


# Farebné modely a práca s nimi

Work with the color models

Ján Chudý

---

Bakalárska práca  
2007

 Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
Ústav aplikované informatiky  
akademický rok: 2006/2007

## **ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Ján CHUDÝ**  
Studijní program: **B 3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**

Téma práce: **Barevné modely a práce s nimi**

Zásady pro vypracování:

1. Vytvořte literární rešerši na zadané téma.
2. Seznamte se s barevnými modely RGB, CMYK, HSV, HSL, LAB, YCBCR, YUV a YIQ. Tyto model charakterizujte a specifikujte oblasti jejich využití.
3. Dle zadané literatury vyhledejte, popř. sestavte algoritmy pro vzájemné převody mezi těmito barevnými modely.
4. Navrhněte a vytvořte aplikaci ve Windows, která bude tyto převody provádět.
5. Vytvořte prezentaci v Powerpointu na téma barevných modelů, která by se dala využít při výuce předmětu Počítačová grafika.

Rozsah práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. Žára, J., a kol.: **Moderní počítačová grafika**. Computer Press, 2005.
2. Liberty, J. **Naučte se C++ za 21 dní**, Computer Press, Praha 2002.
3. <http://cgg.ms.mff.cuni.cz/pepca/> – počítačová grafika na KU
4. <http://en.wikipedia.org> – internetová encyklopedie

Vedoucí bakalářské práce:

**Ing. Pavel Pokorný, Ph.D.**

Ústav aplikované informatiky

Datum zadání bakalářské práce:

**13. února 2007**

Termín odevzdání bakalářské práce:

**24. května 2007**

Ve Zlíně dne 13. února 2007



prof. Ing. Vladimír Vašek, CSc.  
*děkan*



doc. Ing. Ivan Zelinka, Ph.D.  
*ředitel ústavu*

## **ABSTRAKT**

Hlavným cieľom tejto bakalárskej práce je vytvorenie teoretického zhrnutia na tému farebných modelov. Teoretická časť sa zameriava na popis vybraných farebných modelov, popis ich vlastností a spôsobov použitia. Praktická časť práce zahŕňa zostavenie algoritmov pre vzájomné prevody medzi farebnými modelmi, vytvorenie triedy v jazyku C++ ktorá tieto algoritmy využíva. Trieda môže byť použitá v ľubovoľnej aplikácii, kde je potrebné prevádzať farby medzi farebnými modelmi. V záverečnej časti sa práca zameriava na vytvorenie aplikácie pre Windows ktorá využíva spomínanú triedu na vizuálne znázornenie prevodných vzťahov medzi farebnými modelmi. Poslednou náplňou tejto práce je vytvorenie prezentácie, ktorú je možné použiť pri výuke predmetu Počítačová grafika.

Kľúčové slová: farebný model, RGB, CMYK, HSL, HSB, HSV, CIE, konverzia

## **ABSTRACT**

The main aim of the work is creation of theoretical summary in the field of color models. The theoretical part was focused in description of chosen color models, description of their properties and ways of utilization. The practical part contains design of algorithms for mutual conversions among color models and design of a class in the C++ language, which utilizes these algorithms. The class may be used in an arbitrary application where it is necessary to transfer colours among color models. The final part on the work is focused in design of Windows application which uses the class for visualization of transfer relations among color models. The presentation, which can be used for education in the subject Počítačová grafika was also created.

Keywords: color model, RGB, CMYK, HSL, HSB, HSV, CIE, conversion

Na tomto mieste by som rád poďakoval vedúcemu mojej bakalárskej práce, pánovi Ing. Pavlovi Pokornému, Ph.D., za inšpiráciu, cenné rady a čas venovaný mojej práci.

Ďalej by som chcel poďakovať mojej rodine, vďaka ktorej mi bolo umožnené študovať na Univerzite Tomáše Bati ve Zlíně.

Prehlasujem, že som na bakalárskej práci pracoval samostatne a použitú literatúru som citoval. V prípade publikácie výsledkov, ak to bude uvoľnené na základe licenčnej zmluvy, budem uvedený ako spoluautor.

