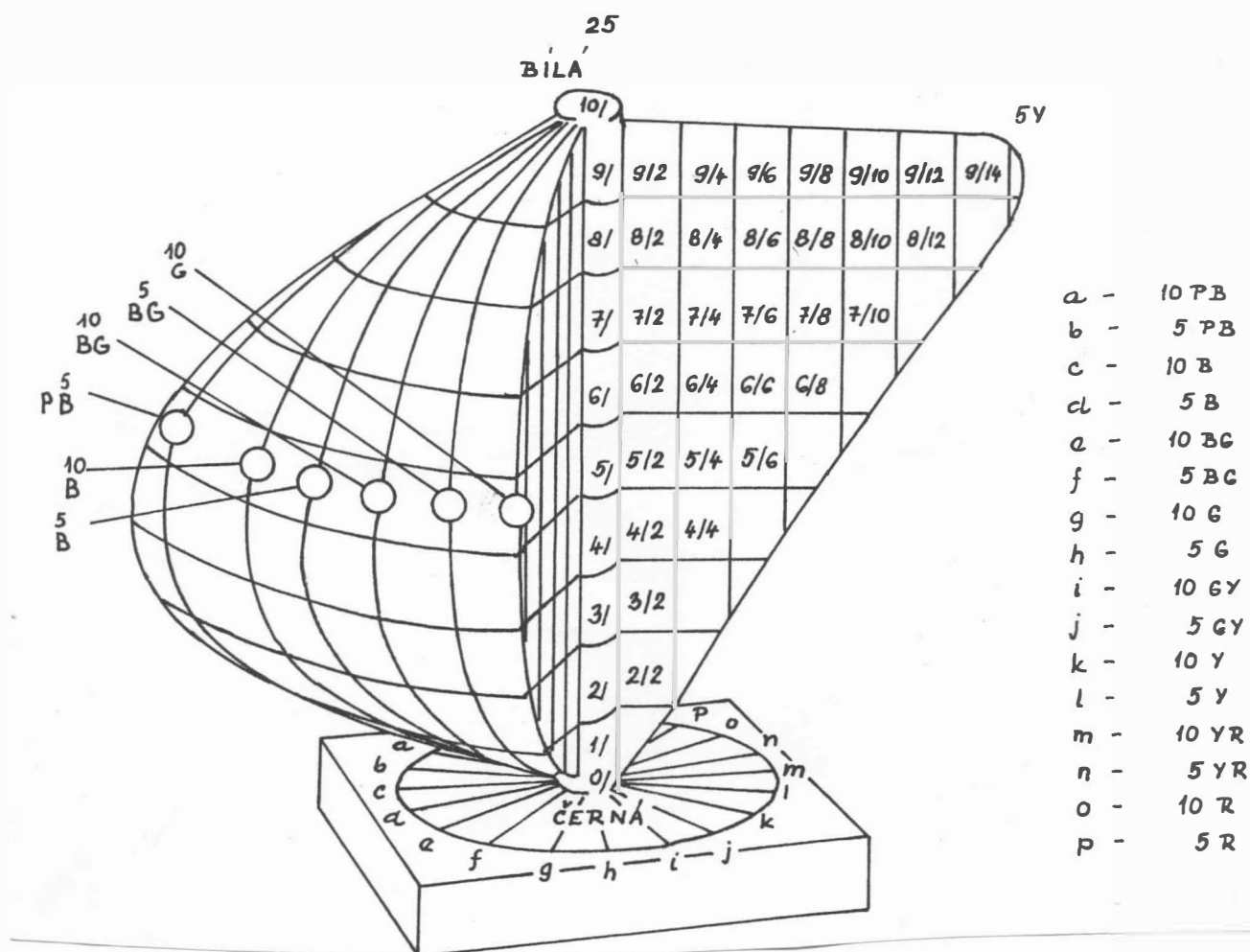
$$M_2 = \frac{\sqrt{2}}{3} S \cos(H)$$

$$I_1 = I\sqrt{3}$$

K systému HSI lze vytvořit i odpovídající model, který je podobný systému HSV a který je znázorněn na obr. 2.3.19.

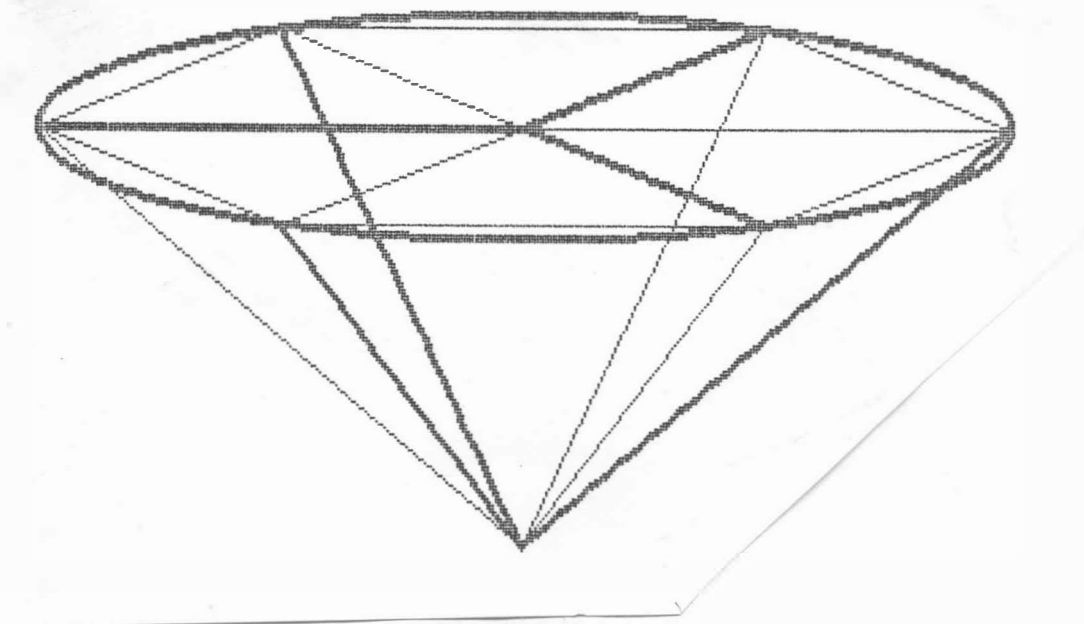
Systémy vzorníků, tabulek a barevné atlasy

Barevné vzorníky, tabulky a atlasy se používají zejména v textilním, chemickém průmyslu a všude tam, kde je zapotřebí rychlá a ne příliš přesná kontrola barev. Principem je rozřídění barev do skupin, přičemž každé barvě je přiřazen jednoznačný kód.

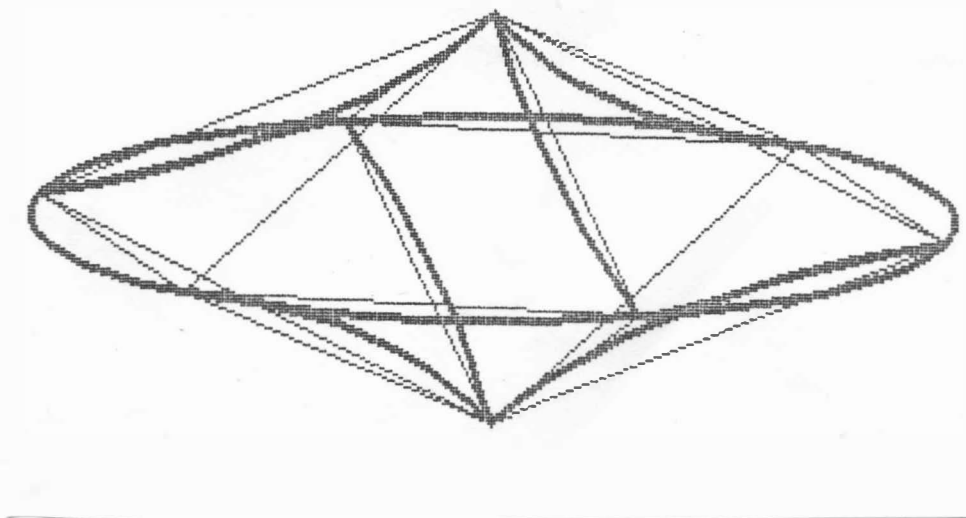


Obr. 2.3.20

Jedním takovým barevným systémem je Munsellův systém, který bere v úvahu i citlivost oka k jednotlivým barvám. Tento systém je zobrazen na obr.2.3.20. Každému políčku v diagramu odpovídá určitá barva, která je v katalogu barev reprezentována barevným lístkem, přičemž číslo barvy, např. 5Y 7/7, jednoznačně barvu identifikuje. Tento systém je důležitý zejména v oblasti průmyslu keramického, textilního, tiskařského apod. Kromě Munsellova systému se ještě používá systém Ostwaldův nebo Hickethierův.



Obr.2.3.21



Obr.2.3.22

Je vhodné poznamenat, že lineární interpolaci barev v RGB systému, resp. v CMY, YIQ, neodpovídá lineární interpolace barev v HLS, resp. HSV systému, neboť obecně nejde o afinní transformaci. Na obr. 2.3.21 je ukázáno, jak se transformuje RGB krychle do systému HSV, na obr. 2.3.22 pak transformace do HLS.

Kromě výše uvedených barevných systémů, které jsou používány nejvíce v počítačové grafice, je celá řada dalších systémů používaných zejména v kolorimetrii. S rozvojem výpočetní techniky a periferních zařízení nabývají tyto systémy na důležitosti zejména v oblastech, kde jde o věrnost podání barev, viz [193], [201], [204], [210], [217].

System CIE LAB

Jedním ze systémů, který se prosazuje a který je využíván celou řadou profesionálních kolorimetrů, je systém CIE LAB, viz [201]. Systém udává jas barvy L^* , odstín barev mezi červenou a zelenou a^* a odstín barev mezi modrou a žlutou b^* . Převodní vztahy mezi systémem XYZ a CIE LAB jsou definovány takto [216]:

$$\bar{Y} = \frac{L^* + 16}{116}$$

$$X = X_w \left(\frac{a^*}{500} + \bar{Y} \right)^3$$

$$Y = Y_w \bar{Y}^3$$

$$Z = Z_w \left(-\frac{b^*}{200} + \bar{Y} \right)^3$$

$$L^* \in \langle 1, 100 \rangle$$

kde X_w , Y_w , Z_w jsou souřadnice smluvního bílého světla v systému XYZ.

Pro opačný převod platí vztahy [218]:

$$L^* = q \left(\frac{Y}{Y_w} \right)^{1/3} - 16$$

$$a^* = 500 \left[\left(\frac{X}{X_w} \right)^{1/3} - \left(\frac{Y}{Y_w} \right)^{1/3} \right]$$

$$b^* = 200 \left[\left(\frac{Y}{Y_w} \right)^{1/3} - \left(\frac{Z}{Z_w} \right)^{1/3} \right]$$

přičemž

$$q = \begin{cases} 116 & \text{pro } \frac{Y}{Y_w} > 0,008856 \\ 903,29 & \text{pro } \frac{Y}{Y_w} \leq 0,008856 \end{cases}$$

Někteří autoři [221] uvádějí vztahy, kde hodnota q je určena:

$$q = 25 \quad \text{pro } 1 \leq Y \leq 100$$

Podobným systémem je systém CIE $L^*u^*v^*$.

Systém CIE $L^*u^*v^*$

Systém CIE $L^*u^*v^*$ [217] je výhodný zejména z důvodu možnosti přesné definice rozdílnosti barev. Systém CIE $L^*u^*v^*$ je definován převodními vztahy

$$L^* = q \left(\frac{Y}{Y_w} \right)^{1/3} - 16$$

$$q = \begin{cases} 116 & \text{pro } \frac{Y}{Y_w} > 0,008856 \\ 903,29 & \text{pro } \frac{Y}{Y_w} \leq 0,008856 \end{cases}$$

$$u^* = 13 L^* (u' - u'_w)$$

$$v^* = 13 L^* (v' - v'_w)$$

kde u'_w , v'_w jsou hodnoty u' , v' pro smluvní bílé světlo, přičemž

$$u' = \frac{4 X}{X + 15 Y + 3 Z} \quad v' = \frac{9 Y}{X + 15 Y + 3 Z}$$

V [221] je hodnota q definována:

$$q = 25 \quad \text{pro } 1 \leq Y \leq 100$$

Pak vzdálenost ΔE^*_{uv} barvy $L^*_1 u^*_1 v^*_1$ a barvy $L^*_2 u^*_2 v^*_2$ je definována [217] takto

$$\Delta E^*_{uv} = \sqrt{(\Delta L^*)^2 + (\Delta u^*)^2 + (\Delta v^*)^2}$$

$$\text{kde: } \Delta L^* = L^*_2 - L^*_1 \quad \Delta u^* = u^*_2 - u^*_1 \quad \Delta v^* = v^*_2 - v^*_1$$

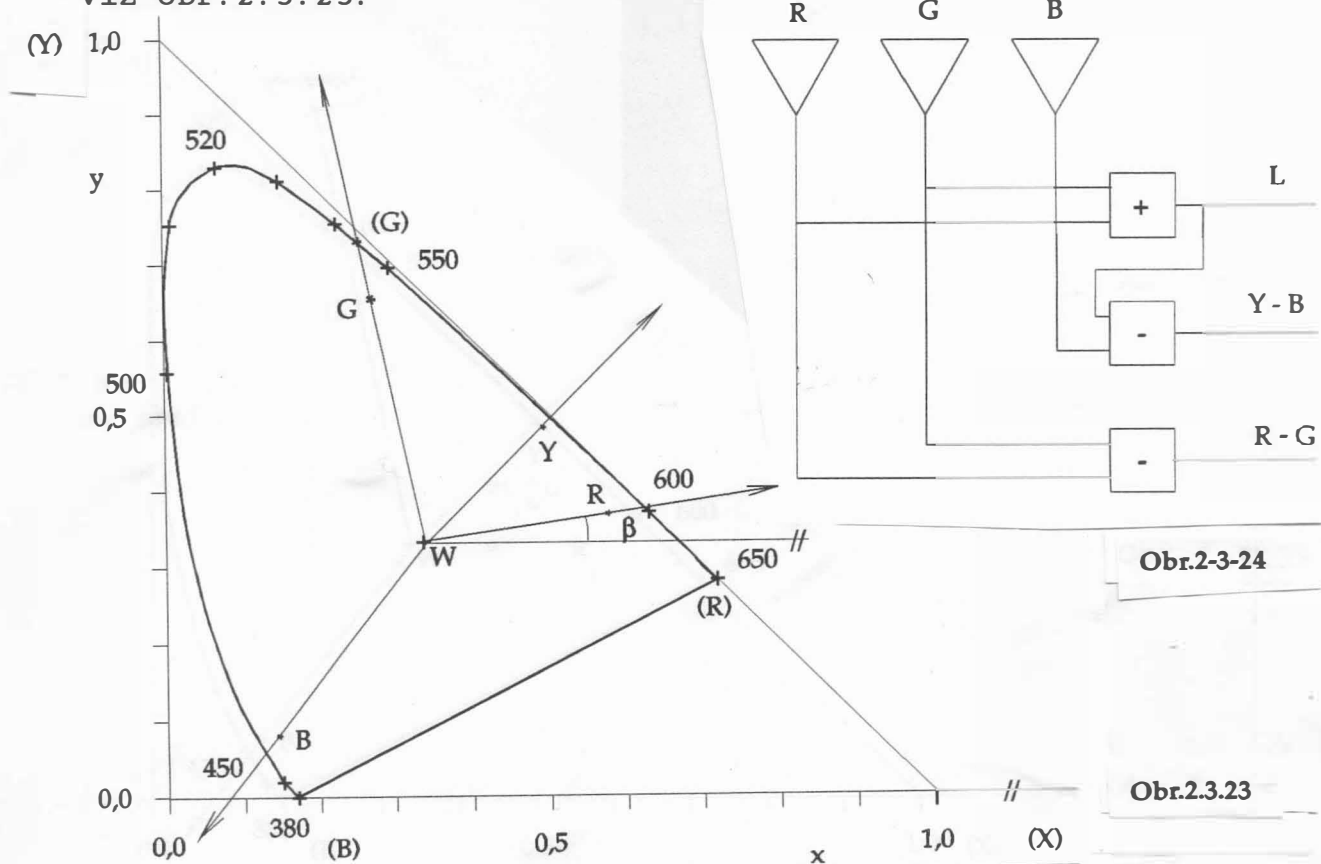
Kromě výše uvedených systémů jsou používány též systémy respektující způsob vnímání barev. Podle teorie jsou na sítnici oka tři druhy čípků, které produkují tři různé signály na základě světelného podnětu [204], a to:

- signál jasový jako souhrnný vjem v oblasti červené a zelené,
- signál pro odlišení barev v oblasti červená - zelená
- signál pro odlišení barev v oblasti žlutá - modrá.

Systémy založené na uvedeném principu vysvětlují celou řadu jevů v oblasti vnímání barev.

System Opponent

Jedním z nejjednodušších systémů je systém Opponent [216], který předpokládá, že barvy jsou určeny polohou v diagramu, viz obr.2.3.23.



Obr.2-3-24

Obr.2.3.23

Obr.2.3.24 pak znázorňuje, jak jsou jednotlivé složky definovány při převodu ze systému CIE-xy, resp. RGB.

Je-li barva c v systému Opponent určena polohou v diagramu RYGB a jasem L, pak je poloha barvy c v systému CIE-xy určena úhlem β , vzdáleností od bodu reprezentujícího barvu bílou a jasem, viz [216]. Úhel β je definován vztahy:

p	q	β
≥ 0	≥ 0	$\vartheta_Y + (\vartheta_G - \vartheta_Y) \frac{2}{\pi} \arctg\left(\frac{q}{p}\right)$
< 0	≥ 0	$\vartheta_G + (\vartheta_B - \vartheta_G) \frac{2}{\pi} \arctg\left(-\frac{p}{q}\right)$
< 0	< 0	$\vartheta_B + (\vartheta_R - \vartheta_B) \frac{2}{\pi} \arctg\left(\frac{q}{p}\right)$
≥ 0	< 0	$\vartheta_R + (\vartheta_Y - \vartheta_R) \frac{2}{\pi} \arctg\left(-\frac{p}{q}\right)$

kde úhel ϑ_Y je úhel sevřený osou x a spojnici bodu x_w , který reprezentuje bílou barvu, a bodu Y, který reprezentuje žlutou barvu (Yellow); analogicky pro ϑ_G , ϑ_R , ϑ_B

Vzdálenost od bodu x_w je pak určena vztahem:

$$R = \sqrt{p^2 + q^2}$$

Pak souřadnice barvy c v systému CIE-xy jsou přibližně určeny takto:

$$x = x_w + R \cdot \cos \beta \quad y = x_w + R \cdot \sin \beta \quad z = 1 - x - y$$

V souřadném systému XYZ má barva c souřadnice:

$$X = x \frac{L}{Y} \quad Y = L \quad Z = z \frac{L}{Y}$$

Tento přepočít je sice poměrně hrubý, ale je kvalitativně správnou aproximací fyziologického vizuálního systému.

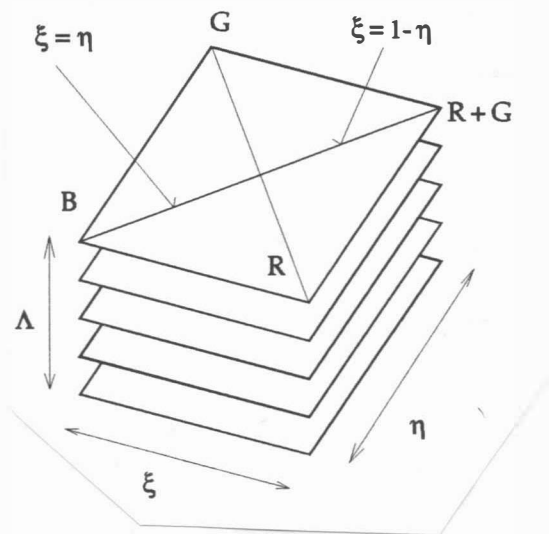
Pro případné srovnávání je vhodné přesně definovat referenční body systému Opponent, viz [216]:

	x	y	z
R	0,62	0,33	0,05
G	0,21	0,68	0,11
B	0,15	0,06	0,79
Y	0,51	0,49	0,00
W	0,31	0,32	0,37

I když tento systém vysvětluje mnohé jevy, je však výpočetně poměrně náročný [204], přičemž gamut barev pro display je nelineární a systém neposkytuje jednoduchá pravidla pro mísení barev. Z tohoto důvodu byl zaveden systém RGYB.

System RGYB

System RGYB [204] byl zaveden pro odstranění některých vad systému Opponent. Jeho princip je znázorněn na obr.2.3.25.



Obr.2.3.25

Z obrázku je zřejmé, že achromatické barvy, tj. bílá, odstíny šedé a černá, jsou určeny bodem $\xi = 0,5$, $\eta = 0,5$, přičemž Λ určuje úroveň jasu, tj. šed'

Pro systém RGYB platí následující převodní vztahy pro převod do systému RGB

$$R = \xi \cdot \Lambda \quad G = \eta \cdot \Lambda \quad B = (1 - \max(\xi, \eta)) \cdot \Lambda$$

kde:

$$\xi \in \langle 0, 1 \rangle \quad \eta \in \langle 0, 1 \rangle \quad \Lambda \in \langle 0, 1 \rangle$$

Z hlediska výpočetní náročnosti jsou požadavky zanedbatelné. Uvedený systém vykazuje navíc i jiné dobré vlastnosti, např. z hlediska mísení barev, kdy pravidla mají lineární charakter. Pro $\Lambda \in \langle 1, 2 \rangle$ systém respektuje i vliv saturace, viz [204].

Systém RGYB lze též modifikovat tak, že

$$R = 2 \cdot \xi - \Lambda \cdot \xi + \Lambda - 1$$

$$G = 2 \cdot \eta - \Lambda \cdot \eta + \Lambda - 1$$

$$B = \max(\xi, \eta) \cdot (\Lambda - 2) + 1$$

Pak "horní vrstva", tj. $\Lambda = 2,0$ je celá bílá, zatímco "dolní vrstva", tj. $\Lambda = 0,0$ je celá černá a "prostřední vrstva", tj. $\Lambda = 1,0$ obsahuje úplnou paletu barev.

Až dosud byly předloženy různé barevné systémy a jejich vzájemné převody. Při použití moderních barevných výstupních zařízení je nezbytné zkoumat i otázku zajištění stejného barevného vjemu jak na obrazovce, tak i na výstupu z barevné tiskárny. Uspokojivé řešení tohoto problému je velmi náročné, viz [193], [201], a zcela překračující možnosti dostupné techniky. Nicméně jde o problém, který v krátké budoucnosti jistě nabude na významu vzhledem k velkému pokroku v oblasti grafických barevných výstupů.

Systém AC_1C_2

Systém AC_1C_2 odvodil Meyer, viz [91]; [221], který definoval převodní vztahy takto:

$$\begin{bmatrix} A \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} -0,0177 & 1,0090 & 0,0073 \\ -1,5370 & 1,0821 & 0,3209 \\ 0,1946 & -0,2045 & 0,5264 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Systém $S\Theta W$

Systém $S\Theta W$ není příliš znám a jde vlastně o vyjádření systému CIE UVW v polárním souřadném systému. Proměnná S popisuje saturaci barvy, Θ barevný tón a W jas barvy, viz CIE-UVW. Pro souřadnice S, Θ , W platí následující převodní vztahy:

$$S = \sqrt{U^2 + V^2} = 13 W \sqrt{(u - u_w)^2 + (v - v_w)^2}$$

$$\Theta = \arctg(v/u) = \arctg[(v - v_w)/(u - u_w)]$$

kde (u_w, v_w) je pozice smluvního bílého světla v diagramu CIE uv.

Mnohá grafická zařízení nemají k dispozici velkou paletu barev, např. laserové nebo inkoustové tiskárny, zatímco např. dnes již standardní Super VGA karta poskytuje alespoň 256 barev současně zobrazitelných. Výstupní rastrová grafická zařízení lze rozdělit z hlediska použití barev takto:

výstupní zařízení	{ s pevnou paletou s volitelnou paletou	inkoustové, laserové tiskárny 3 - 4 barvy v paletě
		{ VGA 16 barev/256 možných Super VGA 256 barev/4K možných speciální 4K barev/10 ⁶ možných

Je zřejmé, že docílení velmi věrného barevného výstupu není víceméně problémem, pokud je použit vhodný video interface a display. Naproti tomu docílit velmi kvalitního výstupu např. na laserové nebo inkoustové tiskárně při použití pouze 3 - 4 barev je velmi obtížné. Obecně lze říci, že je možné použít modifikovaných technik pŕltónování k docílení velmi věrného barevného výstupu, viz obr.1.1.

Tento obrázek dokumentuje, jaké kvality lze docílit při použití systému CMYK (CMY + černá barva). Velmi důležitá je i hustota tisku, která musí být alespoň 300 dpi (bodů na palec).

Další speciální skupinou modifikací dříve uvedených algoritmů je modifikace technik pŕltónování pro případ použití barev, resp. slučování jednotlivých barev palety a vytváření palety nové, pro zařízení mající 16 - 256 barev k docílení vjemu ekvivalentního mnohem větší paletě barev [101]. Tato úloha je vlastně úlohou nalezení takové omezené palety barev a barevných odstínů, že výsledný vjem obrazu zobrazeného s takovou paletou je co "nejbližší" vjemu obrazu, který byl zadán. Tato úloha se řeší několika základními metodami, a to:

- prostor RGB se rozdělí do boxů, které reprezentují jednu barvu. Počet boxů je dán dostupnou paletou barev. Vzhledem k menší citlivosti oka na modrou barvu se obvykle volí počty bitů reprezentující jednotlivé barevné složky takto:

$$R : G : B = 3 : 3 : 2$$

I když je pixel definován pomocí osmi bitů, je kvalita obvykle nízká i při použití ditheringu. Tento algoritmus se v literatuře nazývá **Standard Algorithm**.

- vybere se K nejvíce používaných barev v daném obraze, přičemž K je dáno počtem současně zobrazitelných barev. Má-li obraz hodně rozdílných barev, pak výsledky nejsou uspokojivé. Tento algoritmus se nazývá **Popularity Algorithm**.
- vezme se prvních K barev jako počáteční aproximace. Při přečtení barvy, která není v tabulce, jsou dvě nejbližší barvy z $K + 1$ sloučeny a nahrazeny váženým průměrem. Tento postup se opakuje, pokud není zpracován poslední pixel daného obrazu. Tento algoritmus se v literatuře vyskytuje pod názvem **Octree Algorithm**.
- RGB prostor se rozdělí rovnoměrně do K boxů. V každém kroku se box s největším počtem výskytů rozdělí ve směru nejdelší osy tak, že nové sub-boxy reprezentují přibližně stejný počet výskytů v zadaném obraze. Algoritmus je označován v literatuře jako **Median Cut Algorithm**.

Z výše uvedeného je zřejmé, že jde o problematiku velmi rozsáhlou, přičemž dosažené výsledky jsou nejen velmi závislé na technických prostředcích, ale i na zpracovávaných obrazech. Je vhodné podotknout, že experimenty popsané v literatuře nelze jednoduchým způsobem ověřit vzhledem k nedostupnosti příslušných technických prostředků.

Příklad

V systémech založených na RGB systému se používá k promítnutí jednotková rovina. Odvoďte analogické vztahy pro případ promítání na jednotkovou kouli, zobrazte rozložení barev, a diskutujte dosažené výsledky. Jak vypadá diagram analogický diagramu CIE-xy?

Poznámka

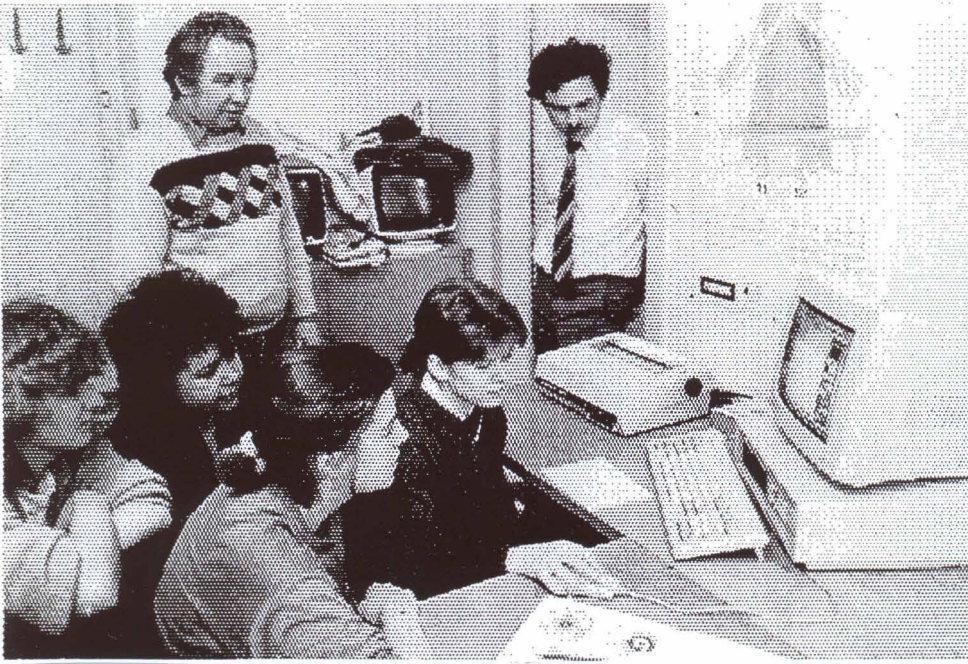
Pro všechny experimenty využijte hodnot uvedených v následující tabulce udávající závislost r , g , b a x , y , z na vlnové délce λ :

λ [nm]	r	g	b	x	y	z
380	0,00003	-0,00001	0,00117	0,1741	0,0050	0,8209
385	0,00005	-0,00002	0,00189	0,1740	0,0050	0,8210
390	0,00010	-0,00004	0,00359	0,1738	0,0049	0,8213
395	0,00017	-0,00007	0,00647	0,1736	0,0049	0,8215
400	0,00030	-0,00014	0,01214	0,1733	0,0048	0,8219
405	0,00047	-0,00022	0,01969	0,1730	0,0048	0,8222
410	0,00084	-0,00014	0,03707	0,1726	0,0048	0,8226
415	0,00139	-0,00070	0,06637	0,1721	0,0048	0,8231
420	0,00211	-0,00110	0,11541	0,1714	0,0051	0,8235
425	0,00266	-0,00143	0,18575	0,1703	0,0058	0,8239
430	0,00218	-0,00119	0,24769	0,1689	0,0069	0,8245
435	0,00036	-0,00021	0,29012	0,1669	0,0086	0,8246
440	-0,00261	0,00149	0,31228	0,1644	0,0109	0,8247
445	-0,00673	0,00379	0,31860	0,1611	0,0138	0,8251
450	-0,01213	0,00678	0,31670	0,1566	0,0177	0,8257
455	-0,01874	0,01046	0,31166	0,1510	0,0227	0,8263
460	-0,02608	0,01485	0,29821	0,1440	0,0297	0,8263
465	-0,03324	0,01977	0,27295	0,1355	0,0399	0,8246
470	-0,03933	0,02538	0,22991	0,1241	0,0578	0,8181
475	-0,04471	0,03183	0,18592	0,1096	0,0868	0,8038
480	-0,04939	0,03914	0,14494	0,0913	0,1327	0,7760
485	-0,05364	0,04713	0,10968	0,0687	0,2007	0,7306
490	-0,05814	0,05689	0,08257	0,0454	0,2950	0,6595
495	-0,06414	0,06948	0,06246	0,0235	0,4127	0,5638
500	-0,07173	0,08536	0,04776	0,0082	0,5384	0,4534
505	-0,08120	0,10593	0,03688	0,0039	0,6548	0,3413
510	-0,08901	0,12860	0,02698	0,0139	0,7502	0,2359
515	-0,09356	0,15262	0,01842	0,0389	0,8120	0,1491
520	-0,09264	0,17468	0,01221	0,0743	0,8338	0,0919
525	-0,08473	0,19113	0,00830	0,1142	0,8262	0,0596
530	-0,07101	0,20317	0,00549	0,1547	0,8059	0,0394
535	-0,05316	0,21083	0,00320	0,1929	0,7816	0,0255
540	-0,03152	0,21466	0,00146	0,2296	0,7543	0,0161
545	-0,00613	0,21487	0,00023	0,2658	0,7243	0,0099
550	0,02279	0,21178	-0,00058	0,3016	0,6923	0,0061
555	0,05514	0,20588	-0,00105	0,3373	0,6589	0,0038
560	0,09060	0,19702	-0,00130	0,3731	0,6245	0,0024
565	0,12840	0,18522	-0,00138	0,4087	0,5896	0,0017
570	0,16768	0,17087	-0,00135	0,4441	0,5547	0,0012
575	0,20715	0,15429	-0,00123	0,4788	0,5202	0,0010
580	0,24526	0,13610	-0,00108	0,5125	0,4866	0,0009
585	0,27989	0,11686	-0,00093	0,5448	0,4544	0,0008
590	0,30928	0,09754	-0,00079	0,5752	0,4242	0,0006
595	0,33184	0,07909	-0,00063	0,6029	0,3965	0,0006
600	0,34429	0,06246	-0,00049	0,6270	0,3725	0,0005
605	0,34756	0,04776	-0,00038	0,6482	0,3514	0,0004
610	0,33971	0,03557	-0,00030	0,6658	0,3340	0,0002
615	0,32265	0,02583	-0,00022	0,6801	0,3198	0,0002
620	0,29708	0,01828	-0,00015	0,6915	0,3083	0,0002
625	0,26348	0,01253	-0,00011	0,7006	0,2993	0,0001
630	0,22677	0,00833	-0,00008	0,7079	0,2920	0,0001
635	0,19233	0,00537	-0,00005	0,7140	0,2859	0,0000
640	0,15968	0,00334	-0,00003	0,7190	0,2809	0,0000

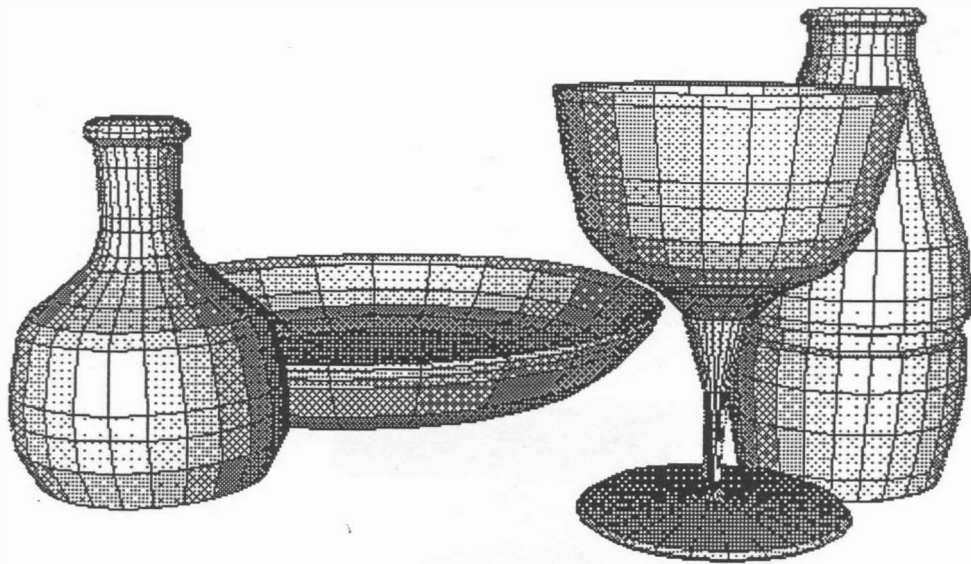
Pokračování závislosti r , g , b a x , y , z na vlnové délce:

λ [nm]	r	g	b	x	y	z
645	0,12905	0,00199	-0,00002	0,7230	0,2770	0,0000
650	0,10167	0,00116	-0,00001	0,7260	0,2740	0,0000
655	0,07857	0,00066	-0,00001	0,7283	0,2717	0,0000
660	0,05932	0,00037	0,00000	0,7300	0,2700	0,0000
665	0,04366	0,00021	0,00000	0,7311	0,2689	0,0000
670	0,03149	0,00011	0,00000	0,7320	0,2680	0,0000
675	0,02294	0,00006	0,00000	0,7327	0,2673	0,0000
680	0,01687	0,00003	0,00000	0,7334	0,2666	0,0000
685	0,01187	0,00001	0,00000	0,7340	0,2660	0,0000
690	0,00819	0,00000	0,00000	0,7344	0,2656	0,0000
695	0,00572	0,00000	0,00000	0,7346	0,2654	0,0000
700	0,00410	0,00000	0,00000	0,7347	0,2653	0,0000
705	0,00291	0,00000	0,00000	0,7347	0,2653	0,0000
710	0,00210	0,00000	0,00000	0,7347	0,2653	0,0000
715	0,00148	0,00000	0,00000	0,7347	0,2653	0,0000
720	0,00105	0,00000	0,00000	0,7347	0,2653	0,0000
725	0,00074	0,00000	0,00000	0,7347	0,2653	0,0000
730	0,00052	0,00000	0,00000	0,7347	0,2653	0,0000
735	0,00036	0,00000	0,00000	0,7347	0,2653	0,0000
740	0,00025	0,00000	0,00000	0,7347	0,2653	0,0000
745	0,00017	0,00000	0,00000	0,7347	0,2653	0,0000
750	0,00012	0,00000	0,00000	0,7347	0,2653	0,0000
755	0,00008	0,00000	0,00000	0,7347	0,2653	0,0000
760	0,00006	0,00000	0,00000	0,7347	0,2653	0,0000
765	0,00004	0,00000	0,00000	0,7347	0,2653	0,0000
770	0,00003	0,00000	0,00000	0,7347	0,2653	0,0000
775	0,00001	0,00000	0,00000	0,7347	0,2653	0,0000
780	0,00000	0,00000	0,00000	0,7347	0,2653	0,0000

Na obr.2.3.26 je ukázka použití techniky pultónování při použití mozaikové tiskárny (9 jehliček). Obr.2.3.27 ukazuje možnost použití techniky vzorů pro zobrazování scén s více úrovněmi šedi při použití laserové tiskárny. Z obr.3.2.28, kde je zobrazena krychle RGB v systému HSI, je zřejmá nelinearita převodních vztahů mezi RGB a HSI systémy.

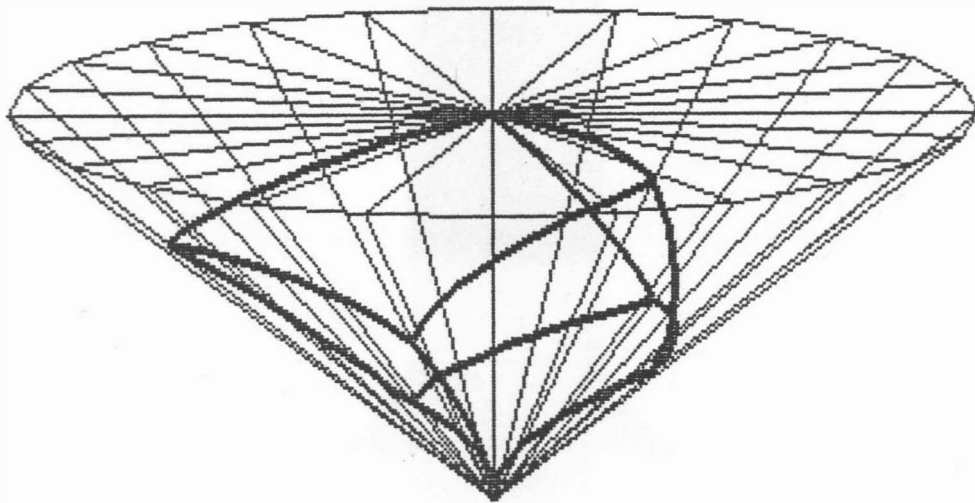


Obr. 2.3.26



Obr. 2.3.27

H S I SYSTEM



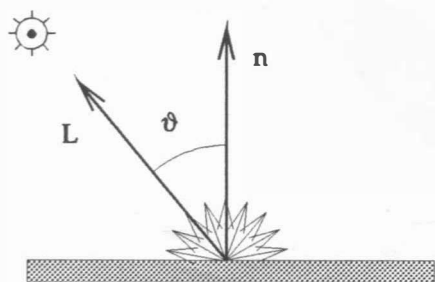
Obr. 2.3.28

3. Osvětlení a stínování

Po provedení geometrických transformací (posuv, rotace apod.) s objekty tvořícími zadanou scénu, následuje poměrně složitá a dlouhotrvající operace odstranění neviditelných hran, resp. povrchů (hidden-line, resp. hidden surface elimination). Řešení problému stínování povrchů (rendering), což je v podstatě metoda umožňující získat obrázky, které se blíží realitě, je dalším krokem. Jde tedy o proces, který zahrnuje nejen fyziologickou stránku, ale též i psychologickou. Je známo, že fyzikální a chemické změny v oku vytvářejí podněty, které jsou přijímány mozkiem a interpretovány jako jas, barva atd., viz kap.2. Navíc jsou stejné vnější podněty různě interpretovány různými osobami. Proto jsou mnohá pravidla založena na výsledcích experimentů a nejsou přesně ve shodě s očekávanými zákonitostmi.

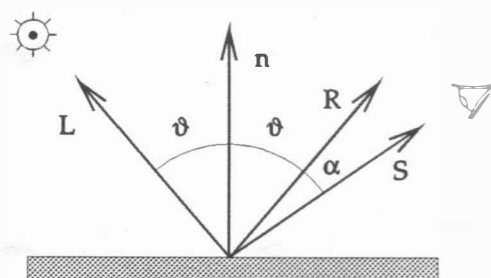
3.1 Modely osvětlení

Světlo dopadající na povrch fyzikálního tělesa je částečně pohlceno a přeměněno na teplo. V případě, že těleso pohlcuje veškeré světlo, je těleso neviditelné a nazývá se "absolutně černým tělesem". Zbytek světla je buď tělesem odražen nebo propuštěn. Viditelnost tělesa je tedy určena světlem odraženým a propuštěným, přičemž "množství" odraženého či propuštěného světla je závislé na vlnové délce. Pokud těleso odráží, resp. propouští, všechny vlnové délky stejně, pak se těleso při osvětlení bílým světlem jeví jako šedé, v ostatních případech barevné, viz kap.2.



rozptylující odraz

Obr. 3.1.1



zrcadlový odraz

Obr. 3.1.2

Odraz světla na povrchu může být dvojího charakteru, a to:

- Rozptylující (diffuse reflection), který světelné paprsky rozprostře do všech směrů, viz obr.3.1.1. Výsledným vjemem je pak matný povrch mající rovnoměrný jas, který je nezávislý na úhlu pozorování dané plochy.
- Zrcadlový (specular reflection), kdy se paprsek odrazí od povrchu, viz obr.3.1.2. Jas plochy je tedy závislý na úhlu pozorování dané plochy.

Na obr.3.1.1 a 3.1.2 jsou uvedeny typy odrazů světla od povrchu, přičemž:

- L je směrový vektor světelného zdroje
- n je normála plochy
- R je směrový vektor odraženého paprsku
- S je směrový vektor pozorovatele

Pro rozptylující typ odrazu platí tzv. Lambertův zákon, určující intenzitu rozptýleného světla v závislosti na úhlu ϑ , ve tvaru

$$I_d = k_d \cdot I_p \cdot \cos \vartheta$$

kde k_d je koeficient určující "množství" odraženého rozptýleného světla, přičemž $k_d \in \langle 0, 1 \rangle$

I_p je intenzita daného bodového zdroje světla

ϑ je úhel dopadu světelného paprsku, $\vartheta \in \langle -\pi/2, \pi/2 \rangle$

Pak lze psát

$$I_d = k_d \cdot I_p \cdot \hat{L}^T \cdot \hat{n}$$

kde \hat{L} a \hat{n} jsou normované vektory, tj.:

$$\hat{L} = \frac{\mathbf{L}}{|\mathbf{L}|} \quad \hat{n} = \frac{\mathbf{n}}{|\mathbf{n}|}$$

Uvedený výpočet intenzity je však ekvivalentní situaci, kdy dané těleso je uvnitř černé místnosti. Z důvodu přirozenějšího vjemu je nutné kromě odražené intenzity respektovat ještě intenzitu danou okolím (ambient light), tj. světlem odraženým od ostatních povrchů ve scéně, případně scénu ohraničujících (např. stěn místnosti). Pak v nejjednodušším iluminačním modelu lze intenzitu I určit:

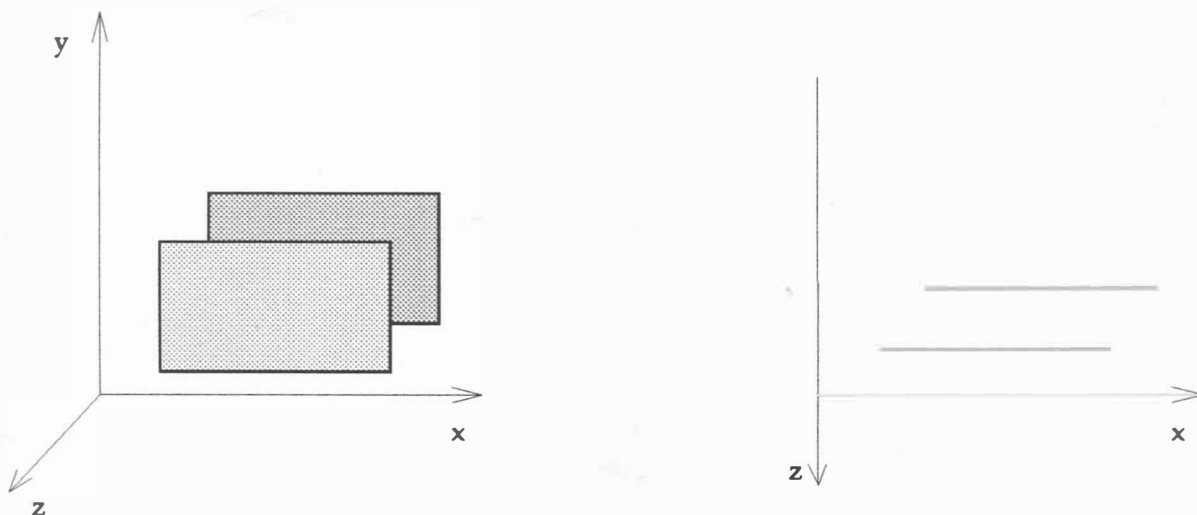
$$I = k_a \cdot I_a + k_d \cdot I_p \cdot \cos \vartheta$$

kde: k_a je koeficient určující vliv okolního světla, $k_a \in \langle 0, 1 \rangle$
 I_a intenzita okolního světla

Přepsáním do vektorového tvaru dostáváme

$$I = k_a \cdot I_a + k_d \cdot I_p \cdot \hat{L}^T \cdot \hat{n}$$

Pokud by nyní byly na scéně dvě stejné rovnoběžné plochy, viz obr.3.1.3, jež se z pozice pozorovatele částečně překrývají, byly by plochy od sebe nerozpoznatelné, i při různých vzdálenostech od pozorovatele.



Obr. 3. 1. 3

Jedním z důvodů tohoto jevu je nerespektování vlivu vzdálenosti na jas, neboť energie světla klesá s kvadrátem vzdálenosti zdroje světla od plochy.

Označíme-li tuto vzdálenost d , lze psát:

$$I = k_a \cdot I_a + k_d \cdot I_p \cdot \hat{L}^T \cdot \hat{n} / d^2$$

Nicméně výsledky, které bychom získali pomocí uvedené rovnice, jsou nerealistické, tj. neodpovídají skutečnosti. Při paralelní projekci je zdroj světla v nekonečnu, při perspektivní projekci hodnota $1/d^2$ pro $d \in (0, \infty)$ nabývá velkého rozsahu hodnot,

neboť pozice pozorovatele bývá blízko zobrazované scéně. Pro $d < 1$ dochází navíc k nežádoucímu zvýšení jasu.

Realističtějších výsledků lze docílit, pokud intenzita bude vyhodnocena pomocí rovnice

$$I = k_a \cdot I_a + k_d \cdot I_p \cdot \hat{L}^T \cdot \hat{n} / (k + d)$$

kde: k je libovolná kladná konstanta reprezentující vzdálenost zdroje světla od plochy

d je vzdálenost pozorovatele od zobrazovaného povrchu.

Pokud je povrch barevný, je nutné aplikovat uvedený vztah na jednotlivé složky použitého barevného systému, a tedy např. pro subtraktivní barevný systém CMY lze psát:

$$I_c = k_{a_c} \cdot I_{a_c} + k_{d_c} \cdot I_{p_c} \cdot \hat{L}^T \cdot \hat{n} / (k + d)$$

$$I_m = k_{a_m} \cdot I_{a_m} + k_{d_m} \cdot I_{p_m} \cdot \hat{L}^T \cdot \hat{n} / (k + d)$$

$$I_y = k_{a_y} \cdot I_{a_y} + k_{d_y} \cdot I_{p_y} \cdot \hat{L}^T \cdot \hat{n} / (k + d)$$

Je nutné podotknout, že použití subtraktivního barevného systému (CMY) při výstupu není jednoduché vzhledem k tomu, že vlivem nepřesností daných fyzikální realizací zařízení dochází k mísení principu subtraktivního a aditivního. Analogické rovnice lze psát pro jednotlivé složky r, g, b , pokud se zvolí tyto tři základní barvy, tj. systém RGB.