V Zlíne, 23. 5. 2007

.....  
Podpis diplomanta

**OBSAH**

<b>ÚVOD.....</b>	<b>8</b>
<b>I TEORETICKÁ ČASŤ .....</b>	<b>9</b>
<b>1 FAREBNÉ MODELY V POČÍTAČOVEJ GRAFIKE .....</b>	<b>10</b>
1.1 FAREBNÝ MODEL RGB.....	10
1.2 FAREBNÝ MODEL CMY(K) .....	12
1.3 FAREBNÝ MODEL HSV (HSB).....	13
1.4 FAREBNÝ MODEL HSL .....	16
1.5 FAREBNÝ MODEL LAB (ČASTO TIEŽ L*A*B).....	16
1.6 FAREBNÝ PRIESTOR CIE 1931 .....	18
<b>2 FAREBNÉ MODELY PRE TELEVÍZNU A VIDEO TECHNIKU.....</b>	<b>22</b>
<b>II PRAKTICKÁ ČASŤ .....</b>	<b>23</b>
<b>3 KONVERZNÉ ALGORITMY MEDZI FAREBNÝMI MODELMI .....</b>	<b>24</b>
3.1 CMYK.....	24
3.1.1 Konverzia CMYK → RGB .....	24
3.1.2 Konverzia RGB → CMYK .....	24
3.2 HSV.....	24
3.2.1 Konverzia HSV → RGB .....	24
3.2.2 Konverzia RGB → HSV .....	25
3.3 HSL .....	25
3.3.1 Konverzia HSL → RGB .....	25
3.3.2 Konverzia RGB → HSL .....	26
3.4 YUV .....	27
3.4.1 Konverzia YUV → RGB .....	27
3.4.2 Konverzia RGB → YUV .....	27
3.5 YIQ.....	27
3.5.1 Konverzia YIQ → RGB.....	27
3.5.2 Konverzia RGB → YIQ.....	28
3.6 YCbCr .....	28
3.6.1 Konverzia YCbCr → RGB .....	28
3.6.2 Konverzia RGB → YCbCr .....	28
<b>4 TRIEDA CCOLORCONVERSION.....</b>	<b>29</b>
4.1 POPIS TRIEDY .....	29
4.2 PRÁCA S TRIEDOU .....	30
<b>5 APLIKÁCIA COLORTRANSLATOR.....</b>	<b>32</b>
<b>6 PREZENTÁCIA K VÝUKE .....</b>	<b>34</b>

<b>ZÁVER.....</b>	<b>35</b>
<b>ZÁVER V ANGLIČTINE .....</b>	<b>36</b>
<b>ZOZNAM POUŽITEJ LITERATÚRY .....</b>	<b>37</b>
<b>ZOZNAM POUŽITÝCH SYMBOLOV A ZKRATIEK.....</b>	<b>38</b>
<b>ZOZNAM OBRÁZKOV .....</b>	<b>39</b>
<b>ZOZNAM PRÍLOH.....</b>	<b>40</b>

## ÚVOD

Predstavy o tom, aká farba vznikne zmiešaním dvoch rôznych farieb sú u rôznych ľudí pomerne odlišné. Je preto dôležité nájsť prostriedky, pomocou ktorých môžu bežný užívatelia grafických systémov dobre popísať farby a ich zloženie. Súbor základných farieb, pravidlá ich miešania a menené farebné charakteristiky sú v počítačovej grafike definované pomocou farebných modelov.