Dalším typem odrazu je zrcadlový odraz, který je na rozdíl od rozptylujícího odrazu velmi směrový, viz obr.3.1.2. Přísně vzato by vektor R měl být totožný s vektorem S . Empirický model osvětlení pro "reálné" povrchy, který vytvořil Bui-Tuong Phong, je definován rovnicí

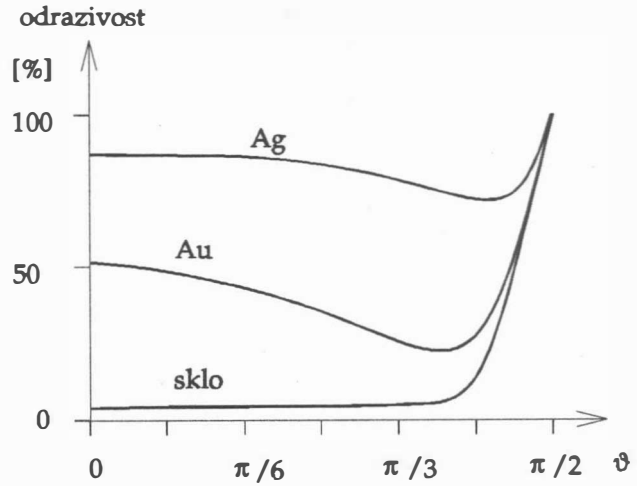
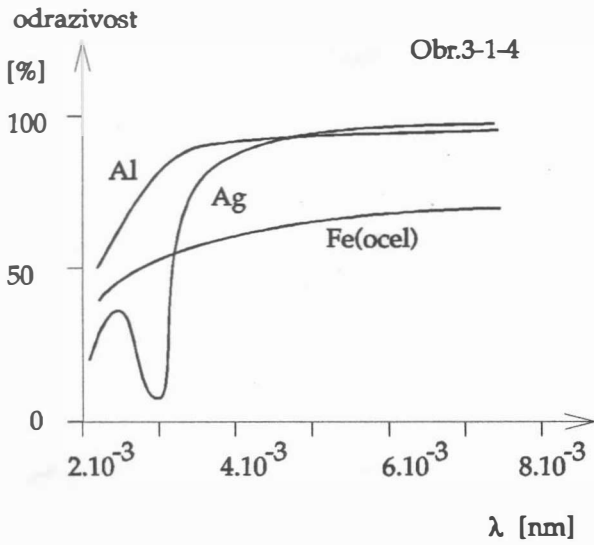
$$I_s = I_p \cdot w(\vartheta, \lambda) \cdot \cos^n \alpha$$

kde I_s je intenzita odraženého světla

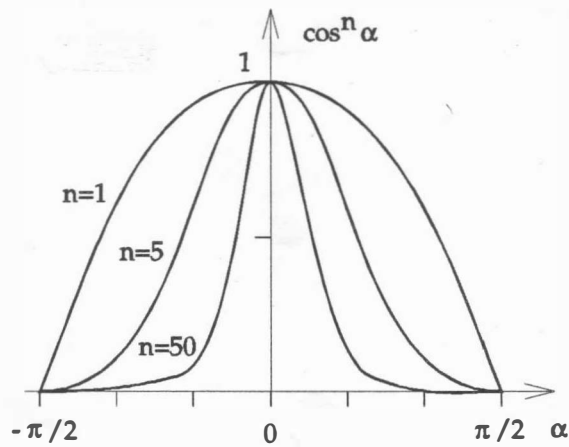
$w(\vartheta, \lambda)$ je odrazová křivka určující "množství" odraženého světla v závislosti na vlnové délce λ , viz obr.3.1.4

α je úhel mezi vektory R a S , viz obr.3.1.2

n určuje charakteristiku povrchu, tj. jak rychle "mizí" jev úplného odrazu, viz obr.3.1.5; hodnota $n \in \langle 0, \infty \rangle$ podle materiálu.



Obr. 3. 1. 4



Obr. 3. 1. 5

Vzhledem ke své složitosti je funkce $w(\vartheta, \lambda)$ nahrazena konstantou k_s . Pak lze psát

$$I_s = k_s \cdot I_p \cdot \cos^n \alpha = k_s \cdot I_p \cdot (\hat{R}^T \cdot \hat{S})^n$$

kde \hat{R} a \hat{S} jsou normalizované vektory R a S , tj.:

$$\hat{R} = \frac{R}{|R|} \quad \hat{S} = \frac{S}{|S|}$$

I pro zrcadlový odraz je nutné respektovat vliv vzdálenosti světelného zdroje na intenzitu světla. Intenzita světla je dána prostým součtem všech složek, tj.:

$$I = k_a \cdot I_a + I_p \cdot \left[k_d \cdot \cos \vartheta + k_s \cdot \cos^n \alpha \right] / (k + d)$$

Přepsáním do vektorového tvaru dostáváme

$$I = k_a \cdot I_a + I_p \cdot \left[k_d \cdot (\hat{L}^T \cdot \hat{n}) + k_s \cdot (\hat{R}^T \cdot \hat{S})^n \right] / (k + d)$$

kde \hat{L} , \hat{n} , \hat{R} , \hat{S} jsou normalizované vektory.

V případě více zdrojů světla dostáváme:

$$I = k_a \cdot I_a + \sum_{i=1}^m I_{p_i} \cdot \left[k_d \cdot \cos \vartheta_i + k_s \cdot \cos^n \alpha_i \right] / (k + d)$$

kde: m je počet světelných zdrojů.

Ve vektorovém tvaru pak:

$$I = k_a \cdot I_a + \sum_{i=1}^m I_{p_i} \cdot \left[k_d \cdot (\hat{L}_i^T \cdot \hat{n}) + k_s \cdot (\hat{R}_i^T \cdot \hat{S})^n \right] / (k + d)$$

Uvedený výraz je možné pro určité situace zjednodušit, např. mají-li všechny zdroje stejnou intenzitu.

V případě uvažování barvy je nutné opět uvedenou rovnici aplikovat na základní barvy použitého barevného systému, přičemž koeficienty k_d , k_s , k_a budou obecně záviset na vlnové délce, a budou tedy rozdílné pro jednotlivé základní barvy.

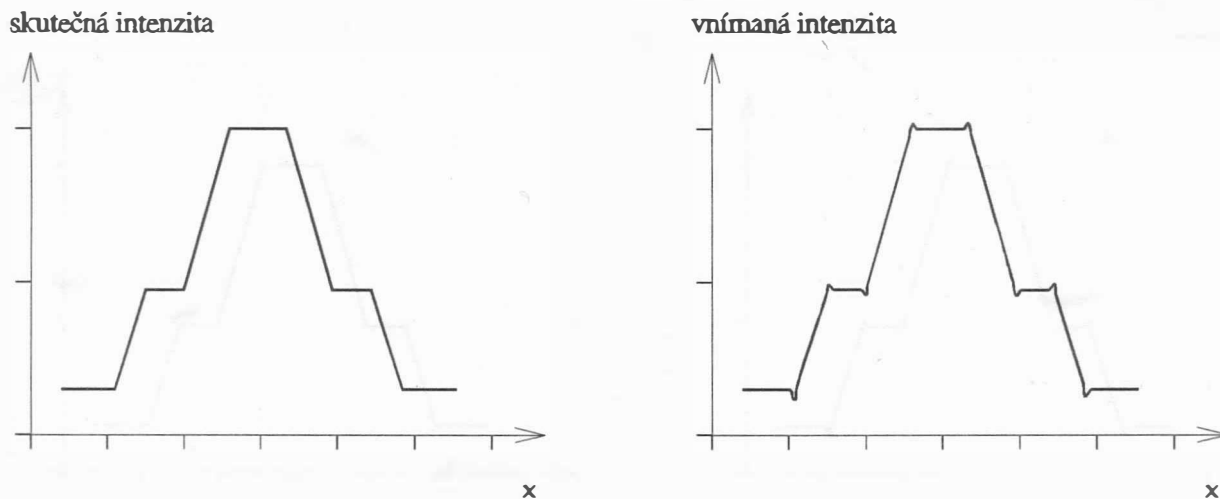
3.2 Metody stínování

Konstantní stínování

Technika konstantního stínování vychází ze zjednodušujících předpokladů, kdy

- zdroje světla jsou v nekonečnu, takže součin $L^T \cdot n$ je konstantní pro celý zobrazovaný n-úhelník,
- pozorovatel je též v nekonečnu, a tedy součin $R^T \cdot s$ je též konstantní.

Konstantní stínování je postačující k zobrazení těles ohraničených lineárními ploškami, přičemž prakticky nepřináší zvýšení výpočetní složitosti, neboť jas, resp. barva je konstantní v rámci daného n-úhelníka. Nicméně tato technika je nevhodná pro zobrazování neplanárních povrchů, které jsou pouze aproximovány lineárními ploškami. V tomto případě se na styku dvou plošek s různými normálami intenzita mění skokem. Tuto změnu intenzity oko vnímá jako hranu, která na původním oblém tělese neexistuje, viz obr.3.2.1.



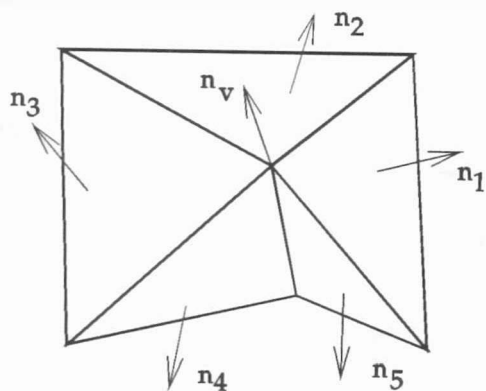
Obr. 3.2.1

Tento jev působící dosti rušivě se nazývá Machův efekt podle svého objevitele Ernsta Macha (*1838 +1916), původem z Chrlic u Brna. Nežádoucí efekt lze částečně eliminovat interpolací jasu.

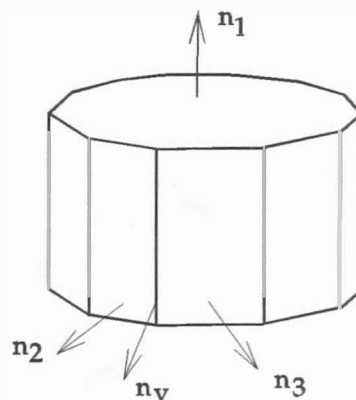
Přes uvedené nevýhody tato technika pro svoji výpočetní nenáročnost v mnoha aplikacích plně postačuje.

Gouraudovo stínování

Metoda Gouraudova stínování [152] je založena na interpolaci jasu vycházející ze znalosti normály plošek ve vrcholech n-úhelníka sítě, která zakřivenou plochu aproximuje. Normála ve vrcholu se určuje jako aritmetický průměr normál plošek, které obsahují daný vrchol.



Obr. 3.2.2



Obr. 3.2.3

Např. normála n_v z obr.3.2.2 je pak určena:

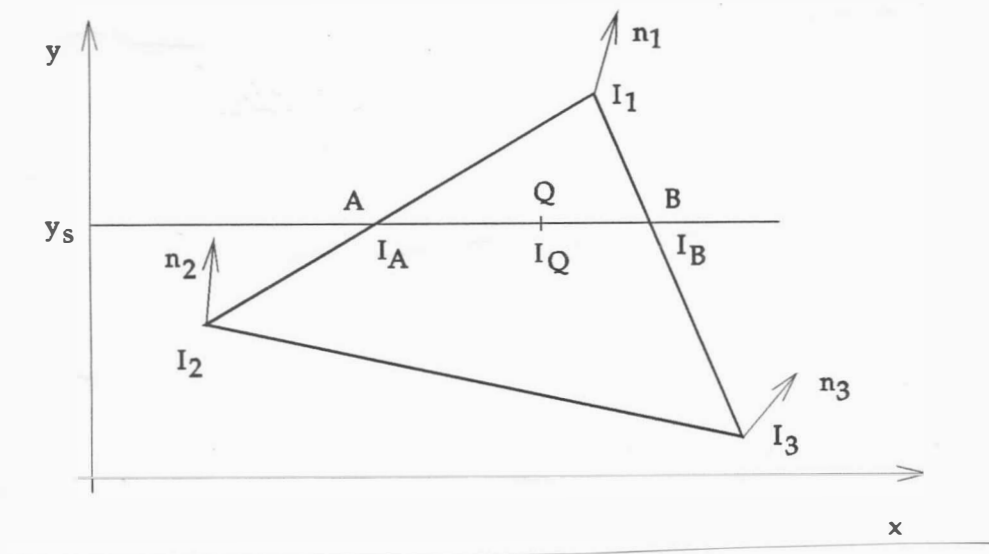
$$n_v = (n_1 + n_2 + n_3 + n_4 + n_5) / 5$$

Tento výpočet je možný z toho důvodu, že zobrazujeme zakřivenou plochu, která je aproximována lineárními ploškami a která neobsahuje hranu.

V případě existence hran, např. válec, kužel apod., je nutné použít k výpočtu normálového vektoru vrcholu jen normály plošek aproximujících příslušnou zakřivenou plochu. Např. v situaci na obr.3.2.3 plocha s normálou n_1 nesmí být použita k výpočtu normálového vektoru vrcholu n_v , a tedy:

$$n_v = (n_2 + n_3) / 2$$

Nyní je možné provést lineární interpolaci jasu pro zobrazovaný n-úhelník (většinou trojúhelník), neboť jsou známy intenzity ve vrcholech. Při realizaci se k minimalizaci výpočetních nároků s výhodou využívá toho, že hodnoty I_A a I_B se v rámci zobrazovaného řádku nemění, viz obr.3.2.4.



Obr. 3.2.4

Hodnoty I_a a I_b lze pak vypočítat:

$$I_A = \left[I_1 \cdot (y_s - y_2) + I_2 \cdot (y_1 - y_s) \right] / (y_1 - y_2)$$

$$I_B = \left[I_1 \cdot (y_s - y_3) + I_3 \cdot (y_1 - y_s) \right] / (y_1 - y_3)$$

Hodnotu intenzity I_p pro daný pixel ležící na řádce y_s lze určit

$$I_Q = \left[I_A \cdot (x_B - x_Q) + I_B \cdot (x_Q - x_A) \right] / (x_B - x_A)$$

nebo alternativně:

$$I_A = I_1 + (I_2 - I_1) \cdot u$$

$$I_B = I_1 + (I_3 - I_1) \cdot w$$

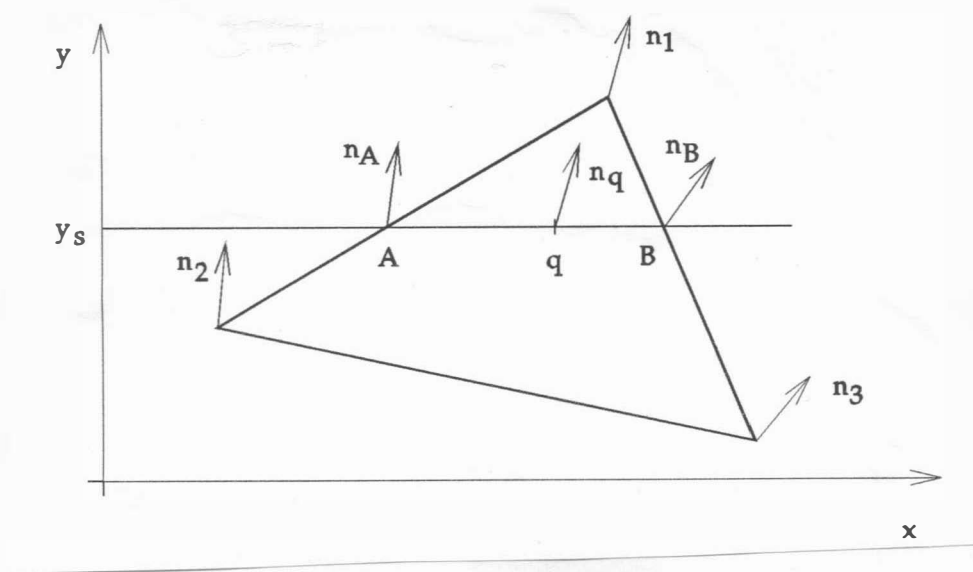
$$I_Q = I_A + (I_B - I_A) \cdot t \quad u, w, t \in \langle 0, 1 \rangle$$

V případě barevného výstupu je nutné uvedený postup aplikovat na jednotlivé primární barvy daného barevného systému.

Phongovo stínování

Na rozdíl od Gouraudova stínování, které interpoluje intenzitu, resp. barvu v rámci zobrazovaného řádku, Phongovo stínování [150] interpoluje normálový vektor podél zobrazovaného řádku, čímž je zajištěna lepší aproximace křivosti povrchu podél řádky, a tedy výsledný vjem se blíží očekávanému, zejména pokud se týče odrazů.

Při Phongově stínování je opět nutné určit normálové vektory vrcholů příslušejících jednotlivým ploškám, viz obr.3.2.5, podobně jako v případě Gouraudova stínování.



Obr. 3. 2. 5

Pro normálové vektory lze psát:

$$\mathbf{n}_A = \mathbf{n}_1 + (\mathbf{n}_2 - \mathbf{n}_1) \cdot u \quad u \in \langle 0, 1 \rangle$$

$$\mathbf{n}_B = \mathbf{n}_1 + (\mathbf{n}_3 - \mathbf{n}_1) \cdot w \quad w \in \langle 0, 1 \rangle$$

$$\mathbf{n}_Q = \mathbf{n}_A + (\mathbf{n}_B - \mathbf{n}_A) \cdot t \quad t \in \langle 0, 1 \rangle$$

přičemž pro \mathbf{n}_{q_1} a \mathbf{n}_{q_2} v rámci daného řádku lze psát

$$\mathbf{n}_{q_2} = \mathbf{n}_{q_1} + (\mathbf{n}_B - \mathbf{n}_A) \cdot (t_2 - t_1)$$

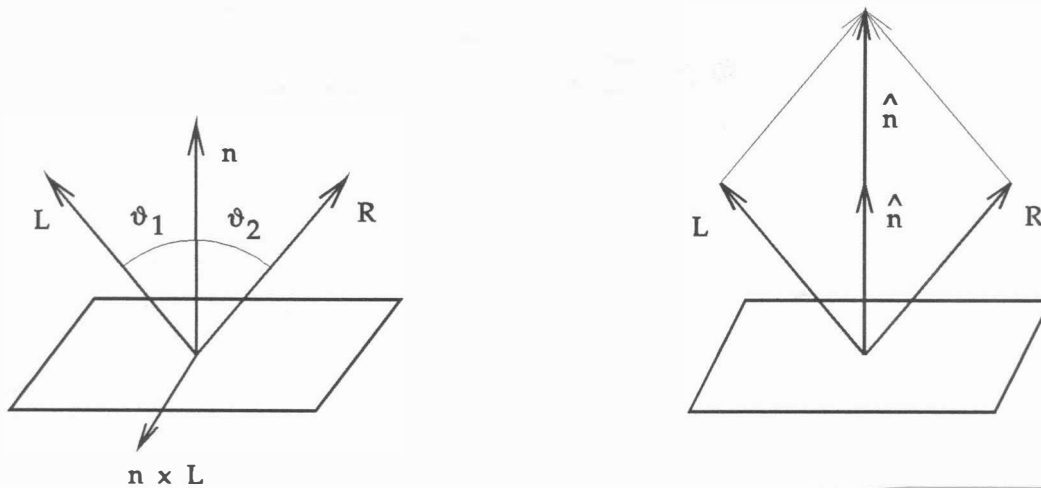
kde t_1 a t_2 jsou odpovídající hodnoty parametru t v bodech q_1 a q_2 .

Normálový vektor lze pro každý bod v řádku určit jako:

$$n_{q_2} = n_{q_1} + \Delta n \cdot \Delta t$$

Z uvedeného vyplývá, že výpočtové nároky při použití Phongova stínování vzrostou, neboť je nutné počítat normálu plochy a odpovídající intenzitu v každém bodě.

Lze nahlédnout, že vektor R je závislý na vektoru L a normále povrchu n , viz obr.3.2.6.a.



Obr. 3.2.6

Lze ukázat, že:

$$L \times n = n \times R \quad \text{nebo} \quad n \times L = R \times n$$

rozepsáním vektorových součinů dostáváme pro jednotlivé složky:

$$L_y n_z - L_z n_y = n_y R_z - n_z R_y$$

$$L_z n_x - L_x n_z = n_z R_x - n_x R_z$$

$$L_x n_y - L_y n_x = n_x R_y - n_y R_x$$

Uvedená soustava rovnic však nemá jedno řešení, neboť rovnice soustavy jsou závislé, tj. hodnota matice soustavy není rovna 3. Je tedy nutné nalézt další dodatečnou rovnici. Vzhledem k tomu, že úhel odrazu se musí rovnat úhlu dopadu, tj. $\vartheta_1 = \vartheta_2$, musí platit:

$$L^T \cdot n = n^T \cdot R$$

tj.

$$L_x n_x + L_y n_y + L_z n_z = n_x R_x + n_y R_y + n_z R_z$$

Z obr.3.2.6 lze odvodit, že obecně pro $\vartheta_1 \neq \vartheta_2$ lze určit jednoparametrické řešení vektorů R takové, že $L \times n = n \times R$. Přepíšeme-li uvedené rovnice do maticového tvaru pro vektor $R = [R_x, R_y, R_z]^T$, dostáváme :

$$\begin{bmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \\ n_x & n_y & n_z \end{bmatrix} \cdot R = \begin{bmatrix} n_z L_y - n_y L_z \\ n_x L_z - n_z L_x \\ n_y L_x - n_x L_y \\ n_x L_x + n_y L_y + n_z L_z \end{bmatrix}$$

Řešení v uvedeném případě je možné, neboť jedna rovnice je nadbytečná. Obecně je nutné soustavu rovnic typu $A \cdot x = b$ řešit pomocí metody nejmenších čtverců, pokud je počet nezávislých rovnic větší než počet neznámých, tj.:

$$x = (A^T \cdot A)^{-1} \cdot A^T \cdot b$$

Výhodněji je možné vektor \hat{R} a R určit pomocí vztahů

$$\hat{R} = 2 \cdot (\hat{L}^T \cdot \hat{n}) \cdot \hat{n} - \hat{L}$$

$$R = |L| \cdot \hat{R}$$

viz obr.3.2.6.b:

Příklad

Určete vektor R , je-li dáno:

$$n = [0, 1, 0]^T \quad a \quad L = [-1, 2, -1]^T$$

Pak

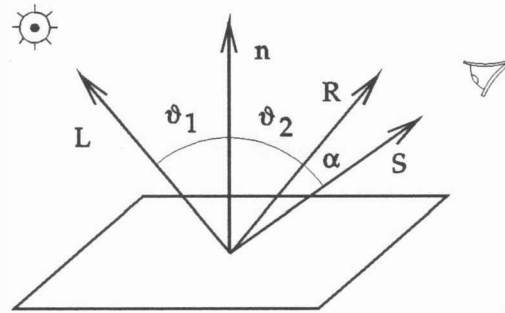
$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \\ 2 \end{bmatrix}$$

Úpravami dostáváme (a vyškrtnutím 2.řádku):

$$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}$$

a tedy výsledný vektor:

$$\mathbf{R} = [1 , 2 , 1]^T$$



Obr. 3.2.7

Příklad

Určete intenzitu vnímanou pozorovatelem, viz obr.3.2.7, pokud:

$$\mathbf{n} = [0 , 1 , 0]^T$$

$$\mathbf{L} = [-1 , 2 , -1]^T$$

$$\mathbf{S} = [1 , 1,5 , 0,5]^T$$

Lze ukázat (viz předchozí příklad), že:

$$\mathbf{R} = [1 , 2 , 1]^T$$

Dále předpokládejme, že poměr intenzity okolního světla k intenzitě zdroje světla je 1:10, tj. např.:

$$I_a = 1$$

a

$$I_p = 10$$

Pro kovový povrch nabývají konstanty hodnot:

$$k_s = 0,80$$

$$k_d = 0,15$$

$$k_a = 0,15$$

$$n = 5$$

(povrch absorbuje 5% energie, neboť $k_s + k_d = 0,95$).

Je-li $d = 0$ a $k = 1$, pak

$$\hat{\mathbf{L}}^T \cdot \hat{\mathbf{n}} = \frac{\mathbf{L}^T \cdot \mathbf{n}}{|\mathbf{L}| \cdot |\mathbf{n}|} = \frac{2}{\sqrt{6}}$$

tj.

$$\vartheta = \arccos(2 / \sqrt{6}) = 35^\circ 26'$$

Dále

$$\hat{\mathbf{R}}^T \cdot \hat{\mathbf{S}} = \frac{\mathbf{S}^T \cdot \mathbf{R}}{|\mathbf{S}| \cdot |\mathbf{R}|} = 4,5 / \sqrt{21}$$

$$\alpha = \arccos(4,5 / \sqrt{21}) = 10^\circ 89'$$

$$\begin{aligned} I &= k_a \cdot I_a + I_p \cdot \left[k_d \cdot (\hat{\mathbf{L}}^T \cdot \hat{\mathbf{n}}) + k_s \cdot (\hat{\mathbf{R}}^T \cdot \hat{\mathbf{S}})^n \right] / (k + d) = \\ &= 0,15 + 10 \cdot \left[0,15 \cdot (2 / \sqrt{6}) + 0,8 \cdot (4,5 / \sqrt{21})^5 \right] = \\ &= 8,65 \end{aligned}$$

Vzhledem k tomu, že vektor pozorovatele \mathbf{S} je téměř totožný s vektorem odrazu \mathbf{R} , je výsledkem velký jas dané plochy. Změní-li se však poloha pozorovatele tak, že

$$\mathbf{S} = [1, 1,5, -0,5]^T$$

pak

$$\hat{\mathbf{R}}^T \cdot \hat{\mathbf{S}} = 3,5 / \sqrt{21}$$

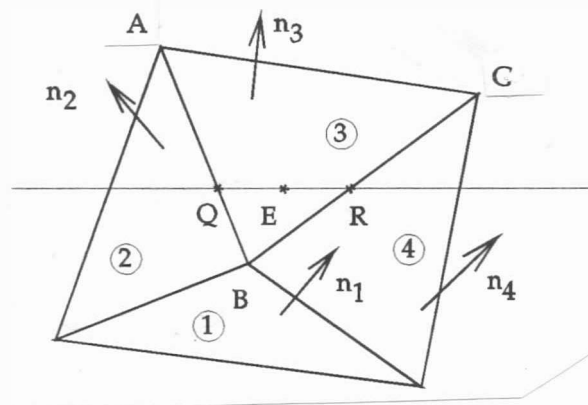
a tedy

$$\alpha = 40,2^\circ$$

Pak výsledná intenzita je rovna

$$I = 0,15 + 10 \cdot (0,12 + 0,21) = 3,45$$

což je podstatně menší úroveň jasu.



Obr. 3.2.8

Příklad

Ukažte rozdíl mezi jednotlivými modely osvětlení pro povrch, který je aproximován rovinnými ploškami [109], viz obr. 3.2.8:

$$\begin{array}{rcll}
 1: & & 2z & - 4 & = 0 \\
 2: & -x & + 1,732y & + 7,5z & - 17 & = 0 \\
 3: & - 2,25x & + 3,897y & + 10z & - 24,5 & = 0 \\
 4: & & & 5,5z & - 11 & = 0
 \end{array}$$

Dále pak:

$$\begin{array}{ll}
 \mathbf{S} = [1, 1, 1]^T & \mathbf{L} = [0, 0, 1]^T \\
 d = 0 & k = 1 & I_a = 1 & I_p = 10 \\
 n = 2 & k_s = 0,8 & k_d = 0,15 & k_a = 0,15
 \end{array}$$

Pro konstantní stínování je nezbytné určit normálový vektor $\hat{\mathbf{n}}_3$:

$$\hat{\mathbf{n}}_3 = \frac{\mathbf{n}_3}{|\mathbf{n}_3|} = [-0,21, 0,36, 0,91]^T$$

Lze opět ukázat, že:

$$\begin{array}{l}
 \hat{\mathbf{R}} = [-0,38, 0,66, 0,66]^T \\
 \hat{\mathbf{S}} = \frac{\mathbf{S}}{|\mathbf{S}|} = \frac{1}{\sqrt{3}} [1, 1, 1]^T = [0,58, 0,58, 0,58]^T
 \end{array}$$

Pak

$$\hat{\mathbf{L}}^T \cdot \hat{\mathbf{n}} = 0,91 \quad \text{a} \quad \hat{\mathbf{R}}^T \cdot \hat{\mathbf{S}} = 0,55$$

Dosazením do rovnice pro výpočet intenzity pro konstantní stínování dostáváme :

$$\begin{aligned}
 I &= k_a \cdot I_a + I_p \cdot \left[k_d \cdot (\hat{\mathbf{L}}^T \cdot \hat{\mathbf{n}}) + k_s \cdot (\hat{\mathbf{R}}^T \cdot \hat{\mathbf{S}})^n \right] / (k + d) = \\
 &= 0,15 + 10 \cdot [0,14 + 0,24] / 1,0 = 3,95
 \end{aligned}$$

Celá plocha bude tedy zobrazena s intenzitou:

$$I = 3,95$$

Pro Gouraudovo stínování je nutné nejdříve určit normály ve vrcholech A, B, C, tj.:

$$\begin{aligned}
 \mathbf{n}_A &= \mathbf{n}_2 + \mathbf{n}_3 = [-3,25, 5,63, 17,5]^T \\
 \mathbf{n}_B &= \mathbf{n}_1 + \mathbf{n}_2 + \mathbf{n}_3 + \mathbf{n}_4 = [-3,25, 5,63, 25]^T \\
 \mathbf{n}_C &= \mathbf{n}_3 + \mathbf{n}_4 = [-2,25, 3,897, 15,5]^T
 \end{aligned}$$

Normalizované vektory:

$$\hat{n}_A = n_A / |n_A| = [-0,17, 0,3, 0,94]^T$$

$$\hat{n}_B = n_B / |n_B| = [-0,12, 0,22, 0,97]^T$$

$$\hat{n}_C = n_C / |n_C| = [-0,14, 0,24, 0,96]^T$$

Pak směrové vektory odražených paprsků jsou určeny:

$$\hat{R}_A = [-0,33, 0,57, 0,76]^T$$

$$\hat{R}_B = [-0,24, 0,42, 0,87]^T$$

$$\hat{R}_C = [-0,27, 0,46, 0,84]^T$$

Pak odpovídající intenzity ve vrcholech A, B, C jsou opět určeny rovnicí

$$I = k_a \cdot I_a + I_p \cdot \left[k_d \cdot (\hat{L}^T \cdot \hat{n}) + k_s \cdot (\hat{R}^T \cdot \hat{S})^n \right] / (k + d)$$

a tedy:

$$I_A = 0,15 + 10 \cdot [0,14 + 0,27] / 1 = 4,25$$

$$I_B = 0,15 + 10 \cdot [0,15 + 0,30] / 1 = 4,65$$

$$I_C = 0,15 + 10 \cdot [0,14 + 0,29] / 1 = 4,45$$

Označíme-li

$$u_Q = AQ / AB = 0,6$$

$$w_R = CR / BC = 0,3$$

pak interpolováním intenzit dostáváme pro bod Q, resp. R:

$$I_Q = I_A + (I_B - I_A) \cdot u = 4,25 + (0,4) \cdot 0,6 = 4,49$$

$$I_R = I_C + (I_B - I_C) \cdot w = 4,65 + (-0,2) \cdot 0,3 = 4,59$$

Je-li nyní bod E na řádku v pozici odpovídající hodnotě parametru $t_E = QE/QR = 0,5$, pak dostáváme:

$$I_E = I_Q + (I_R - I_Q) \cdot t = 4,49 + (0,1) \cdot 0,5 = 4,54$$

V případě použití Gouraudova stínování bude bod E aktivní s intenzitou:

$$I_E = 4,54$$

V případě Phongova stínování je nutné interpolovat normály a dosazením za u dostáváme:

$$\hat{n}_Q = \hat{n}_A + (\hat{n}_B - \hat{n}_A) \cdot u =$$

$$[-0,17, 0,3, 0,94]^T + \left([0,05, -0,08, 0,03]^T \right) \cdot 0,6 =$$

$$[-0,14, 0,25, 0,96]^T$$

Analogicky dosazením za w :

$$\hat{n}_R = \hat{n}_B + (\hat{n}_C - \hat{n}_B) \cdot w = [-0,04, 0,23, 0,97]^T$$

Interpolací normály v rámci řádky po dosazení za t dostáváme:

$$\hat{n}_E = \hat{n}_Q + (\hat{n}_R - \hat{n}_Q) \cdot t = [-0,09, 0,24, 0,97]^T$$

Lze ukázat, že:

$$\hat{R}_E = [-0,17, 0,46, 0,87]^T$$

(je nutné řešit dříve uvedenou soustavu rovnic, nebo použít dříve uvedené vzorce pro výpočet \hat{R}).

Pak intenzita I_E je určena opět rovnicí

$$I = k_a \cdot I_a + I_p \cdot \left[k_d \cdot (\hat{L}^T \cdot \hat{n}) + k_s \cdot (\hat{R}^T \cdot \hat{S})^n \right] / (k + d)$$

dosazením pak:

$$I_E = 0,15 + 10 \cdot [0,15 + 0,36] / 1 = 5,25$$

V případě Phongova stínování bude bod E aktivován s intenzitou:

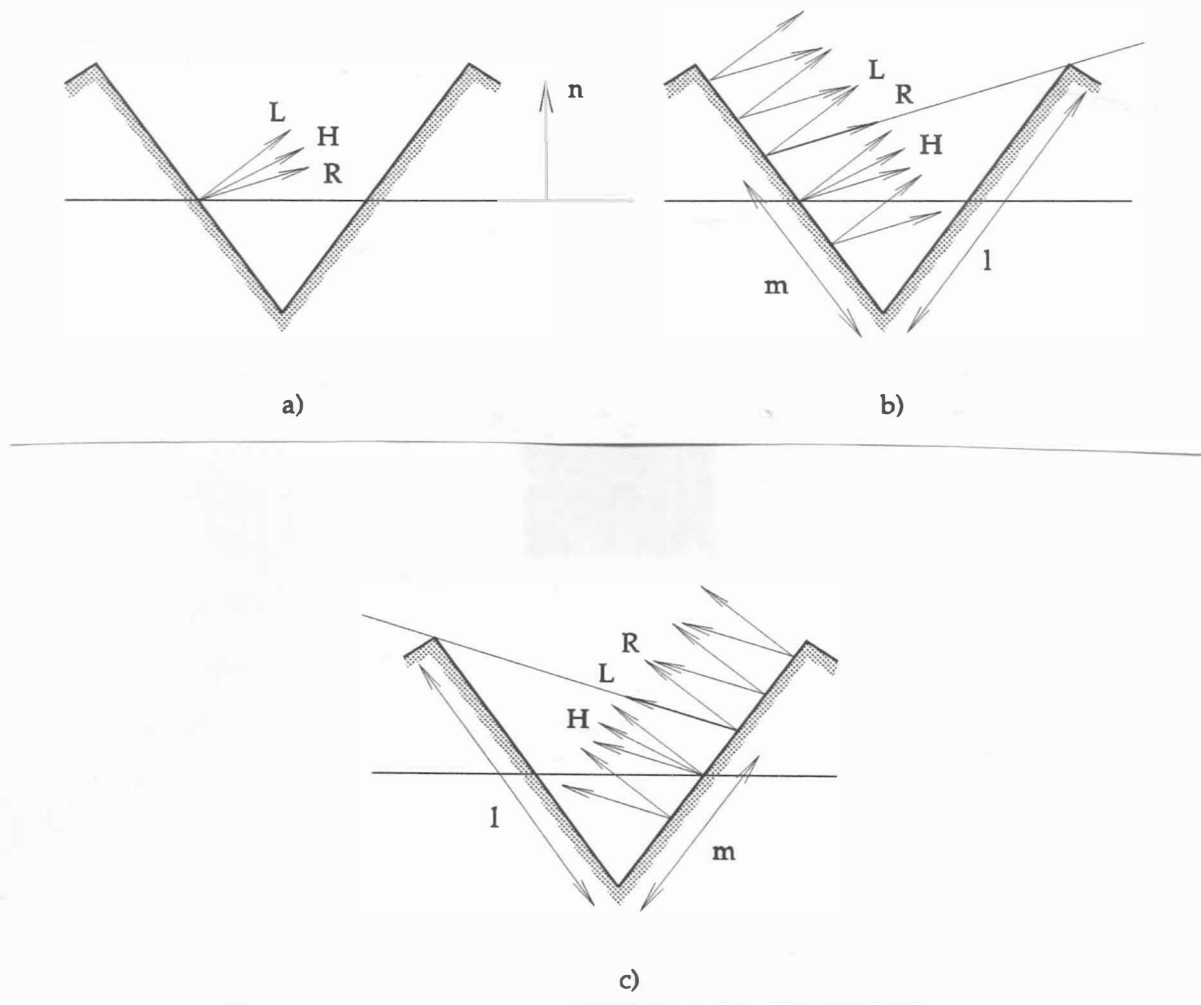
$$I_E = 4,54$$

Porovnáme-li získané výsledky, pak dostáváme následující hodnoty intenzit v bodě E pro jednotlivé typy stínování:

konstantní	$I_E = 3,93$
Gouraudovo	$I_E = 4,54$
Phongovo	$I_E = 5,25$

Model Torrance - Sparrow

Až dosud uvedené modely osvětlení aproximovaly více či méně realitu. Odraz světla od reálného povrchu byl předmětem mnoha experimentů, které lze datovat až do r.1910 a dokonce ještě dříve. Výsledky z Gouraudova a Phongova modelu se přibližují k experimentálně získaným hodnotám, v určitých situacích jsou však rozdíly dosti značné. V roce 1967 byl odvozen tzv. Torrance - Sparrowův model, viz [157],[158], který velmi dobře aproximuje experimentálně získané hodnoty, ale je výpočtově poměrně náročný.



Obr. 3.2.9

Tento model vychází z představy, že povrch není ideálně hladký, ale že se skládá z "mikroplošek", které způsobují zrcadlový odraz. Jednotlivé "mikroplošky" jsou orientovány

náhodně a odraz na ploše se určí jako celkový příspěvek od všech "mikroplošek".

V obecném případě mohou nastat tyto případy:

- nenastává žádná interference odraženého paprsku s ostatními "mikroploškami", viz obr. 3.2.9.a
- nastává "maskování" odraženého světla, viz obr. 3.2.9.b, tj. odražený paprsek dopadá na jinou "mikroplošku" a není přímo vnímán pozorovatelem
- dochází k tvoření stínu vlivem zastínění části plochy jinou "mikroploškou", viz obr. 3.2.9.c (úsek označený m)

Zaveďme následující označení (viz obr. 3.2.11):

\mathbf{n} normálu plochy bez uvažování "mikroplošek"

\mathbf{L} směrový vektor zdroje světla

\mathbf{R} směrový vektor odraženého paprsku

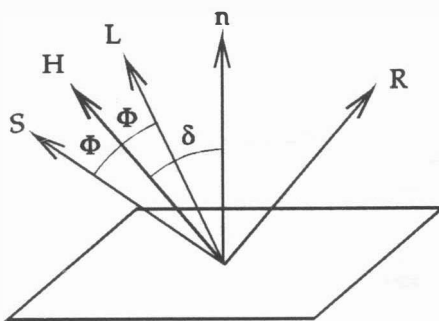
\mathbf{S} směrový vektor pozorovatele

\mathbf{H} normálový vektor "mikroplošky"

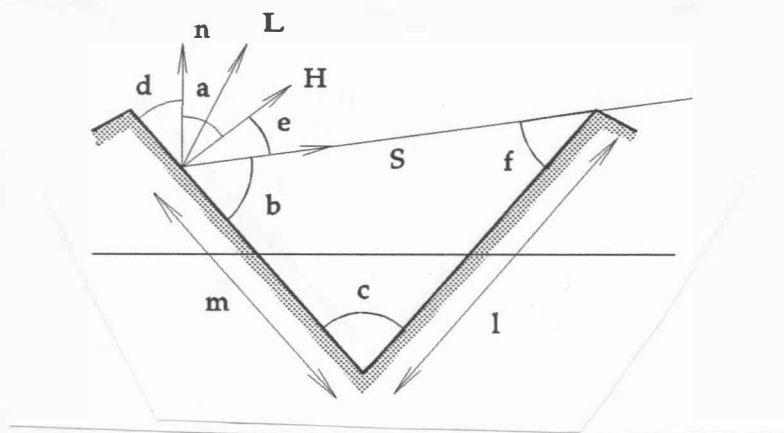
l (malé L) délka hrany "mikroplošky" (udává vlastně hrubost povrchu - hodnota se volí)

m určuje část "mikroplošky", která se svým odrazem podílí na osvětlení sousední "mikroplošky", viz obr. 3.2.9.b, nebo část, která určuje zastíněnou část "mikroplošky", viz obr. 3.2.9.c

$\hat{\mathbf{n}}, \hat{\mathbf{L}}, \hat{\mathbf{R}}, \hat{\mathbf{S}}, \hat{\mathbf{H}}$ normované tvary odpovídajících vektorů.



Obr. 3.2.10



Obr. 3.2.11

Z obr. 3.2.10 vyplývá:

$$\cos \phi = \frac{\hat{H} \cdot \hat{L}}{|\hat{H}| |\hat{L}|} = \frac{\hat{S} \cdot \hat{H}}{|\hat{S}| |\hat{H}|}$$

$$H = \frac{L + S}{|\hat{L} + \hat{S}|} = \frac{\hat{L} + \hat{S}}{2}$$

To znamená, že pouze "mikroploška" s normálou \hat{H} může přispět k celkové intenzitě vnímané pozorovatelem. V prvním případě, tj. v případě ad a, nenastává žádné zeslabení intenzity světla, zatímco v ostatních případech zeslabení nastává.

Podle [16],[157] je odraz obecně určen výrazem:

$$k_s = \frac{D G F}{n^2 \hat{S}}$$

kde D je distribuční funkce směrů "mikroplošek" daného povrchu
 F koeficient odrazivosti respektující Fresnelův zákon odrazu
 G je funkce určující zeslabení intenzity vlivem maskování a zastínění

Blinn [15],[16] určil, že pro případ, kdy "mikroplošky" mají tvar symetrického V , lze psát (význam jednotlivých proměnných viz obr. 3.2.11):

$$\frac{m}{l} = \frac{\sin f}{\sin b}$$

$$d + a = \pi / 2$$

$$c = 2 d$$

$$f + b + c = \pi$$

Použitím výše uvedených předpokladů a trigonometrických identit lze ukázat, že:

- pro případ maskování, viz obr. 3.2.11, platí

$$\frac{m}{1} = \frac{\hat{H}^T \hat{S} - 2 \cdot (\hat{n}^T \hat{H}) \cdot (\hat{n}^T \hat{S})}{\hat{H}^T \hat{S}}$$

Označíme-li

$$G_m = \frac{2 \cdot (\hat{n}^T \hat{H}) \cdot (\hat{n}^T \hat{S})}{\hat{H}^T \hat{S}}$$

pak poměrné zeslabení intenzity vnímané pozorovatelem lze vyjádřit výrazem $(1 - m) / 1$, a tedy:

$$1 - \frac{m}{1} = G_m$$

Pro případ stínování pak analogicky

$$G_s = \frac{2 \cdot (\hat{n}^T \hat{H}) \cdot (\hat{n}^T \hat{L})}{\hat{H}^T \hat{L}}$$

$$1 - \frac{m}{1} = G_m$$

Celkové zeslabení intenzity je pak určeno výrazem [16]:

$$G = \min \{ 1, G_m, G_s \}$$

Distribuční funkce D byla ve Phongově modelu představována funkcí ve tvaru $\cos^n \delta$. Použitý Torrance-Sparrowův model předpokládá Gaussovo rozložení

$$D = c_1 \cdot e^{-(\delta \cdot c)^2}$$

kde D je počet "mikroplošek" orientovaných ve směru odchýleném o úhel δ od normály n

c, c_1 jsou volitelné kladné konstanty. Volbou volíme "drsnotu" povrchu.

Beckmann [155] ukázal, že D lze též určit jako:

$$D = \frac{1}{m^2 \cdot \cos^4 \delta} e^{-(\operatorname{tg} \delta / m^2)^2}$$

přičemž není nutné volit žádnou konstantu. Nicméně existují

i jiné distribuční funkce, které jsou vhodné např. pro skleněné předměty [159].

Pro výpočet k_s je nezbytné určit hodnotu odrazivosti F podle Fresnelova zákona, tj.

$$F = \frac{1}{2} \cdot \left[\frac{\sin^2(\phi - \vartheta)}{\sin^2(\phi + \vartheta)} + \frac{\operatorname{tg}^2(\phi - \vartheta)}{\operatorname{tg}^2(\phi + \vartheta)} \right]$$

kde ϕ je incidenční úhel,

η je index odrazu

$$\cos \phi = \hat{\mathbf{L}}^T \cdot \hat{\mathbf{H}} \cdot \sin \vartheta = \sin \phi / \eta$$

Vzhledem k tomu, že index odrazu η je závislý na vlnové délce, tj. $\eta = \eta(\lambda)$, je též F funkcí vlnové délky. Pokud funkce $\eta(\lambda)$ není známá, může být určena z experimentálně získaných dat.

Cook a Torrance [156] navrhli, aby funkce F byla určena jako

$$F = \frac{1}{2} \cdot \left(\frac{c - g}{c + g} \right)^2 \cdot \left\{ 1 + \frac{[c \cdot (g + c) - 1]^2}{[c \cdot (g - c) + 1]^2} \right\}$$

kde $c = \cos \phi = \hat{\mathbf{L}}^T \cdot \hat{\mathbf{H}}$

$$g^2 = \eta^2 + c^2 - 1$$

Pro $\phi = 0$, $c = 1$, $g = \eta$ lze ukázat [109] (označme $F_0 = F_{\phi=0}$), že

$$F_0 = \left(\frac{\eta - 1}{\eta + 1} \right)^2$$

Je-li funkce $F_0(\lambda)$ známá, lze vyjádřit $\eta(\lambda)$ jako

$$\eta(\lambda) = \frac{1 + \sqrt{F_0(\lambda)}}{1 - \sqrt{F_0(\lambda)}}$$

Uvedený postup je nutné aplikovat pro každou mikroplošku. Celkový odraz na ploše se určí jako celkový příspěvek od všech "mikroplošek". Nyní lze psát

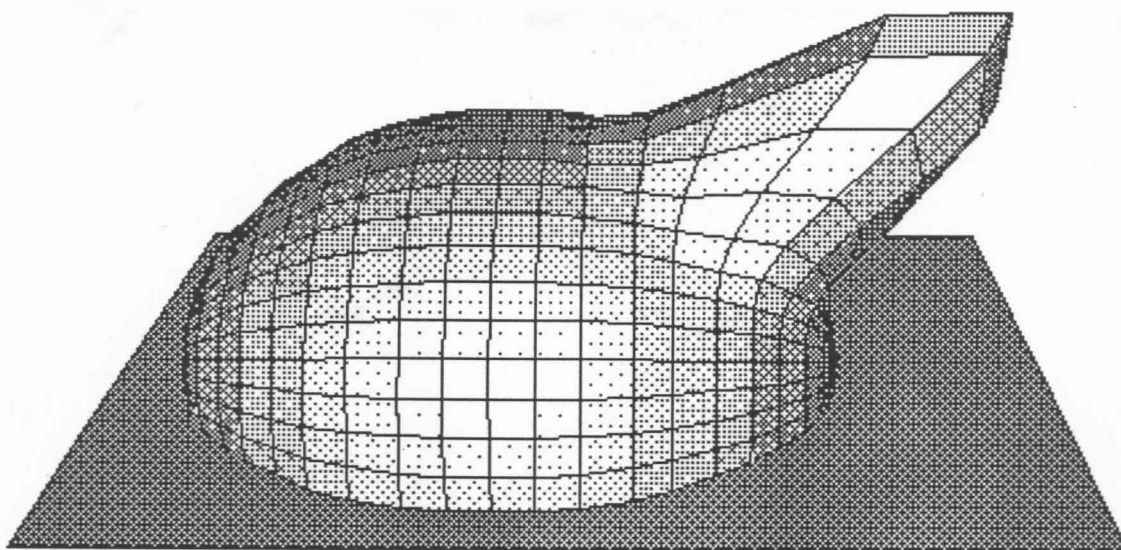
$$I = k_a \cdot I_a + I_p \cdot \left[k_d \cdot (\hat{n}^T \cdot \hat{L}) + \frac{D \cdot G \cdot F}{\hat{n}^T \cdot \hat{S}} \right]$$

nebo v případě více zdrojů světla

$$I = k_a \cdot I_a + \sum_{j=1}^q I_{p_j} \cdot \left[k_d \cdot (\hat{n}^T \cdot \hat{L}_j) + \frac{D_j \cdot G_j \cdot F_j}{\hat{n}^T \cdot \hat{S}} \right]$$

Pro získání vjemu odpovídajícího realitě je možné opět vypočtenou intenzitu zeslabit v závislosti na vzdálenosti pozorovatele od plochy stejným způsobem jako v předchozím textu.

Je zřejmé, že uvedené vztahy jsou poměrně hodně výpočtově náročné. I přes tuto nevýhodu jsou uvedené metody používané v komerčně dostupných programových celcích.

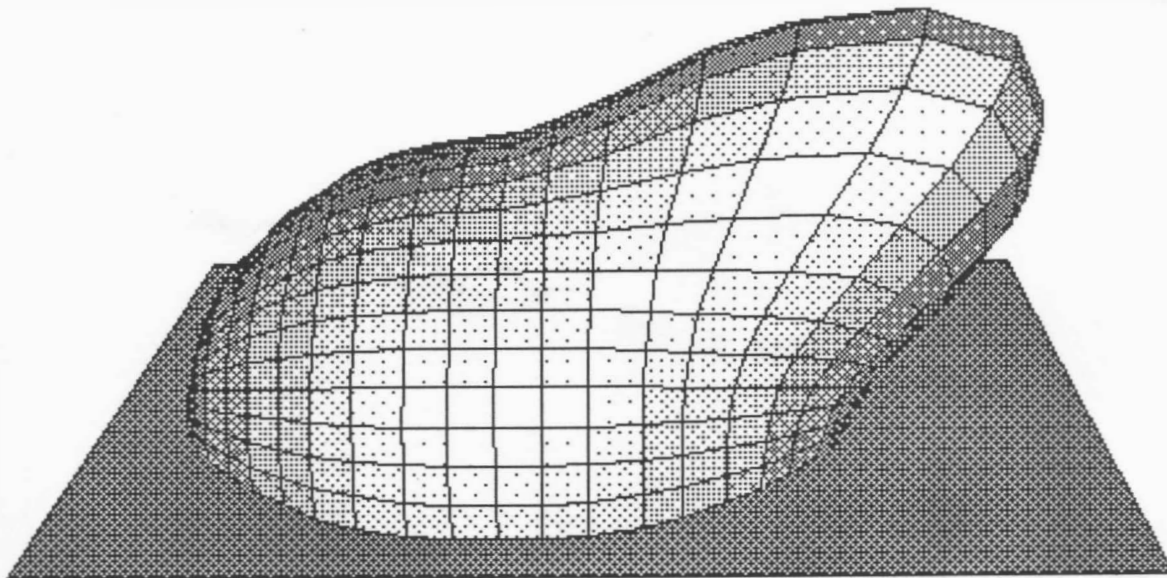


Obr. 3.2.12

Příklad

Zobrazte závislost průběhu intenzit v závislosti na poloze pozorovatele pro různé hodnoty vektoru L . Předpokládejte, že normálový vektor plochy n je konstantní. Závislost vyjádřete ve sférických souřadnicích, kde φ a η jsou definovány vektorem S ,

koeficient n určuje charakteristiku povrchu - viz kap.3.1, a souřadnice r je úměrná vypočtené intenzitě. Uvedená závislost vytvoří obecnou plochu v E_3 , viz obr.3.2.12 - 3.2.13.

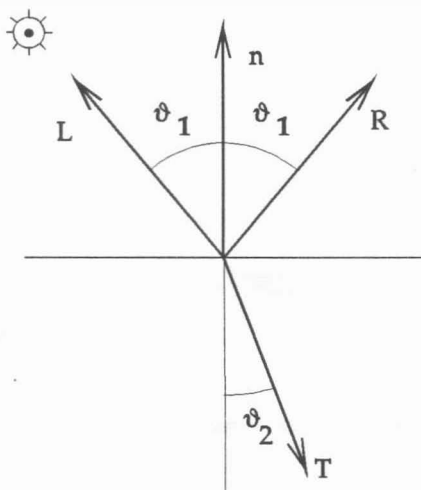


Obr. 3. 2. 13

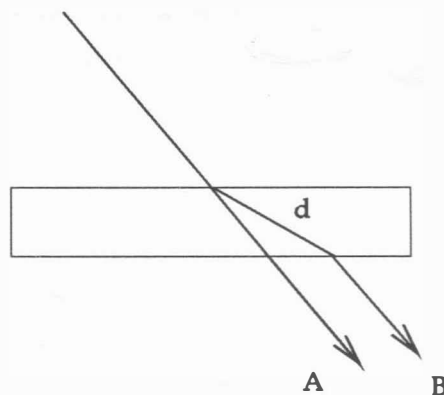
3.3 Lom světla

Uvedené základní iluminační modely je vhodné doplnit o možnost respektovat nejen odrazy vznikající na povrchu, ale i průhlednost či průsvitnost těles, tzn. respektovat lom světla na rozhraní dvou prostředí. Ze Snellova zákona je známo (viz obr.3.3.1), že:

$$\eta_1 \cdot \sin \vartheta_1 = \eta_2 \cdot \sin \vartheta_2$$



Obr. 3. 3. 1



Obr. 3. 3. 2.

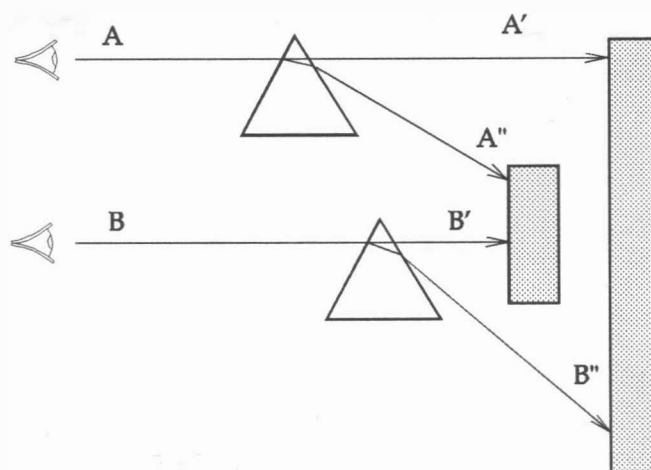
Navíc intenzita propuštěného světla závisí exponenciálně na délce dráhy, kterou paprsek v daném prostředí urazí, tj.

$$I = I_0 \cdot t \cdot e^{-ad}$$

kde a , t jsou koeficienty charakteristické pro daný materiál, d je délka dráhy paprsku v daném prostředí.

Koeficient t určuje množství propuštěného světla, zatímco koeficient a určuje, jak rychle je světlo absorbováno.

Na obr.3.3.2 je ukázáno posunutí paprsku při průchodu jiným prostředím, přičemž A označuje geometrický paprsek nerespektující zákon lomu světla. Paprsek označený B představuje optickou dráhu, po které se paprsek pohybuje vlivem lomu. Je nutné si uvědomit, že při respektování zákonů týkajících se lomu světla může docházet k neočekávatelným efektům, viz obr.3.3.3.



Obr. 3.3.3

Zjednodušené modely pro zobrazování scén s transparentními tělesy mohou být použity pro modifikaci algoritmů řešení viditelnosti, s výjimkou algoritmů založených na myšlence z-bufferu, kde pořadí vykreslování ploch je libovolné. Modifikace vycházejí z následující představy:

Pokud právě zobrazovaná plocha s intenzitou I_1 je transparentní a plocha naposledy zobrazovaná má intenzitu I_2 , pak výsledná intenzita zobrazované plochy je dána jako

$$I = I_2 + (I_1 - I_2) \cdot t \quad t \in \langle 0, 1 \rangle$$

kde pro $t = 0$ je zobrazovaná plocha průhledná,

pro $t = 1$ je zobrazovaná plocha neprůhledná.

Lineární model však nedává pro oblá tělesa očekávané výsledky, což je způsobeno zanedbáním proměnné tloušťky materiálu. Je tedy nezbytné modifikovat uvedený vztah tak, aby výsledná hodnota závisela na normále zobrazovaného povrchu. Vztah, který navrhl Kay [153], [154], [109], vystihuje požadovanou závislost, přičemž

$$t = t_{\min} + (t_{\max} - t_{\min}) \cdot \left[1 - \left(1 - |n_z| \right)^p \right]$$

kde t_{\min} , t_{\max} je minimální a maximální hodnota míry

transparentnosti; $t_{\min}, t_{\max} \in \langle 0, 1 \rangle$

n_z je z-ová složka normálového vektoru \hat{n} v daném bodě

p koeficient, jehož hodnota se obvykle volí v rozsahu 1-3.

Z uvedeného je zřejmé, že hodnota t je závislá na hodnotě normálového vektoru n v bodě, který je právě zobrazován.

I když uvedený nelineární model pro zobrazování transparentních objektů nemůže být přímo použit pro z-buffer algoritmus, může být algoritmus z-bufferu modifikován použitím dodatečných seznamů transparentních ploch tak, aby bylo možné zobrazovat i scény s transparentními plochami. Algoritmus využívá obrazovou paměť (frame-buffer) k zobrazení neprůhledných ploch, z-buffer k reprezentaci největší z-ové souřadnice zobrazovaného pixelu, I-buffer pro uložení transparentních intenzit a T-buffer pro reprezentaci váhových funkcí. Vzhledem k paměťovým nárokům je nutné uvedený algoritmus realizovat tak, aby scéna byla řešena vždy po řádkách, tj. použit řádkový z-buffer. Algoritmus 3.3.1 jednoduše řeší problematiku transparentních ploch, avšak výsledek nebude vždy perfektní.

n : = počet n -úhelníků;

for i : = 1 to n do

 if n -úhelník _{i} je transparentní

 then ulož jej v seznamu transparentních n -úhelníků

 else zpracuj jej pomocí algoritmu z-bufferu;

m : = počet transparentních n -úhelníků;

for i : = 1 to m do

 if $z \leq z_{\text{buffer}}$ then

 begin

 vypočti novou intenzitu jako $I_N = t_0 I_0 + t_c I_c$;

 ulož I_N do I-bufferu;

 ulož t_c do T-bufferu;

 end;

 { tímto způsobem je určena váhová suma intenzit všech }

 { ploch, které jsou před nejbližší neprůhlednou plochou }

urči nyní váhový součet intenzity I_F uložené ve frame bufferu a intenzity I_0 uložené v I-bufferu pomocí rovnice:

$$I = I_F + (I_0 - I_F) \cdot t_0$$

kde I_N je nová intenzita, I_0 je původní intenzita,

I_c je intenzita pro zpracovávaný n -úhelník

t_0 je původní koeficient transparentce z T-bufferu

t_c je koeficient transparentnosti zpracovávaného n -úhelníka

Algoritmus 3.3.1

Z výše uvedeného je zřejmé, že k zobrazování scén s vysokou výslednou věrností obrazu je nutné použít kvalitativně jiných algoritmů.

Obecně lze říci, že iluminační model je dán rovnicí:

$$I = k_a \cdot I_a + k_d \cdot I_d + k_s \cdot I_s + k_t \cdot I_t$$

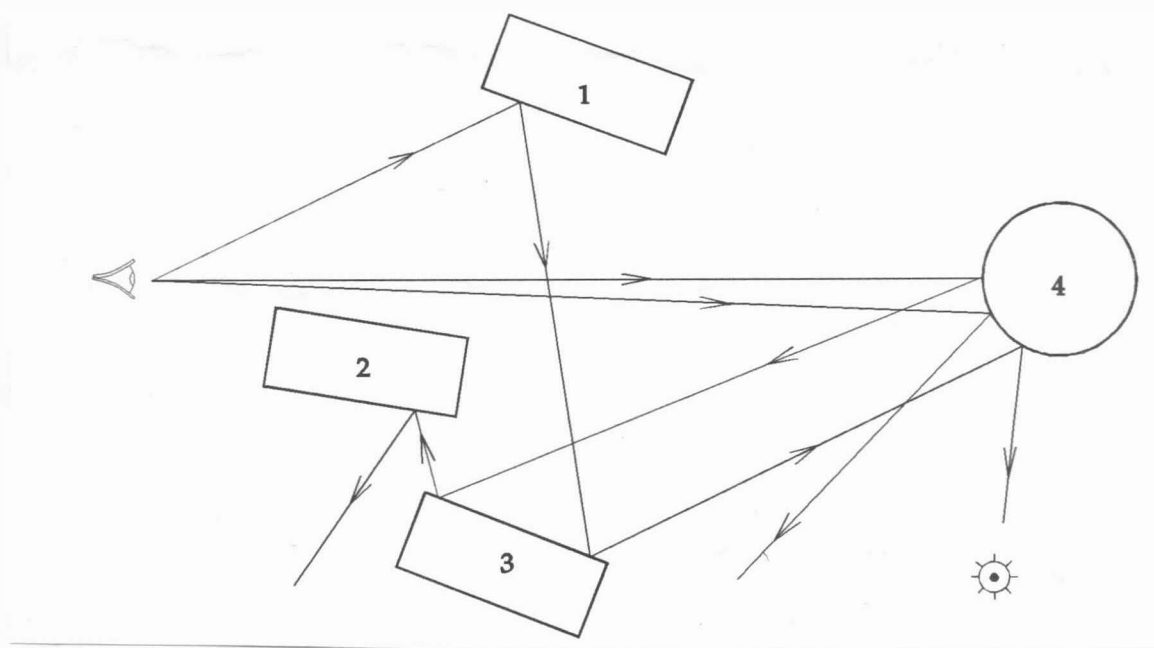
kde člen :

- $k_a \cdot I_a$ určuje vliv okolního světla
- $k_d \cdot I_d$ určuje vliv rozptýleného odrazu
- $k_s \cdot I_s$ určuje vliv zrcadlového odrazu
- $k_t \cdot I_t$ určuje vliv propuštěného světla

Dosud uvedené přístupy k zobrazení dané scény neumožňují však věrné zobrazení dané scény a většina dosud realizovaných algoritmů využívá různých zjednodušení iluminačního modelu. Pro získání vysoce jakostních výstupů je nutné použít dokonalejších technik.

3.4 Globální model osvětlení

Globální model osvětlení vychází z představy, že úkolem je zjistit intenzitu odraženého světla pro každý pixel výsledného obrazu. Dosud uvedené techniky umožňovaly vypočítat intenzitu, která byla určena dopadem světelného paprsku na plochu. Nebylo možné zachytit vliv zvýšení intenzity osvětlení odrazem světelného paprsku od jiné plochy.

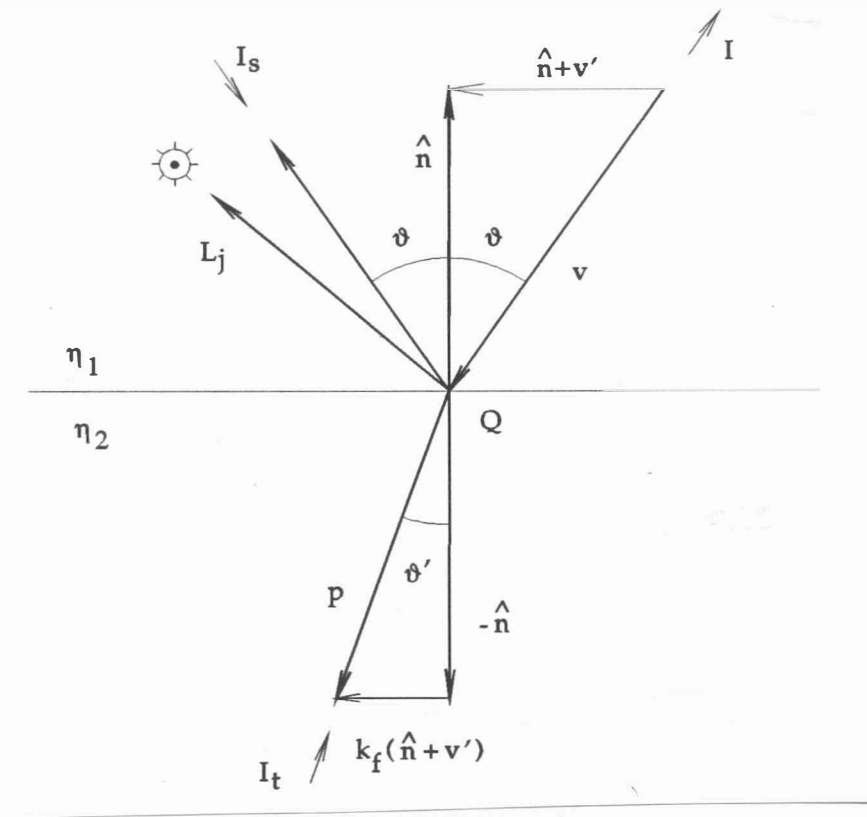


Obr. 3.4.1

Z obr.3.4.1 vyplývá, že pokud odraz tělesa 1 bude mít charakter zrcadlového odrazu, pak pozorovatel uvidí u tělesa 3 i ty plochy, které jsou zakryty nebo od pozorovatele odvráceny. V případě, že těleso 3 odráží světlo alespoň částečně zrcadlovitě, pak pozorovatel uvidí i odvrácenou plochu tělesa 2. Je vhodné poznamenat, že při každém odrazu dochází k vytváření zdánlivého obrazu scény, který je stranově či výškově převrácený. Navíc uživatelův vjem musí být ovlivňován též zdrojem světla, které se odráží od ploch těles 4, 3, 1, i když tento zdroj světla není přímo viditelný z pozice pozorovatele.

Z uvedeného vyplývá, že metodu sledování paprsku lze doplnit vhodným modelem osvětlení tak, aby výsledné zobrazení scény bylo velmi věrné. Kay [153], [154] a Whitted [143] realizovali metodu

sledování paprsku doplněnou globálními metodami určování celkového osvětlení jednotlivých ploch.



Obr. 3.4.2

Paprsek, viz obr.3.4.2, který je určen vektorem \mathbf{v} , dosáhne povrchu v bodě Q , kde se \mathbf{v} obecně částečně odrazí ve směru určeném vektorem \mathbf{r} a částečně je propuštěn ve směru \mathbf{p} . Paprsek o intenzitě I_t přicházející ve směru $-\mathbf{p}$ je propuštěn rozhraním ve směru $-\mathbf{v}$. Analogicky pro paprsek o intenzitě I_s přicházející k bodu Q ve směru $-\mathbf{r}$ je odražen do směru určeného vektorem $-\mathbf{v}$. Pak intenzita, kterou pozorovatel vnímá, je dána vztahem, viz [143]:

$$I = k_a \cdot I_a + \sum_{i=1}^m I_{p_i} \cdot \left[k_d \cdot (\hat{\mathbf{L}}_i^T \cdot \hat{\mathbf{n}}) + k_s \cdot (\hat{\mathbf{R}}_i^T \cdot \hat{\mathbf{S}}_i)^n \right] + k_s \cdot I_s + k_t \cdot I_t$$

kde k_a , k_d , k_s , k_t jsou v tomto obecném případě závislé na vlnové délce a úhlu mezi normálou a směrovým vektorem paprsku.

Nicméně Whitted [143] pokládá tyto koeficienty za konstantní a člen $k_s I_s$ za nezávislý na úhlu dopadu.

Ze zákonů optiky vyplývá, že odražený paprsek r , resp. propuštěný paprsek p , musí ležet ve stejné rovině jako paprsek v a normála n . Podle obr.3.4.2 platí, že

$$r = v' + 2 \cdot \hat{n} \cdot \hat{n} \cdot v' \quad p = k_f \cdot (\hat{n} + v') - \hat{n}$$

kde $v' = \frac{v}{v \cdot \hat{n}}$

$$k_f = \left(k_\eta^2 \cdot |v'|^2 - |v' + \hat{n}|^2 \right)^{-1/2}$$

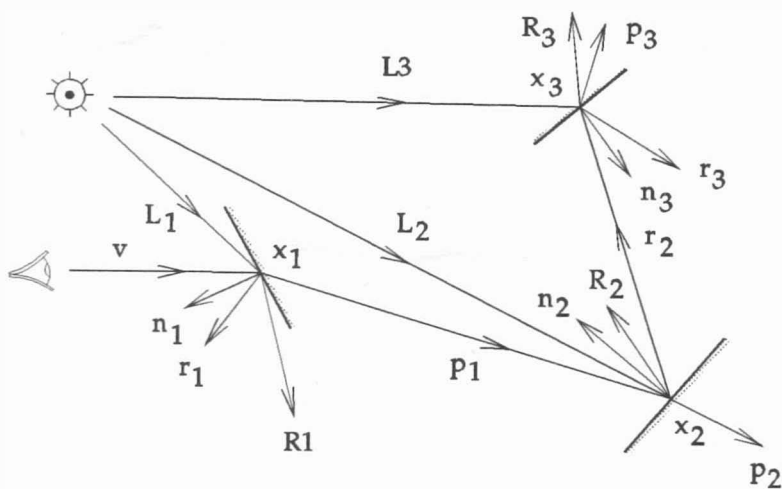
$$k_\eta = \frac{\eta_1}{\eta_2}$$

přičemž k_η je podíl indexů lomu jednotlivých materiálů

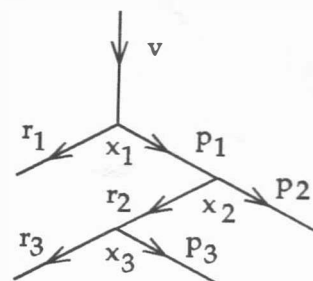
\hat{n} je normálový vektor ve směru přicházejícího paprsku.

Na rozdíl od metody sledování paprsku pro scény s neprůsvitnými plochami není proces ukončen při nalezení nejbližšího průsečíku, neboť paprsek se vlastně na rozhraní dvou prostředí rozštěpí obecně na paprsek odražený a na paprsek propuštěný do druhého prostředí, viz obr.3.4.2.

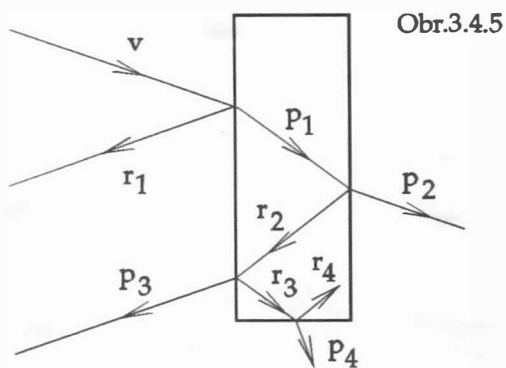
V případě, že $k_f^2 < 0$, pak jde o celkový interní odraz a předpokládá se, že $I_t = 0$.



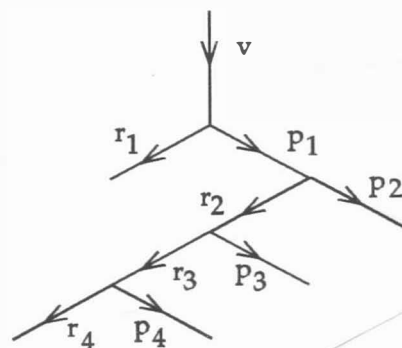
Obr. 3.4.3



Obr. 3.4.4



Obr. 3. 4. 5



Obr. 3. 4. 6

Uvedený proces končí v okamžiku, kdy paprsek opouští scénu. Určení intenzity v místě průsečíku paprsku s povrchem vyžaduje vyhodnocení celého stromu. Intenzita v odpovídajícím uzlu stromu je zmenšena úměrně dráze, kterou paprsek prošel, tj. vzdálenosti vstupního bodu a průsečíku paprsku s další stěnou. Po vyhodnocení celého stromu je odpovídající pixel aktivován s odpovídající intenzitou. Teoreticky je strom nekonečný, avšak při realizaci je výpočet ukončen, pokud paprsek opustí scénu nebo pokud intenzita klesne pod určitou mez, nebo pokud je dosažena povolená hloubka rekurze.

Obr. 3.4.5 a 3.4.6 ukazují případ vnitřní reflexe pro uzavřený průsvitný objekt. V případě, že při návratu paprsku z tělesa dochází na rozhraní k zrcadlovému odrazu, je paprsek "uvězněn" a absorbován. Nemůže tedy přispět k intenzitě vnímané uživatelem. V obecných případech je generován paprsek p , který "opouští" těleso a který může ovlivnit intenzitu, která je pozorovatelem vnímána.

Uvedený postup umožňuje zobrazování scén s poměrně vysokou věrností, lze však vytvořit scény, které vypadají nepřírodně. Vysokého stupně věrnosti, zejména pokud se týče barevných scén, je možné docílit až použitím globálních iluminačních technik založených na výpočtu vyzařované energie.

Příklad

Určete intenzitu vnímanou pozorovatelem, je-li dána jednoduchá scéna s plochami kolmými na rovinu xz , viz obr. 3.4.3. Pozorovatel je v nekonečnu ve směru osy z , přičemž $x = 5$. Uvažme dále jeden bodový zdroj světla v poloze $x = 3$, $y = 0$, $z = 10$ a necht' plochy jsou definovány rovnicemi, viz [109]:

$$1: \quad x + z - 12,5 = 0 \quad 4 \leq x \leq 6$$

$$2: \quad x - z - 2 = 0 \quad 4 \leq x \leq 6$$

$$3: \quad x - 3z + 9 = 0 \quad 1 \leq x \leq 3$$

Jejich iluminační charakteristiky jsou určeny jednotlivými koeficienty:

$$1: \quad k_{a_1} = 0,15 \quad k_{d_1} = 0,15 \quad k_{s_1} = 0,8 \quad k_{t_1} = 0,5 \quad k_{\eta_1} = 1/1,1$$

$$2: \quad k_{a_2} = 0,15 \quad k_{d_2} = 0,15 \quad k_{s_2} = 0,8 \quad k_{t_2} = 0,5 \quad k_{\eta_2} = 1,1$$

$$3: \quad k_{a_3} = 0,15 \quad k_{d_3} = 0,15 \quad k_{s_3} = 0,8 \quad k_{t_3} = 0,0 \quad k_{\eta_3} = 1,1$$

Intenzita okolního světla $I_a = 1,0$ a intenzita zdroje světla $I_p = 10$, přičemž koeficient n pro Phongovo stínování je pro všechny plochy roven 50.

Paprsek je orientován od pozorovatele směrem ke scéně a je určen rovnicemi

$$x = 5 \quad y = 0$$

přičemž výsledný strom znázorňující dělení paprsku je na obr. 3.4.6.

Průsečík paprsku s plochou 1 je určen řešením soustavy rovnic, která je dána rovnicemi roviny a paprsku

$$x + z - 12,5 = 0$$

$$x = 5 \quad a \quad y = 0$$

a tedy

$$z = 7,5$$

Prvním průsečíkem s plochou je bod $\mathbf{x}_1 = [5, 0, 7,5]^T$, který tvoří první uzel stromu. Normovaný normálový vektor $\hat{\mathbf{n}}_1$ plochy 1 je:

$$\hat{\mathbf{n}}_1 = \frac{1}{\sqrt{2}} [1, 0, 1]^T$$

Z polohy pozorovatele lze určit pohledový vektor \mathbf{v}_1 :

$$\mathbf{v}_1 = [0 , 0 , -1]^T$$

a tedy:

$$\mathbf{v}'_1 = \frac{\mathbf{v}_1}{\mathbf{v}_1^T \cdot \hat{\mathbf{n}}_1} = -\sqrt{2} [0 , 0 , 1]^T$$

Směrový vektor odraženého paprsku je

$$\begin{aligned} \mathbf{r}_1 &= \mathbf{v}'_1 + 2 \hat{\mathbf{n}}_1 = -\sqrt{2} [0 , 0 , 1]^T + \frac{2}{\sqrt{2}} [1 , 0 , 1]^T \\ &= \sqrt{2} [1 , 0 , 0]^T \end{aligned}$$

příčemž:

$$\hat{\mathbf{n}}_1 + \mathbf{v}'_1 = \frac{1}{\sqrt{2}} [1 , 0 , -1]^T$$

Pak

$$\begin{aligned} k_{f_1} &= \left(k_{\eta}^2 \cdot |\mathbf{v}'_1|^2 - |\mathbf{v}'_1 + \hat{\mathbf{n}}_1|^2 \right)^{-1/2} = \left[\left(\frac{1}{1,1} \right)^2 \cdot 2 - 1 \right]^{-1/2} \\ &= 1,238 \end{aligned}$$

$$\begin{aligned} \mathbf{p}_1 &= k_{f_1} \cdot (\hat{\mathbf{n}}_1 + \mathbf{v}'_1) - \hat{\mathbf{n}}_1 \\ &= \frac{1}{\sqrt{2}} \cdot \left[1,238 \left([1 , 0 , -1]^T \right) - [1 , 0 , 1]^T \right] \\ &= [0,168 , 0 , -1,582]^T \end{aligned}$$

Odražený paprsek \mathbf{r}_1 opouští scénu a nebude proto dále uvažován. (V uvedeném příkladě toto vyplývá z obrázku, jinak by bylo nutné hledat průsečík s plochami!)

Nyní je možné zapsat rovnice propuštěného paprsku \mathbf{p}_1 v parametrickém tvaru, tj. :

$$x = 5 + 0,168 \cdot t \quad y = 0 \quad z = 7,5 - 1,582 \cdot t$$

Průsečík \mathbf{x}_2 paprsku \mathbf{p}_1 s druhou rovinou je určen dosazením těchto rovnic do rovnice 2. plochy, tj.

$$\begin{aligned} x - z - 2 &= 5 + 0,168 \cdot t - 7,5 + 1,582 \cdot t - 2 \\ &= 1,75 \cdot t - 4,5 = 0 \end{aligned}$$

a tedy

$$t = 2,571$$

Dosazením hodnoty t do parametrických rovnic pro paprsek p_1 .

$$x_2 = 5 + 0,168 \cdot (2,571) = 5,432$$

$$y_2 = 0$$

$$z_2 = 7,5 - 1,582 \cdot (2,571) = 3,433$$

Vzdálenost mezi průsečíky x_1 , x_2 je dána:

$$d_{12} = |x_2 - x_1| = \sqrt{(5,432 - 5)^2 + (3,433 - 7,5)^2} = 4,09$$

Paprsek p_1 je nyní vzat jako nový paprsek určený:

- směrovým vektorem $v_2 = p_1$

- bodem x_2 .

Pak

$$x_2 = [5,432, 0, 3,433]^T$$

$$v_2 = p_1 = [0,168, 0, -1,582]^T$$

Normalizovaný vektor \hat{n}_2 druhé plochy (pozor na její orientaci!) je určen:

$$\hat{n}_2 = \frac{1}{\sqrt{2}} [-1, 0, 1]^T$$

Pak

$$r_2 = [-1,278, 0, 0,136]^T$$

$$p_2 = [0,215, 0, -1,999]^T$$

Paprsek p_2 opouští scénu a nebude tedy dále uvažován.

Pro paprsek r_2 lze psát:

$$x(t) = x_2 + r_2 \cdot t$$

tj.

$$x = 5,432 - 1,278 \cdot t \quad y = 0 \quad z = 3,433 + 0,136 \cdot t$$

Dosazením do rovnice pro třetí plochu dostáváme :

$$\begin{aligned} x - 3z + 9 &= 5,432 - 1,278 \cdot t - 3(3,433 + 0,136 \cdot t) + 9 \\ &= -1,686 \cdot t + 4,133 = 0 \end{aligned}$$

Řešením dostáváme

$$t = 2,451$$

Dosazením do parametrické rovnice pro paprsek r_2

$$x_3 = 5,432 - 1,278 \cdot 2,451 = 2,299$$

$$y_3 = 0$$

$$z_3 = 3,433 - 0,136 \cdot 2,451 = 3,766$$

Vzdálenost mezi dvěma průsečíky x_2 , x_3 je určena

$$d_{23} = |x_3 - x_2| = \sqrt{(x_3 - x_2)^2 + (z_3 - z_2)^2} = 3,151$$

Nyní je paprsek r_2 vzat opět jako nový paprsek určený

- bodem $x_3 = [2,999, 0, 0,136]^T$

- vektorem $v_3 = r_2 = [-1,278, 0, 0,136]^T$

Nyní lze opět určit

$$\hat{n}_3 = \frac{1}{\sqrt{10}} [1, 0, -3]^T$$

$$p_3 = [-1,713, 0, 0,483]^T$$

$$r_3 = [-1,765, 0, -1,643]^T$$

Vzhledem k tomu, že oba paprsky opouštějí scénu, je výpočet ukončen. Navíc $k_{t_3} = 0$, a tedy se negeneruje žádný paprsek vzniklý průchodem plochou, neboť je neprůsvitná. Z obrázku vyplývá, že světelný zdroj je plochou zakryt, a tedy 3. plocha se nachází ve stínu. Tudíž bod x_3 obdrží pouze jas okolí, tj.

$$I_3 = k_a \cdot I_a = 0,15 \cdot 1 = 0,15$$

Uvedená intenzita je přenášena podél vektoru $-r_2$ na povrch 2. plochy, přičemž jas je zeslaben vlivem vzdálenosti, viz [143], na hodnotu:

$$I_{s_2} = \frac{I_3}{d_{23}} = \frac{0,15}{3,151} = 0,0476$$

Z polohy průsečíku na druhé ploše vyplývá, že nic nezakrývá světelný zdroj (spojnice bodu se světelným zdrojem neprotíná jinou plochu).

Směrový vektor L_2 paprsku přicházejícího od zdroje světla k ploše 2 je určen

$$L_2 = \mathbf{x}_p - \mathbf{x}_2 = [\mathbf{x}_p - \mathbf{x}_2 , 0 , z_p - z_2]^T$$

kde \mathbf{x}_p je poloha zdroje světla I_p

\mathbf{x}_2 je průsečík paprsku s druhou plochou

Dosazením

$$L_2 = [-2,432 , 0 , 6,567]^T$$

a normalizací dostáváme

$$\hat{L}_2 = [-0,347 , 0 , 0,938]^T$$

Následně lze určit

$$\begin{aligned} \hat{L}_2^T \cdot \hat{n}_2 &= [-0,347 , 0 , 0,938]^T \cdot \frac{1}{\sqrt{2}} [-1 , 0 , 1]^T \\ &= 0,909 \end{aligned}$$

Paprsek dopadající na plochu 2 ve směru L_2 se odrazí ve směru R_2 , přičemž:

$$\hat{R}_2 = [-0,938 , 0 , 0,347]^T$$

Směrový vektor pozorovatele je

$$\hat{S}_2 = -\hat{p}_1$$

a tedy

$$\begin{aligned} \hat{R}_2^T \cdot \hat{S}_2 &= -[-0,938 , 0 , 0,347]^T \cdot [0,168 , 0 , -1,582]^T \\ &= -0,707 \end{aligned}$$

Pak

$$\begin{aligned} I_2 &= k_{a_2} \cdot I_a + k_{d_2} \cdot I_p \cdot (\hat{L}_2^T \cdot \hat{n}_2) + k_{s_2} \cdot I_p \cdot (\hat{R}_2^T \cdot \hat{S}_2)^n + k_{s_2} \cdot I_{s_2} + k_{t_2} \cdot I_{t_2} \\ &= 0,15 \cdot 1 + 0,15 \cdot 10 \cdot 0,909 + 0,8 \cdot 10 \cdot 0 + 0,8 \cdot 0,0476 \\ &\quad + 0,5 \cdot 0 = 1,552 \end{aligned}$$

Tato intenzita je přenášena podél vektoru $-\hat{p}_1$ na první povrch a je vlivem vzdálenosti zmenšena na hodnotu:

$$I_{t_1} = \frac{I_2}{d_{12}} = \frac{1,552}{4,09} = 0,379$$

Průsečík x_1 není zastíněn žádnou plochou.

Analogicky L_2 a R_2 lze určit L_1 a R_1 :

$$L_1 = x_p - x_1 = [-2, 0, 2,5]^T$$

a po normalizaci

$$\hat{L}_1 = [-0,625, 0, 0,781]^T$$

Následně

$$\begin{aligned} \hat{L}_1^T \cdot \hat{n}_1 &= [-0,625, 0, 0,781]^T \cdot \frac{1}{\sqrt{2}} [1, 0, 1]^T \\ &= 0,110 \end{aligned}$$

Odražený paprsek R_1 je určen vektorem:

$$\hat{R}_1 = [0,781, 0, -0,625]^T$$

Nyní pohledový vektor $\hat{S}_1 = -v_1$, a tedy

$$\hat{R}_1^T \cdot \hat{S}_1 = -0,625$$

Tudíž

$$\begin{aligned} I_1 &= k_{a_1} \cdot I_a + k_{d_1} \cdot I_p \cdot (\hat{L}_1^T \cdot \hat{n}_1) + k_{s_1} \cdot I_p \cdot (\hat{R}_1^T \cdot \hat{S}_1)^n + k_{s_1} \cdot I_{s_1} + k_{t_1} \cdot I_{t_1} \\ &= 0,15 \cdot 1 + 0,15 \cdot 10 \cdot 0,11 + 0,8 \cdot 10 \cdot 0 + 0,8 \cdot 0 \\ &\quad + 0,5 \cdot 0,379 = 0,505 \end{aligned}$$

Toto je intenzita, která je vnímána pozorovatelem. Nízká hodnota je způsobena tím, že první povrch je osvětlen "téměř" tečně. Vzhledem k velké hodnotě n nejsou odlesky zobrazovány.

V případě barevného světla či barevných ploch je nutné uvedený výpočet provést pro všechny základní barvy (např. pro RGB). Je zřejmé, že je nutné zadat i iluminační charakteristiky pro příslušné základní barvy.

Celý algoritmus sledování paprsku při použití globálního modelu pro osvětlení je znázorněn algoritmem 3.4.1. Algoritmus využívá zásobník pro ukládání informací např. o odraženém

a propuštěném paprsku. Je tedy nutné mít zásobník tak dlouhý, jak je dlouhá nejdelší větev stromu při sledování paprsku. Ukončení algoritmu pro danou větev stromu zpracovávaného paprsku nastává za těchto podmínek:

- paprsek odražený i propuštěný opouští scénu
- naplnění zásobníku (pro jednoduchost se v algoritmu 3.4.1 používá pouze tato podmínka)
- intenzita, která je určena pouze intenzitou okolního, rozptýleného a odraženého světla v bodě průsečíku, klesne pod zvolenou minimální hodnotu

Princip algoritmu je obdobný algoritmu sledování paprsku, viz [214], pro neprůhledná tělesa s tím, že zásobník obsahuje tyto informace :

číslo paprsku - jednoznačné přiřazení pro každý paprsek

typ paprsku - určení typu paprsku (v, r nebo p)

číslo zdrojového paprsku - číslo paprsku, z něhož vznikl tento paprsek

typ zdrojového paprsku - určení typu zdrojového paprsku (v, r nebo p)

indikátor průsečíku - indikace, zdali byl nalezen průsečík daného paprsku s některou plochou

ukazatel na objekt - udává polohu protnutého tělesa v seznamu popisů objektů

souřadnice průsečíku - souřadnice x, y, z průsečíku pro daný paprsek

směrové úhly - specifikace směru daného paprsku

vzdálenost mezi nalezeným průsečíkem daného paprsku a průsečíkem zdrojového paprsku s některou plochou - odpovídá hodnotě d

intenzita propuštěného světla - I_t

intenzita zrcadlově odraženého světla - I_s

Při inicializaci, když je paprsek ukládán do zásobníku, jsou hodnoty I_s , I_t , d nastaveny na nulu, indikátor průsečíku na false. Algoritmus 3.4.1 končí naplněním zásobníku.


```

{ xmax, ymax - maximální prostor adresace }

for x := 0 to xmax do
  for y := 0 to ymax do
    begin indikátor průsečíku := false;
      typ paprsku:=v;
      číslo paprsku:=0; číslo generace:=0;
      It:=0; Is:=0; d:=0;
      generuj paprsek v pro daný pixel;
      ulož paprsek v do zásobníku;
      while not zásobník = prázdný do
        begin { Vypočti intenzitu }
          SOLVE RAY ( I )
        end;
        Display pixel x,y s intenzitou I
      end;
procedure SOLVE RAY ( var I: real);
{ deklarace lokálních proměnných }
begin Vyber paprsek ze zásobníku;
  if indikátor průsečíku = true then
    begin
      Vypočti intenzitu(I);
      if typ paprsku = v
        then EXIT
        else
          begin
            if typ paprsku = r
              then nastav Is v zásobníku pro zdrojový paprsek
              else nastav It v zásobníku pro zdrojový paprsek;
            Dekrementuj číslo generace;
            EXIT
          end
    end
  end
else
  if not průsečík { není průsečík se žádnou plochou } then
    begin
      if typ paprsku = v then
        begin I:= barva pozadí;
          Vyber paprsek ze zásobníku
        end
    end
  end
end

```

```

else
    if typ paprsku = r
        then Nastav  $I_s$  v zásobníku pro zdrojový paprsek
        else Nastav  $I_t$  v zásobníku pro zdrojový paprsek;
EXIT
end
else
    if číslo generace = Stackmax { zásobník je plný ? } then
        begin
             $I_s := 0$ ;  $I_t := 0$ ;  $d := 1$ ;
            Vypočti intenzitu(I);
            if typ paprsku = r
                then Nastav  $I_s$  v zásobníku pro zdrojový paprsek
                else Nastav  $I_t$  v zásobníku pro zdrojový paprsek;
            EXIT
        end
    else
        begin
            Vypočti vzdálenost d mezi průsečíky paprsku;
            indikátor průsečíku := true;
            Zvyš číslo generace paprsku;
            Vypočti odražený a propuštěný paprsek;
            { propuštěný paprsek je nutné určit pouze }
            { v případě, že se respektuje lom světla }
            if odražený paprsek existuje
                then
                    begin Zvyš číslo generace paprsku;
                        ulož odražený paprsek do zásobníku
                    end;
            if propuštěný paprsek existuje
                then
                    begin Zvyš číslo generace paprsku;
                        ulož propuštěný paprsek do zásobníku
                    end;
            end;
        end;
    EXIT
end;
end;

```

```

procedure Vypočti intenzitu ( var I: real );
{ lokální deklarace }
begin
  I:=0;
  for j := 1 to počet světelných zdrojů do
  begin
    Generuj spojnicí mezi průsečíkem a zdrojem  $I_{p_j}$ 
    if existuje takový průsečík then
      begin
        if je alespoň jeden objekt neprůsvitný
          then goto 9
        else pro každý průsvitný objekt spočti zeslabení
          intenzity zdroje  $I_{p_j}$ 
      end;
      Generuj  $\hat{R}_j$  a  $\hat{S}_j$ ;
      
$$I := I + I_{p_j} \cdot \left[ k_d \cdot (\hat{L}_j^T \cdot \hat{n}_j) + k_s \cdot (\hat{R}_j^T \cdot \hat{S}_j)^n \right]$$

    end;
  9:
     $I := I + k_a \cdot I_a + (k_s \cdot I_s + k_t \cdot I_t) / d;$ 
  end;

```

Algoritmus 3.4.1

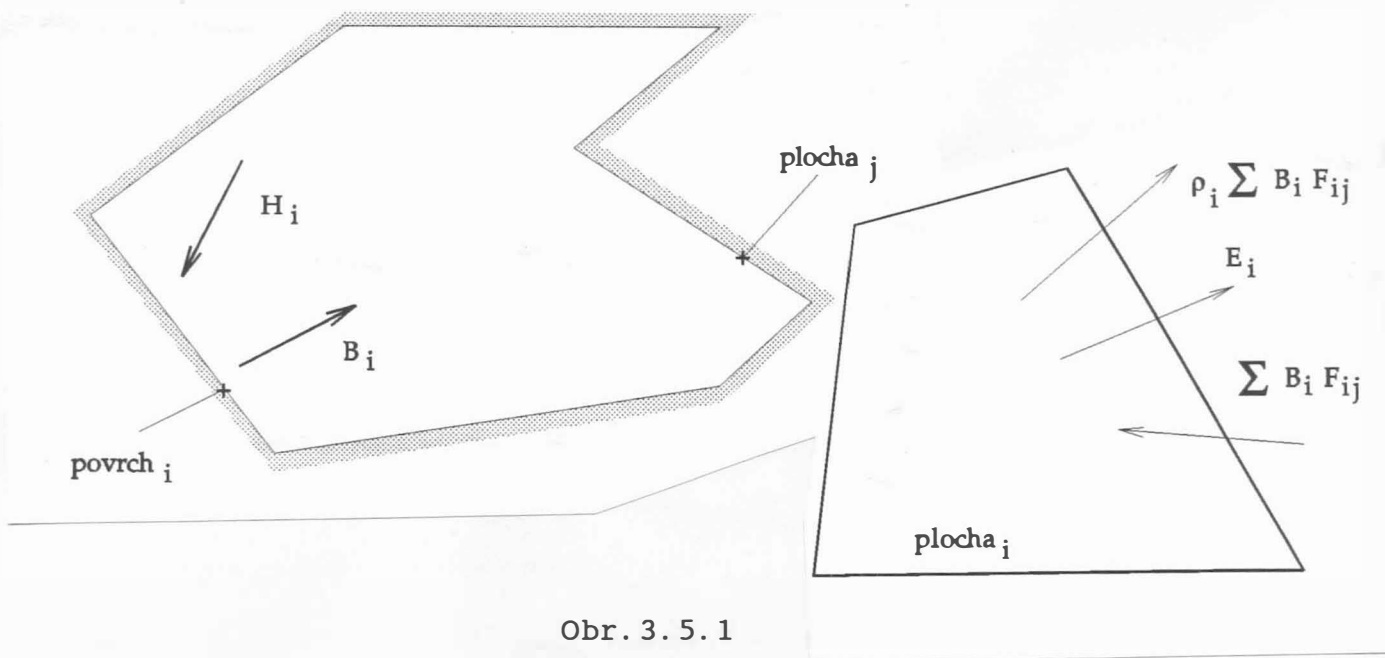
Experiment

Realizujte program, který bude umožňovat zadávání scény z obr.3.3.3. Za základní volte tato tělesa: optický hranol, kvádry obecné velikosti, planární plochy s nesymetrickým vzorem. Po zadání plochy simulujte "procházku" pozorovatele ve směru k optickému hranolu a zobrazujte scénu. Co se stane s částí vzoru, který je "zakryt" hranolem. Jaký bude vjem pozorovatele uvnitř hranolu?

3.5 Model založený na radiačním principu

Radiační model pro výpočet osvětlení zobrazované scény byl navržen v [174] a v současné době je známá celá řada jeho modifikací. Tento model odstraňuje některé nedostatky metody sledování paprsku, která v některých případech nerespektuje fyzikální zákony, ale využívá empirických poznatků a aproximací; to znemožňuje zobrazit řadu efektů, které nastávají zejména při použití barev, např. zbarvení stínů vlivem barvy odraženého světla od jiného tělesa.

Radiační model je založen na energetické rovnováze a byl odvozen z metod používaných při tepelných výpočtech. V určitém smyslu nahrazuje lokální Phongův model přesným globálním modelem, nicméně se předpokládá, že všechny plochy ve scéně mají pouze difúzní odraz. V radiační metodě se předpokládá, že zobrazovaná scéna je obklopena "uzávěrem" nebo jej sama tvoří. Tento uzávěr, který může být i hypotetický, pak zajišťuje, že energie je zachována v daném prostoru. Scéna se sestává ze světelných zdrojů nebo odrážejících ploch. Předpokládá se, že každá plocha může difúzně ideálně odrážet, nebo ideálně difúzně emitovat světlo, nebo obojí.



Obr. 3.5.1

Uvažme případ z obr.3.5.1. Metoda vychází z principu rovnováhy energie uvnitř uzávěru. Radiace (vyzařování) i -té plochy je označena B_i . Ta se sestává z vlastní emise E_i dané plochy (energetický zdroj) a z odražené části přijaté radiace H_i

dopadající na i -tou plochu. Za předpokladu, že uzávěr je konvexní obsahující N ploch, pak jde o příspěvek od všech ostatních ploch, tj. $N-1$ ploch. Poznamenejme, že při použití této metody není zapotřebí pracovat zvlášť s emitovanou energií a zvlášť s odraženou energií, neboť jsou obě difúzního charakteru a jsou pro pozorovatele nerozeznatelné.

Celková radiace (energie na jednotku plochy - $[W/m^2]$) i -té plochy je určena vztahem

$$B_i = E_i + \rho_i \cdot H_i$$

kde: B_i je radiace i -té plochy

E_i je vlastní emise i -té plochy

ρ_i je odrazivost plochy; určuje, jak velká část z energie přijaté i -tou plochou bude emitována zpět

H_i je radiace, která je i -tou plochou přijímána od ostatních ploch

Je zřejmé, že radiace H_i přijatá od ostatních $N-1$ ploch je přijímána jen od těch ploch, které jsou z i -té plochy "viditelné", tj. nejsou odvrácené. Pouze část energie, kterou vyzařuje j -tá plocha, dosáhne i -tou plochu. Tato část je určena konfiguračním faktorem F_{ij} , který reprezentuje vzájemnou polohu i -té a j -té plochy.

Pro energetickou bilanci lze psát (počítá se celková energie, tj. nikoliv radiace)

$$H_i \cdot A_i = \sum_{j=1}^N B_j \cdot A_j \cdot F_{ji} \quad i=1, \dots, N$$

kde A_i , resp A_j je velikost i -té, resp. j -té plochy.

Úpravou pak

$$H_i = \frac{1}{A_i} \sum_{j=1}^N B_j \cdot A_j \cdot F_{ji} \quad i=1, \dots, N$$

Vzhledem k přirozené dualitě konfiguračního faktoru F_{ji} musí platit, že:

$$A_i \cdot F_{ij} = A_j \cdot F_{ji} \quad i, j=1, \dots, N$$

Pak lze

$$H_i = \frac{1}{A_i} \sum_{j=1}^N B_j \cdot A_j \cdot F_{ji} = \sum_{j=1}^N B_j \cdot F_{ij} \quad i=1, \dots, N$$

Emitovaná radiace i -té plochy je pak dána jako

$$B_i = E_i + \rho_i \cdot \sum_{j=1}^N B_j \cdot F_{ij} \quad i=1, \dots, N$$

a tedy

$$B_i - \rho_i \cdot \sum_{j=1}^N B_j \cdot F_{ij} = E_i \quad i=1, \dots, N$$

Uvážíme-li všech N ploch, pak lze soustavu lineárních rovnic přepsat do tvaru

$$\begin{bmatrix} 1 - \rho_1 F_{11} & -\rho_1 F_{12} & \dots & -\rho_1 F_{1N} \\ -\rho_2 F_{21} & 1 - \rho_2 F_{22} & \dots & -\rho_2 F_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ -\rho_N F_{N1} & \dots & \dots & 1 - \rho_N F_{NN} \end{bmatrix} \cdot \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_N \end{bmatrix}$$

kde neznámými je vektor proměnných $\mathbf{B} = [B_1, \dots, B_N]^T$.

Uvedená soustava lineárních rovnic má parametry E_i , F_{ij} a ρ_i , které musí být dány, nebo určeny pro každou plochu. Hodnota E_i je nenulová pro plochy, které přispívají k radiaci uvnitř uzávěru. Jde tedy o plochy, kde dochází k difúznímu odrazu od bodového zdroje světla nebo plošný zdroj záření.

Pokud platí, že

$$E_i = 0 \quad \forall i \in \{1, \dots, N\}$$

pak také platí, že

$$B_i = 0 \quad \forall i \in \{1, \dots, N\}$$

Obecně E_i a ρ_i jsou funkcemi vlnové délky, přičemž je obvykle vybrána určitá hodnota reprezentující "průměrnou" hodnotu pro uvažovaný rozsah vlnových délek, např. pro jednotlivé vlnové délky základních barev RGB. Soustavu rovnic je tedy nutné řešit třikrát, vždy pro vlnovou délku každé základní barvy. Pro řešení soustavy lineárních rovnic může být použita Gauss-Seidlova

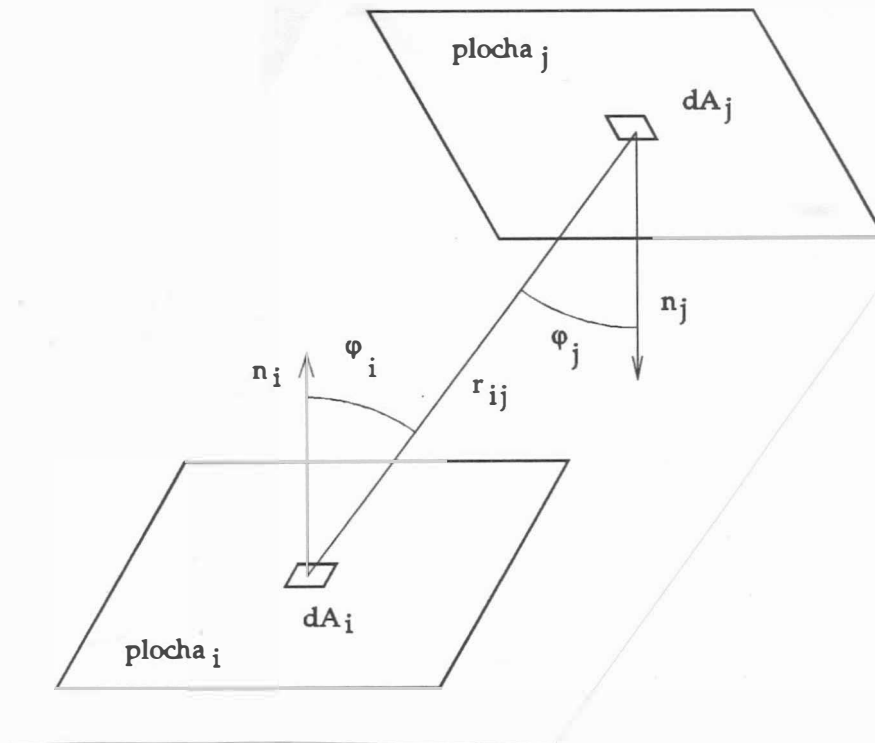
metoda, neboť matice soustavy je diagonálně dominantní a k řešení obvykle stačí pouze 6 až 8 iterací, viz [173].

Nyní je zapotřebí vypočtenou radiaci převést do vrcholů příslušného n -úhelníka tak, aby tato radiace mohla být použita k interpolaci radiace, např. pomocí Gouraudova stínování.

Celý postup generace výstupu pomocí radiční metody může být vyjádřen takto:

- zadání vstupní geometrie a případné rozdělení ploch pomocí záplat
- výpočet konfiguračních faktorů F_{ij}
- výpočet radiace B_i i -té plochy pro každou základní barvu
- extrapolace hodnot B_i do vrcholů n -úhelníka s následným stínováním

Je zřejmé, že konfigurační faktory F_{ij} jsou pro danou scénu konstantní a nemusejí být přepočítávány při změně ρ_i a E_i . Podstatnou výhodou je, že pokud se nezmění geometrie scény, pak pro jiný pohled na scénu se musí znovu provést **pouze** stínování. To je nesmírnou výhodou pro systémy mající VLSI grafické procesory, které přímo podporují řešení viditelnosti a stínování, viz např. grafické pracovní stanice HP 9000 Model 835 Turbo SRX.



Obr. 3. 5. 2

Nyní je nezbytné určit konfigurační faktor F_{ij} , tj. množství energie vyzařované plochou i , které bude příspěvkem k energetické bilanci j -té plochy.

Pro plochy ničím nezakryté platí (v diferenciálním tvaru):

$$F_{dA_i dA_j} = \frac{\cos \varphi_i \cos \varphi_j}{\pi r_{ij}^2}$$

kde r_{ij} je vzdálenost elementárních plošek dA_i a dA_j , přičemž

$$r_{ij} = r_{ji}$$

Integrací přes j -tou plochu dostáváme

$$F_{dA_i A_j} = \int_{A_j} \frac{\cos \varphi_i \cos \varphi_j}{\pi r_{ij}^2} dA_j$$

což je celkový energetický příspěvek od elementární plošky dA_i obdrženy celou plochou A_j . Pak celková energie získaná j -tou plochou od i -té plochy je dána

$$F_{A_i A_j} = \int_{A_i} \int_{A_j} \frac{\cos \varphi_i \cos \varphi_j}{\pi r_{ij}^2} dA_j dA_i$$

Průměrný radiační příspěvek i -té plochy od j -té plochy lze označit jako

$$F_{ij} = \frac{1}{A_i} F_{A_i A_j} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \varphi_i \cos \varphi_j}{\pi r_{ij}^2} dA_j dA_i$$

Ze symetrie vyplývá, že

$$F_{ji} = \frac{1}{A_j} F_{A_j A_i} = \frac{1}{A_j} \int_{A_j} \int_{A_i} \frac{\cos \varphi_j \cos \varphi_i}{\pi r_{ij}^2} dA_i dA_j$$

Ze zákona o zachování energie vyplývá, že

$$\sum_{j=1}^N F_{ij} = 1 \quad i=1, \dots, N$$

Vzhledem k tomu, že plocha "nemůže vidět" sama sebe, platí:

$$F_{ii} = 0 \quad i=1, \dots, N$$

Pro skutečné případy je nutné zvážit situaci, kdy jsou plochy zakryté nějakou jinou plochou. Pak konfigurační faktor je dán jako

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \varphi_i \cos \varphi_j}{\pi r_{ij}^2} (\text{HID}) dA_j dA_i$$

kde **HID** je funkce nabývající pouze hodnot 0 a 1 podle toho, zda elementární ploška dA_j je viditelná z elementární plošky dA_i , tj. není-li nějakou jinou plochou zakrytá.

Z dosud uvedeného vyplývá, že analytický výpočet konfiguračních faktorů F_{ij} je téměř nemožný, neboť jde o **dvojný plošný integrál**. Tento integrál může být sice řešen numericky, avšak za cenu obrovských požadavků na numerické výpočty. V případě, že vzdálenost ploch je velká ve srovnání s rozměry ploch A_i a A_j a pokud nejsou plochy nějakou jinou plochou zakryty, pak lze psát

$$F_{ij} = F_{A_i A_j} \approx \frac{1}{A_i} \int_{A_i} K dA_i$$

kde

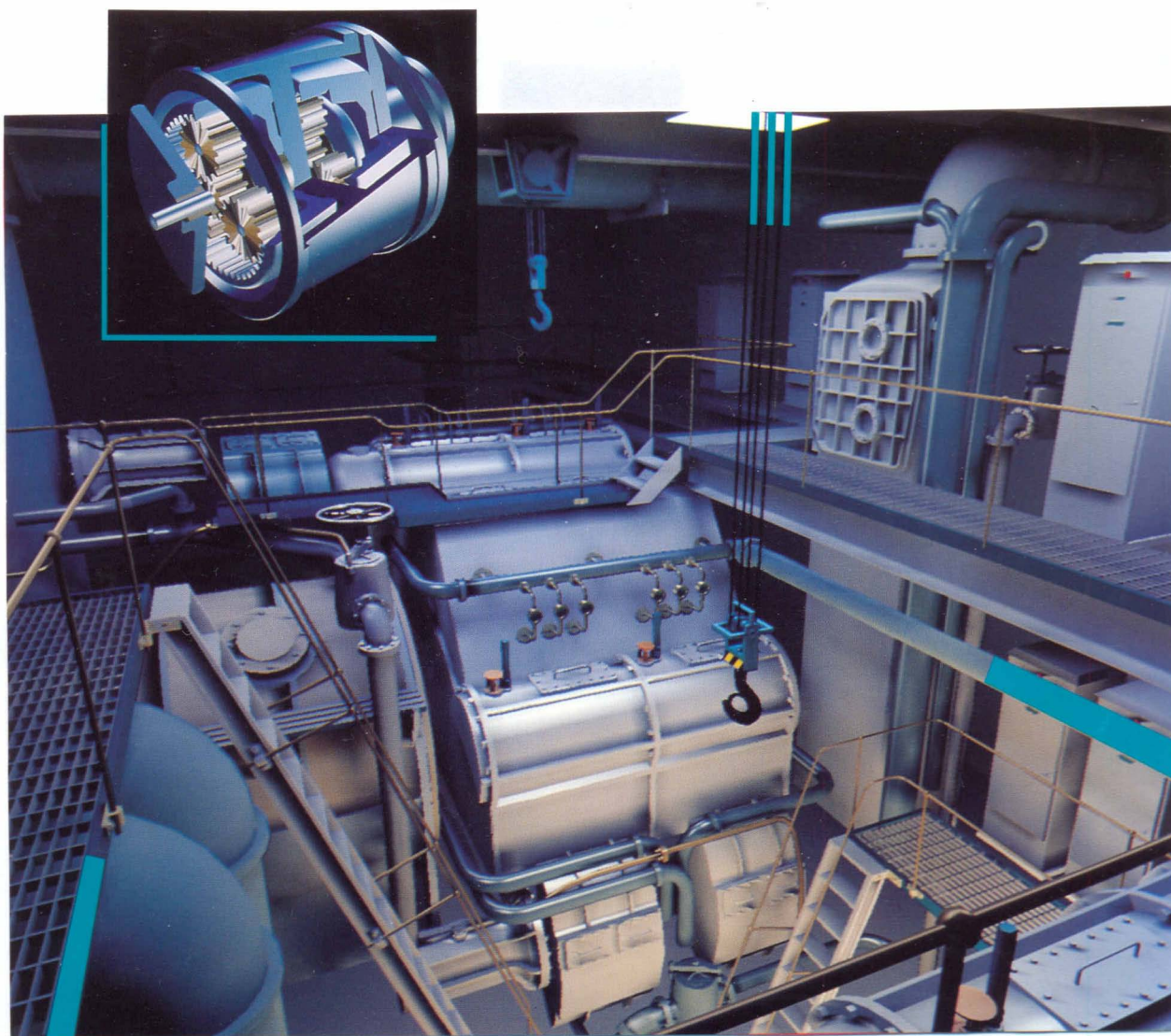
$$K \approx \int_{A_j} \frac{\cos \varphi_i \cos \varphi_j}{\pi r_{ij}^2} dA_j \approx F_{dA_i A_j}$$

neboť K se v těchto případech prakticky nemění.

V případě, že plochy jsou blízko sebe nebo jsou částečně zakryty, je nutné dané plochy A_i a A_j dělit na menší tak dlouho, až bude $F_{dA_i A_j}$ dobře aproximováno hodnotou K (např. hodnota K se nebude dělením měnit).

Technika radiačních výpočtů může být při určitých předpokladech dále zjednodušena, přičemž je nutné brát v úvahu, že řešení soustavy lineárních rovnic je složitosti $O(N^2)$ a dělení ploch na menší přináší též složitost $O(N^2)$.

Další podrobnosti radiačních výpočtů lze nalézt např. v [172], [174], [175], přičemž grafické pracovní stanice Turbo SRX firmy Hewlett Packard podporují jak techniku radiačních výpočtů, viz obr.3.5.3 a obr.3.5.4, tak i sledování paprsku pomocí VLSI grafického procesoru.



Obr. 3. 5. 3



Obr. 3. 5. 4

5. Literatura

- [1] Akimoto, T., Mase, K., Hashimoto, A., Suenaga, Y.: Pixel Selected Ray Tracing, in [64], 1989, pp.29-50.
- [2] Acland, B., West, N.: Real Time Animation on a Frame Store Display System, Computer Graphics (SIGGRAPH'80), Vol.14, 1980, pp.182-188.
- [3] Ackland, B., West, N.: The Edge Flag Algorithm - A Method for Raster Scan Displays, IEEE Trans. on Computers, Vol. C30, 1981, pp.41-48.
- [4] ACM Computer Science Conference, Proceedings of the 1984, Philadelphia, ACM 1984.
- [5] Angell, I.O.: A Practical Introduction to Computer Graphics, MacMillan Press, 1985.
- [6] Alvisi, L., Casciola, G.: Two and Four Array Mask Algorithms in Practice, TR Dept. of Mathematics, Univ. of Bologna, 1988.
- [7] Alvisi, L., Casciola, G.: TAM rivisitato: un metodo rapido ed astto per la rappresentazione prospettica di superfice, PIXEL No.10, 1988, pp.15-24.
- [8] Baker, M.P., Hearn, D.: Computer Graphics, Prentice Hall, International Edition, 1986.
- [9] Barrett, R.C., Jordan, B.W.: A Cell Organized Raster Display for Line Drawings, CACM, Vol.17, 1974, pp.70-77.
- [10] Bartsch, H.J.: Matematické vzorce, SNTL, 1971.
- [11] Berger, M.: Computer Graphics with Pascal, Benjamin Cummings Publishing Comp., Menlo Park, 1986.
- [12] Bergeron, R.D. (Ed.): SIGGRAPH'82 Conference Proceedings, ACM SIGGRAPH, Vol.16, No.3, July 1982.
- [14] Birren, F.: Creative Color, Van Nostrand Reinhold Co., New York, 1961.
- [15] Blinn, J.F.: Models of Light Reflection for Computer Synthesized Pictures, Computer Graphics (SIGGRAPH'77), Vol.11, 1977, pp.191-198.
- [16] Blinn, J.F.: Computer Display of Curved Surfaces, PhD Thesis, Univ. of Utah, 1978.
- [17] Blinn, J.F., Newell, M.E.: Clipping Using Homogeneous Coordinates, Computer Graphics (SIGGRAPH'78), Vol.12, 1978, pp.245-251.

- [18] Bono, P.R., Herman, I. (Ed.): GKS Theory and Practice, EUROGRAPHICS, 1987.
- [19] Bouknight, J.: A Procedure for Generation of Three Dimensional Half-toned Computer Graphics Presentation, CACM, Vol.13, 1970, pp.527-563.
- [20] Butland, J.: Surface Drawing Made Simple, CAD Journal, Vol.11, 1979, pp.19-22.
- [21] Casciola, G.: Basic Concepts to Accelerate Line Algorithms, Computer & Graphics, Vol.12, 1988, pp.489-502.
- [22] Castle, C.M.A., Pitteway, M.L.V.: An Application of Euklid's Algorithm to Drawing Straight Lines, in [39],, 1985, pp.135-139.
- [23] Catmull, E.E.: A Subdivision Algorithm for Computer Display of Curved Surfaces, PhD Thesis, Univ. of Utah, 1974.
- [24] Catmull, E.: A Tutorial on Compensation Tables, SIGGRAPH'79, Computer Graphics, Vol.14, No.3, July 1980, pp.279-285.
- [25] Cheng, F., Yen, Y.: A Parallel Line Clipping Algorithm and Its Implementation, in [41], 1989.
- [26] Clark, J.H.: The Geometry Engine: A VLSI Geometry System for Graphics, Computer Graphics (SIGGRAPH'82), Vol.16, 1982, pp.127-133.
- [27] Claussen, U.: On Reducing the Phong Shading Method, in [64], 1989, pp.333-380.
- [28] Computational Geometry, Proceedings of the Fifth Annual Conference, ACM, 1989.
- [29] Cook, R.L.: A Reflection Model for Realistic Image Synthesis, PhD. Thesis, Cornell Univ., 1982.
- [30] Cyrus, M., Beck, J.: Generalized Two and Three Dimensional Clipping, Computers & Graphics, Vol.3, No.1, 1979, pp.23-28.
- [31] Devillers, O.: The Macro Regions: An Efficient Space Subdivision Structure for Ray Tracing in [64], 1989, pp.27-38.
- [32] Drs, L., Všetečka, J.: Objektivem počítače, SNTL, 1981.
- [33] Duce, D.A., Jancene, P. (Ed.): EUROGRAPHICS'89, Conference Proceedings, North Holland Publ. Comp., 1989.

- [34] Dunlavey, M.R.: Efficient Polygon Filling Algorithms for Raster Displays, Trans. on Graphics, Vol.2., 1983, pp.264-273.
- [35] Ellis, T.M.R., Semenov, O.I. (Ed.): Advances in CAD/CAM, North Holland Publ. Comp., IFIP, 1983.
- [36] Encarnacao, J., Schlechtendahl, E.G.: Computer Aided Design - Fundamentals and System Architectures, Springer Verlag, 1983.
- [37] Enderle, G., Kansy, K., Pfaff, G.: Computer Graphics Programming, Springer Verlag, 1984.
- [38] Enderle, G., Grave, M., Lillenhagen, F. (Ed.): Advances in Computer Graphics I, Springer Verlag, 1986.
- [39] Earnshaw, R.A. (Ed.): Fundamental Algorithms for Computer Graphics, NATO ASI Series, Series F, Vol.17., Springer Verlag, 1985.
- [40] Earnshaw, R.A. (Ed.): Theoretical Foundations of Computer Graphics and CAD, NATO ASI Series, Series F, Vol.40, Springer Verlag, 1987.
- [41] Earnshaw, R.A., Wyvill, B. (Ed.): New Advances in Computer Graphics, Proceedings of Computer Graphics International 89, Springer Verlag, 1989.
- [42] Fitzgerald, W., Gracer, F., Wolfe, R.: GRIN: Interactive Graphics for Modeling Solids, IBM Res.& Devel., Vol.25, No.4., July 1981, pp.281-294.
- [43] Floyd, R., Steinberg, L.: An Adaptive Algorithm for Spatial Gray Scale, SID 1975, Int. Symp. Dig. Techn., 1975, pp.36-37.
- [44] Foley, J.D., van Dam, A.: Fundamentals of Interactive Computer Graphics, Addison-Wesley, 1984.
- [45] Foley, J.D.: Next Generation User Interface Development Tools, in [64], 1989, pp.537-538.
- [46] Franklin, W.R., Lewis, H.R.: 3-D Graphic Display of Discrete Spatial Data by Prism Maps, Computer Graphics (ACM SIGGRAPH'78), Vol.12, No.3, August 1978.
- [47] Franklin, W.R.: An Exact Hidden Sphere Algorithm That Operates in Linear Time, Computer Graphics and Image Processing, Vol.15, 1981, pp.364-379.
- [48] Fuchs, H. (Ed.): SIGGRAPH'81 Conference Proceedings, ACM, SIGGRAPH, Vol.15, No.3, August 1981.

- [49] Gervantz, M., Purgathofer, W.: A Simple Method for Color Quantization: Octree Quantization, Proceedings Computer Graphics International'88, Springer Verlag, 1988, pp.219-231.
- [50] Getto, P.: Fast Ray Tracing of Unevaluated Constructive Solid Geometry Models, in [41], 1989, pp.563-578.
- [51] Ghazanfarpour, D., Peroche, B.: Anti - aliasing by successive Steps with a Z-Buffer, in [64], 1989, pp.235-244.
- [52] Gonzales, R.G., Wintz, P.: Digital Image Processing, Addison Wesley, 1977.
- [53] Gottlieb, M: Hidden Line Subroutines for Three Dimensional Plotting, Byte, Vol.3, No.5, 1978, pp.49-58.
- [54] Gorelik, A.G.: Logical Functions as a Means of Modelling Geometrical Objects, in [35], pp.135-151.
- [55] Granát, L., Sechovský, H.: Počítačová grafika, SNTL, 1980.
- [56] Graphical Kernel System for Three Dimensions (GKS-3D) - Functional Description, Norma ISO/TC97/SC21 IS 7942.
- [57] Greenaway, D. S., Warman, E. A. (Ed.): EUROGRAPHICS'82 Conference Proceedings, North Holland Publ.Comp., 1982.
- [58] Greenberg, D. P., Meyer, G.W.: Perceptual Color Spaces for Computer Graphics, Computer Graphics, Vol.14, 1980, pp.254-261.
- [59] Greenberg, D. P., Marcus, A., Schmidt, A., Gorter, V.: The Computer Image-Applications of Computer Graphics, Addison Wesley, 1982.
- [60] Greenberg, D. P., Meyer, G.W.: Color Education and Color Synthesis in Computer Graphics, Color Research and Application, Vol.11, John Wiley & Sons, Supplement 1986, pp.S39-44.
- [61] Greenberg, D. P.: Advances in Global Illumination Algorithms, in [64], 1989, pp.401-402.
- [62] ten Hagen, P. J.W., Tomiyama, T. (Ed.): Intelligent CAD Systems I, Springer Verlag, 1987.
- [63] Hamlin, G., Gear, C.: Raster Scan Hidden Surface Algorithm Techniques, Computer Graphics (SIGGRAPH'77), Vol.11, 1977, pp.206-213.
- [64] Hansmann, W., Hopgood, F. R. A., Strasser, W. (Ed.): EUROGRAPHICS'89, Conference Proceedings, North Holland Publ. Comp., 1989.

- [65] Haralick, R.M.: Pictorial Data Analysis, NATO ASI, Series F, Vol.4., Springer Verlag, 1983.
- [66] Harrington, S.: Computer Graphics - A Programming Approach, McGraw Hill, 1987.
- [67] Heckbert, P.: Color Image Quantization for Frame Buffer Display, Computer Graphics, Vol.16, No.3, July 1982, pp.297-305.
- [68] Hilbert, R.: Construction and Display of Three Dimensional Polygon Histograms, Computer Graphics, Vol.15, No.2, July 1981.
- [69] Hopgood, F.R.A., Duce, D.A., Gallop, J.R., Sutcliffe, D.C.: Introduction to the Graphical Kernel System (GKS), Academic Press, 1983.
- [70] Hopgood, F.R.A., Hubbolt, R.J., Duce, D.A. (Ed.): Advances in Computer Graphics II, Springer Verlag, 1986.
- [71] Hubbolt, R.J. (Ed.): EUROGRAPHICS'82, Tutorial Notes, EUROGRAPHICS Assos., Geneva, 1982.
- [72] Hubbolt, R.J., Arnold, A.C., Hewitt, W.T.: Interactive Computer Graphics - Course Notes, Univ. of Manchester, Computer Graphics Unit, 1984.
- [73] Inselberg, A.: The Plane with Parallel Coordinates, The Visual Computer, Vol.1, 1985, pp.69-91.
- [74] Inselberg, A., Comut, T., Reif, M.: Convexity Algorithms in Parallel Coordinates, JACM, Vol.34, No.4, October 1987, pp.765-801.
- [75] Inselberg, A., Dimsdale, B.: Parallel Coordinates for Visualizing Multi-Dimensional Geometry, in [84], 1987, pp.25-44.
- [76] Jarvis, J.F., Judice, C.N., Ninke, W.H.: A Survey of Techniques for the Display of Continuous Tone Pictures on Bilevel Displays, Computer Graphics and Image Processing, Vol.5, 1976, pp.13-40.
- [77] Jevans, D.A.J.: Optimistic Multiprocessor Ray Tracing, in [41], 1989, pp.507-522.
- [78] Joseph, H.: Computer Graphics Hardware - Introduction and State of the Art, EUROGRAPHICS'87 Tutorial, EUROGRAPHICS, 1987.

- [79] Kay, D.S.: Transparency, Refraction and Ray Tracing for Computer Synthesized Images, PhD Thesis, Cornell Univ., 1979.
- [80] Kilgour, A.C.: Unifying Vector and Polygon Algorithm for Scan Conversion and Clipping, TR CSC/87/R7, Univ. of Glasgow, May 1987.
- [81] Knuth, D.E.: Digital Halftones by Dot Diffusion, ACM Trans. on Graphics, Vol.6, No.4, October 1987, pp.245-273.
- [82] Kubo, S.: Continuous Color Presentation Using a Low Cost Ink Jet Printers, in [84], 1987, pp.348.
- [83] Kunii, T.L. (Ed.): Frontiers in Computer Graphics, Springer Verlag, 1985.
- [84] Kunii, T.L. (Ed.): Computer Graphics 1987, Proceedings of the 5th International Conference on Computer Graphics, Springer Verlag, 1987.
- [85] Liang, Y.D., Barsky, B.A.: An Analysis and Algorithms for Polygon Clipping, CACM, Vol.26, No.11, 1984, pp.868-876.
- [86] Liang, Y.D., Barsky, B.A.: A New Concept and Method for Line Clipping, ACM Transaction on Graphics, Vol.3, No.1, 1984, pp.1-22.
- [87] Lewell, J.: Computer Graphics - A Survey of Current Techniques and Applications, Orbis Publ. Ltd., London, 1985.
- [88] Mach, K.D., Petty, J.S.: Contouring and Hidden Line Algorithms for Vector Graphic Display, Rep. AFAPL-TR-77-3, 1977.
- [89] Měření barev, ČSN 01 1718.
- [90] Meyer, G.W.: Wavelength Selection for Synthetic Image Generation, Computer Vision, Graphics and Image Processing, Vol.41, 1988, pp.57-79.
- [91] Meyer, G.W.: Tutorial on Color Science, The Visual Computer, Vol.2, Springer Verlag, pp.278-290.
- [92] Murch, G.M.: Human Factors of Color Displays, TR, Tektronix, Oregon, 1989.
- [93] Murch, G.: Color Matching of Display and Printer, in [64], 1989, pp.313-314.
- [94] Newmann, W.M., Sproull, R.F.: Principles of Interactive Computer Graphics, 2nd ed., McGraw Hill, 1981.

- [95] Nicholl, T.M., Lee, D.T., Nicholl, R.A.: An Efficient New Algorithm for 2D Line Clipping: Its Development and Analysis, ACM Computer Graphics, Vol.21, No.4, July 1987, pp.253-262.
- [96] O'Bara, R.M., Abi-Ezzi, S.: An Analysis of Modeling Clip, in [64], 1989, pp.367-380.
- [97] Pavlidis, T.: Graphics and Image Processing, Springer Verlag, 1982.
- [98] Peitgen, H.O.: The Impact of Fractal Geometry for Computer Graphics, in [64], 1989, pp.315-316.
- [99] Perdue, L.: Supercharging Your PC, McGraw Hill, 1987.
- [100] Phillips, R.L. (Ed.): SIGGRAPH'78, Conference Proceedings, ACM SIGGRAPH, Vol.12, No.3, August 1978.
- [101] Pins, M., Hild, H.: Variation on Dither Algorithm, in [64], 1989, pp.381-392.
- [102] Pitteway, M.L.V.: The Algebra of Algorithms - A New Toy for the Theoretician?, IUCC Bulletin, Vol.1, 1979, pp.139-144.
- [103] Pitteway, L.M.V., Watkinson, D.J.: Bresenham's Algorithm with Gray Scale, CACM, Vol.23, 1980, pp.625-626.
- [104] Plastock, R.A., Kaley, G.: Theory and Problems of Computer Graphics, McGraw Hill, New York, 1986.
- [105] Pollack, B.W. (Ed.): SIGGRAPH'79 Conference Proceedings, ACM, SIGGRAPH, Vol.13., No.2., August 1979.
- [106] Popsel, J., Hornung, C.: Highlighting Shading - Lighting and Shading in a PHIGS+/PEX Environment, in [64], 1989, pp.317-332.
- [107] Preparata, F.P., Shamos, M.I.: Computational geometry - An Introduction, Springer Verlag, 1985.
- [108] Rogers, D.F., Adams, J.A.: Mathematical Elements for Computer Graphics, McGraw Hill, New York, 1976, 2. vydání 1990.
- [109] Rogers, D.F.: Procedural Elements for Computer Graphics, McGraw Hill, 1985.
- [110] Rogers, D.F., Earnshaw, R.A. (Ed.): Techniques for Computer Graphics, Springer Verlag, 1987.
- [111] de Ruiter, M.M. (Ed.): Advances in Computer Graphics III, Springer Verlag, 1988.
- [112] Santo, H.P.: Métodos Gráficos e Geometria Computacionais, Dinalivro, Lisboa, 1985.

- [113] Sheppard, J.: Human Color Perception - A Critical Study of the Experimental Foundation, Elsevier, New York, 1968.
- [114] Shirai, Y.: Three Dimensional Computer Vision, Springer Verlag, 1987.
- [115] Skala, V.: An Interesting Modification to the Bresenham Algorithm for Hidden-Line Problem Solution, in [39], 1985, pp.593-602.
- [116] Skala, V.: An Intersecting Modification to the Bresenham Algorithm, Computer Graphics Forum, Vol.6, No.4, 1987, pp.343-347.
- [117] Skala, V.: Algorithms for 2D Line Clipping, in [41], 1989, pp.121-128.
- [118] Skala, V.: Algorithms for 2D Line Clipping, in [64], 1989, pp.355-367.
- [119] Slavkovský, P.: Problém viditelnosti v počítačové grafice, kandidátská disertační práce, MFF UK, Bratislava, 1987.
- [120] Smith, A.R.: Color Gamut Transform Pairs, SIGGRAPH'78 Conference Proceedings, ACM, SIGGRAPH, 1978, pp.12.
- [121] Smith, A.R.: Tint Fill, Computer Graphics (SIGGRAPH'79), Vol.13, 1979, pp.276-283.
- [122] Světelně-technické názvosloví, ČSN 36 0000.
- [123] Staudhammer, J., Livadas, P.E.: Computer Graphics - A Tutorial, The Second International Conf. on Computers and Applications, Beijing, China, 1987.
- [124] Strasser, W. (Ed.): Advances in Computer Graphics Hardware I, Springer Verlag, 1987.
- [125] Sutherland, I.E., Hodgman, G.W.: Reentrant Polygon Clipping, CACM, Vol. 17., No.1, January 1974, pp.32-42.
- [126] Sutherland, I.E., Sproul, R.F., Schumacker, R.A.: A Characterization of Ten Hidden-Surface Algorithms, Computing Surveys, Vol.6, 1974, pp.1-55.
- [127] Tanner, P. (Ed.): SIGGRAPH'83, Conference Proceedings, ACM, SIGGRAPH, Vol.17, No.3, July 1983.
- [128] Teunissen, W.J.M.: HIRASP - A Hierarchical Modelling System for Raster Graphics, PhD Thesis, 1988.
- [129] Thalmann, N.M., Thalmann, D. (Ed.): Computer Generated Images, Proceedings of Graphics Interface'85, Springer Verlag, 1985.

- [130] Thomas, J.J. (Ed.): SIGGRAPH'80, Conference Proceedings, ACM, SIGGRAPH'80, Vol.14,, No.3, July 1980.
- [131] Toifl, J.: Grafické vstupní zařízení počítače, Výběr informací, č.2, SNTL, 1973.
- [132] Toifl, J.: Grafické výstupní zařízení počítače, Výběr informací, č.4, SNTL, 1973.
- [133] UNIRAS - Firemní materiály firmy European Software Contractors, 1985.
- [134] UNIRAS - Universal Raster Report, Firemní materiály, 1985.
- [135] Vandoni, C.E. (Ed.): EUROGRAPHICS'85, Conference Proceedings, North Holland Publ. Comp., 1985.
- [136] Vít a kol.: Televizní technika, SNTL, Praha, 1979.
- [137] Warnock, J.E.: A Hidden Line Algorithm for Halftone Picture Representation, Univ. of Utah, Comp.Sci.Dept., Report TR 4-5, May 1968.
- [138] Warnock, J.E.: A Hidden Surface Algorithm for Computer Generated Halftone Pictures, Univ. of Utah, Comp.Sci.Dept., TR 4-15, June 1969.
- [139] Watkins, G.S.: A Real Time Visible Surface Program, Univ. of Utah, Comp.Sci.Dept., Report UTEC-CSC-70-101, June 1970.
- [140] Watkins, S.L.: Masked Three Dimensional Plot Program with Rotation, Algorithm 483, CACM, Vol.17, 1974, pp.520-523.
- [141] Watters, G., Willis, P.: Scan Converting Extruded Lines at Ultra High Definition, Computer Graphics Forum, Vol.6, No.2, May 1987, pp.133-140.
- [142] Weiler, K., Atherton, P.: Hidden Surface Removal Using Polygon Area Sorting, Computer Graphics (SIGGRAPH'77), Vol.11, 1977, pp.214-222.
- [143] Whitted, T.: An Improved Illumination Model for Shaded Display, CACM, Vol. 23, 1980, pp.343-349.
- [144] Williams, H.: Hidden-Line Plotting Program, Algorithm 420, CACM, Vol.15, 1972, pp.100-103.
- [145] Wright, T.J.: A Two-Space Solution to the Hidden Line Problem for Plotting Functions of Two Variables, IEEE Trans. on Computers, Vol.C-22, 1973, pp.28-33.
- [146] Wyvill, G., Sharp, P.: Fast Antialiasing of Ray Traced Images, in [41], 1989, pp.579-590.

- [147] Xu, H., Peng, Q. S., Liang, Y. D.: Accelerated Radiosity Method for Computer Environment, in [64], 1989, pp.51-62.
- [148] Zhang, J.: A Fast Hidden-Line Removal Algorithm, in [41], 1898, pp.591-602.
- [149] Blinn, J. F., Carpenter, L. C., Lane, J. M., Whitted, T.: Scan Line Methods for Displaying Parametrically Defined Surfaces, CACM, Vol.23, pp.23-34, 1980.
- [150] Phong, B. T.: Illumination for Computer Generated Images, PhD Thesis, Univ. of Utah, 1973.
- [151] Carpenter, L. C., Lane, J. M.: A Generalized Scan Line Algorithm for the Computer Display of Parametrically Defined Surfaces, Computer Graphics and Image Processing, Vol.11, pp.290-297, 1979.
- [152] Gouraud, H.: Computer Display of Curved Surfaces, PhD Thesis, Univ. of Utah, 1971.
- [153] Kay, Douglas Scott: Transparency, Refraction and Ray Tracing for Computer Synthesized Images, MSc Thesis, Cornell Univ., 1979.
- [154] Kay, Douglas Scott, Greenberg, D.: Transparency for Computer Synthesized Images, Computer Graphics (SIGGRAPH'79 Proceedings), Vol.13, pp. 158-164, 1979.
- [155] Beckmann, P., Spizzichiono, A.: Scattering of Electromagnetic Waves from Rough Surfaces, MacMillan Press, New York, 1963, pp.1-33, 70-98.
- [156] Cook, R. L., Torrance, K. E.: A Reflectance Model for Computer Graphics, ACM on Graphics, Vol.1., pp.7-24, 1982.
- [157] Torrance, K. E., Sparrow, E. M.: Polarization, directional distribution, and off-specular peak phenomena in light reflected from roughened surfaces, Journal of the Optical Society of America, Vol.57, 7 (July 1966), 916-925.
- [158] Torrance, K. E., Sparrow, E. M.: Theory for off-specular reflection from roughened surfaces, Journal of the Optical Society of America, Vol.57, 9 (Sept. 1967), 1105-1114.
- [159] Trowbridge, T. S., Reitz, K. P.: Average irregularity representation of roughened surface for ray reflection, Journal of the Optical Society of America, Vol.65, 5 (May 1975), 531-536.
- [160] Agoston, G. A.: Color Theory and Its Application in Art and Design, Springer Verlag 1987

- [161] Baldwin, L.: Color Consideration, BYTE September 1984, pp. 227-246
- [162] Cahill, B.: Drawing on the 8514/A, BYTE March 1990, pp. 279-289
- [163] Drs, L.: Plochy ve výpočetní technice, Matematický seminář SNTL, SNTL 1984
- [164] Skala, V.: Filling and Hatching Operations for Non-Convex Areas with Conic Edges for the Raster Environment, ACM-SIGGRAPH Workshop Lisboa Local Group, Lisboa, Portugal, 1988
- [165] Samet, H.: The Quadtree and Related Hierarchical Data Structures, Computing Surveys, Vol. 16, No. 2, June 1984, pp. 187-260
- [166] Samet, H., Webber, R.E.: Storing a Collection of Polygons Using Quadtrees, ACM Trans. on Graphics, Vol. 4, No. 3, July 1985, pp. 182-222
- [167] Kessener, L.R.A., Peters, F.J., van Lierop, M.L.P. (Ed.): Data Structures for Raster Graphics, Proceedings of a Workshop held at Steensel, Springer Verlag, 1986
- [168] Baumgart, B.G.: A Polyhedron Representation for Computer Vision, Proc. Nat. Comp. Conf., AFIPS 1975, pp. 589-596
- [169] Kilgour, A.: Techniques for Modelling and Displaying 3D Scenes, Technical Report, Univ. of Glasgow, 1986
- [170] Pratt, M.J.: Types of Modeller, Technical Report, Dept. of Mathematics, Cranfield Inst. of Technology, Cranfield U.K., September 1982
- [171] Thalmann, N.M., Thalmann, D. (Ed.): Computer Animation, Theory and Practice, Springer Verlag, 1985.
- [172] Greenberg, D.P.: Ray Tracing and Radiosity, State of Art in Image Synthesis, course notes, SIGGRAPH'86, ACM, 1986
- [173] Greenberg, D.P., Cohen, M.F., Torrance, K.E.: Radiosity: A Methods for Computing Global Illumination, The Visual Computer, Vol. 2., No. 5., September 1986.
- [173] Burgoon, D.A.: Global Illumination Modeling Using Radiosity, Hewlett Packard Journal, December 1989, pp. 78-88.
- [174] Goral, C.M., Torrance, K.E., Greenberg, D.P., Battaile, B: Modelling the Interaction of Light between Diffuse Surfaces, SIGGRAPH'84, ACM, 1984.

- [175] Cohen, M. F., Greenberg, D. P.: The Hemi-Cube: Radiosity Solution for Complex Environments, SIGGRAPH'85, ACM, 1985.
- [176] Graphics Databook, firemní materiály INMOS SGS-Thompson, 1990.
- [177] Systems Solutions, firemní materiály IChips Europe, 1990.
- [178] Jarvis, J. F., Roberts, C. S.: A New Technique for Displaying Continuous Tone Images on a Bilivel Display, IEEE Trans. Communic., Vol. 24, pp. 891-898, 1976.
- [179] Abhyankar, S. S., Chandrasekar, S., Chandru, V.: Improper Intersection of Algebraic Curves, ACM Trans. on Graphics, vol. 9, No. 2, 1990, pp. 147-159.
- [180] Andreev, R. D.: Algorithm for Clipping Arbitrary Polygons, Computer Graphics Forum, Vol. 7, No. 3, 1988, pp. 183-192.
- [181] Brunet, P., Navazo, I.: Solid Representation and Operation Using Extended Octrees, ACM Trans. on Graphics, vol. 9, No. 2, 1990, pp. 170-197.
- [182] Day, A. M.: The Implementation of an Algorithm to find the Convex Hull of a Set of Three Dimensional Points, ACM Trans. on Graphics, vol. 9, No. 1, 1990, pp. 105-132.
- [183] Dobkin, D. P., Levy, S. L. F., Thurston, W. P., Wilks, A. R.: Contour Tracing by Piecewise Linear Approximations, ACM Trans. on Graphics, vol. 9, No. 4, 1990, pp. 389-423.
- [184] Edelsbrunner, H., Mucke, E. P.: Simulation of Simplicity: A Technique to Cope with Degenerate Cases in Geometric Algorithms, ACM Trans. on Graphics, vol. 9, No. 1, 1990, pp. 66-104.
- [185] Falcidieno, B., Floriani, L.: A Hierarchical Boundary Model for Solid Object Representation, ACM Trans. on Graphics, vol. 7, No. 1, 1988, pp. 42-60.
- [186] Fournier, A., Fussel, D.: On the Power of the Frame Buffer, ACM Trans. on Graphics, vol. 7, No. 2, 1988, pp. 103-128.
- [187] Gaude, S., Hobson, R., Chilka, P., Calvert, T.: Multiprocessor Experiments for High Speed Ray Tracing ACM Trans. on Graphics, vol. 7, No. 3, 1988, pp. 151-179.
- [188] Heal, B.: Hidden Octree Removal, Computer Graphics Forum, Vol. 7, No. 3, 1988, pp. 199-206.
- [189] Herman, I., Revczky, J.: Some Remarks on the Modelling Clip Problem, Computer Graphics Forum, Vol. 7, No. 4, 1988, pp. 265-272.

- [190] Herman, I.: On The Projective Invariant of Conics in Computer Graphics, Computer Graphics Forum, Vol.8, No.4, 1990, pp.301-314.
- [191] Hobby, J.D.: Rasterization of Nonparametric Curves, ACM Trans. on Graphics, vol.9, No.3, 1990, pp.3262-277.
- [192] Kuijk, A.A.M., Blake, E.H.: Faster Phong Shading via Angular Interpolation, Computer Graphics Forum, Vol.8, No.4, 1990, pp.315-325.
- [193] Lamming, L., Rhodes, W.L.: A Simple Method for Improved Color Printing of Monitor Images, ACM Trans. on Graphics, vol.9, No.4, 1990, pp.345-375.
- [194] Levoy, M.: Efficient Ray Tracing of Volume Data, ACM Trans. on Graphics, vol.9, No.3, 1990, pp.245-261.
- [195] Nicholl, R.A., Nicholl, T.M.: Performing Geometric Transformations by Program Transformation, ACM Trans. on Graphics, vol.9, No.1, 1990, pp.28-40.
- [196] Preparata, F.P., Vitter, J.S., Yvinec, M.: Computation of the Axial View of a Set of Isothetic Parallelograms, ACM Trans. on Graphics, vol.9, No.3, 1990, pp.278-300.
- [197] Pun, T., Blake, E.: Relationships Between Image Synthesis and Analysis: Towards Unification?, Computer Graphics Forum, Vol.9, No.2, 1990, pp.149-164.
- [198] Rokne, J.G., Wyvill, B., Wu, X.: Fast Line Scan Conversion, ACM Trans. on Graphics, vol.9, No.4, 1990, pp.376-388.
- [199] Rossignac, J., Voelcker, H.B.: Active Zones in CSG for Accelerating Boundary Evaluation, Redundancy Elimination, Interference Detection, and Shading Algorithms, ACM Trans. on Graphics, vol.8, No.1, 1989, pp.51-87.
- [200] Rushmeier, H.E., Torrance, K.E.: Extended the Radiosity Method to Include Specularly Reflecting and Translucent Materials, ACM Trans. on Graphics, vol.9, No.1, 1990, pp.1-17.
- [201] Stone, M.C., Cowan, W.B., Beatty, J.C.: Color Gamut Mapping and the Printing of Digital Color Images, ACM Trans. on Graphics, vol.7, No.4, 1988, pp.249-293.
- [202] Thomas, D., Netravali, A.N., Fox, D.S.: Anti-aliased Ray Tracing with Covers, Computer Graphics Forum, Vol.8, No.4, 1990, pp.315-324.

- [203] Veenstra, J., Ahuja, N.: Line Drawing of Octree Represented Objects, ACM Trans. on Graphics, vol.7, No.1, 1988, pp.61-75.
- [204] Ware, C., Cowan, W.: The RGYB Color Geometry, ACM Trans. on Graphics, vol.9, No.2, 1990, pp.226-232.
- [205] Zyda, M.J.: A decomposable Algorithm for Contour Surface Display Generation, ACM Trans. on Graphics, vol.7, No.2, 1988, pp.129-148.
- [206] Ježek, F.: AutoCAD - učební text, VŠSE Plzeň, 1991.
- [207] Poláček, J., Ježek, F., Kopincová, E.: Počítačová grafika, skripta ČVUT-FS Praha, 1991.
- [208] Ranklin, J.R.: Computer Graphics Software Construction, Advances in Computer Science Series, Prentice Hall, 1989.
- [209] Rushmeier, H.E., Torrance, K.E.: The Zonal Method for Calculating Light Intensities in the Presence of a Participating Medium, Computer Graphics, Vol.21, No.4, July 1987.
- [210] Počítačová grafika - Názvosloví, ČSN 36 9001 část 13.
- [211] Systémy zpracování informací (GKS), ČSN 36 9180.
- [212] Firemní materiály firmy INTERGRAPH, June 1991.
- [213] Skala, V.: Počítačová grafika I, Skripta VŠSE (Západočeská univerzita), Plzeň, 2. vydání, 1991.
- [214] Skala, V.: Počítačová grafika II, Skripta VŠSE (Západočeská univerzita), Plzeň, 2. vydání, 1991.
- [215] Drzaic, P.S.: Nematic Droplet Polymer Films for High Contrast coloured Reflective Displays, Displays Technology and Applications, Butterworth-Heinemann Ltd., Vol.12, No.1, January 1991, pp.2-13.
- [216] Schwarz, M.W., Cowan, W.B., Beatty, J.C.: An Experimental Comparison of RGB, YIQ, LAB, HSV, and Opponent Color Models, ACM Transaction on Computer Graphics, Vol.6, No.2, April 1987, pp.123-158.
- [217] Colour Addendum to ISO 8613 - Working Draft, ISO TC97 SC18 WG5, X3H3/88-47.
- [218] Prett, V.: Cifrovaja obrobotka izobraženij, Mir, Moskva, 1982.
- [219] Serba, I.: Termodynamický přístup k výpočtu osvětlení prostorové scény v počítačové grafice, seminář MOP 91, 1991.

[220] Sochor, J.: Sledování paprsku ve 3D scéně, seminář MOP 91, 1991.

[221] Hall, R.: Illumination and Color in Computer Generated Imaginary, Springer Verlag, 1989.

6. Obsah

1. Úvod	1
2. Barva a barevné systémy	4
2.1 Achromatické světlo	5
2.2 Půltónování	7
Metoda vzorů	8
Metoda konstantního prahu	11
Floyd - Steinbergova metoda	12
Metoda dvourozměrného rozptylu chyb	15
Dithering	17
2.3 Chromatické světlo	23
Systém RGB	24
Systémy XYZ, CIE-xy, CIE-uv	29
Systém CMY	41
Systém YIQ	43
Systémy HLS a HSV	44
Systém HSI	50
Systémy vzorníků, tabulek a barevné atlasy	52
Systém CIE LAB	54
Systém CIE $L^*u^*v^*$	55
Systém Opponent	56
Systém RGYB	58
Systém SOW	59
3. Osvětlení a stínování	66
3.1 Modely osvětlení	66
3.2 Metody stínování	72
Konstantní stínování	72
Gouraudovo stínování	73
Phongovo stínování	75
Model Torrance - Sparrow	83
3.3 Lom světla	90
3.4 Globální model osvětlení	94
3.5 Model založený na radiačním principu	108
4. Závěr	116
5. Literatura	117
6. Obsah	132