Práca je členená do dvoch logicky odlišných častí. Prvá obsahuje teoretickú časť, ktorá sa zaoberá popisom vlastností a spôsobov použitia vybraných farebných modelov, ktorými sú farebné modely RGB, CMYK, HSV, HSL, LAB, YCbCr, YUV a YIQ. Druhá časť je venovaná praktickej časti. Tu sú popísané jednotlivé algoritmy pre vzájomné prevody medzi týmito farebnými modelmi, ďalej popis triedy `cColorConversion`, ktorá tieto algoritmy využíva. Trieda je veľmi univerzálna a je možné ju použiť prakticky v ľubovoľnej aplikácii, ktorá potrebuje prevádzať farby medzi farebnými modelmi. V ďalšom bode praktickej časti sa práca venuje popisu aplikácie `ColorTranslator`, ktorá bola vytvorená s úmyslom použitia k výuke predmetu Počítačová grafika. Posledná časť sa zameriava na vytvorenie prezentácie v programe Powerpoint na tému farebných modelov.



## I. TEORETICKÁ ČASŤ

## 1 FAREBNÉ MODELY V POČÍTAČOVEJ GRAFIKE

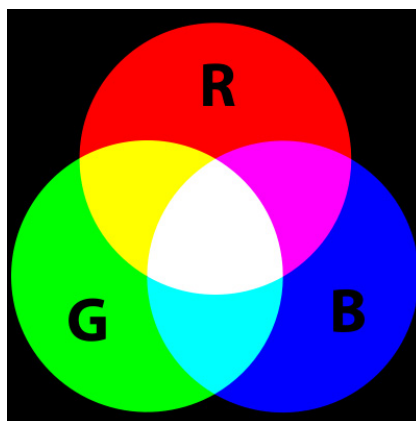
Pri práci s farbami je dôležitá jednak voľba základných farieb, ich spôsob kombinovania(miešania). Predstavy o tom, aká farba vznikne zmiešaním dvoch rôznych farieb sú u rôznych ľudí pomerne odlišné. Je preto dôležité nájsť prostriedky, pomocou ktorých môžu bežný užívatelia grafických systémov dobre popísať farby a ich zloženie.

Súbor základných farieb, pravidiel ich miešania a menené farebné charakteristiky sú v počítačovej grafike definované pomocou farebných modelov. Grafické systémy často obsahujú niekoľko farebných modelov a prostriedky pre prevody medzi nimi.

### 1.1 Farebný model RGB

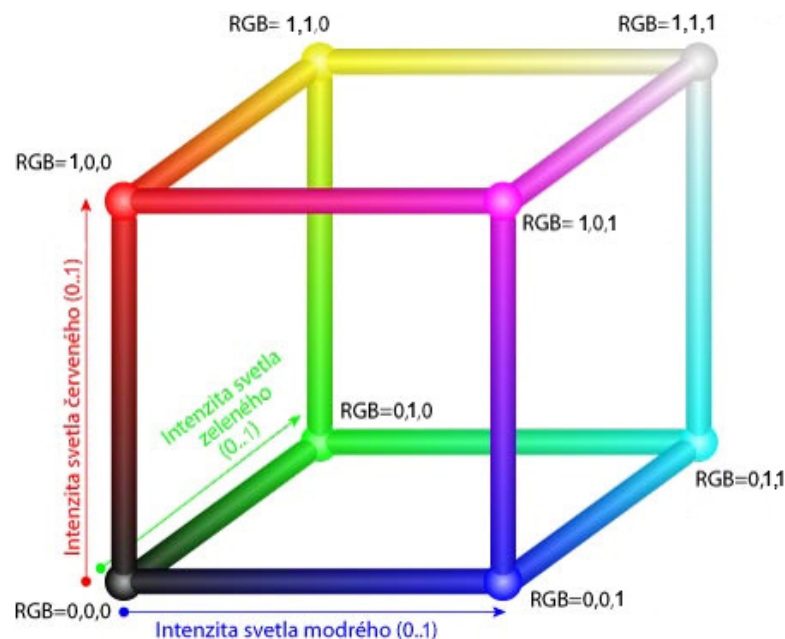
Základné zložky sú: červená (Red), zelená (Green) a modrá (Blue). Pre tieto farby je príznačné práve to, že ľudské oko má najlepšiu citlivosť práve pre ich vlnové dĺžky (630nm, 530nm a 450nm). Intenzita základných farieb sa v tomto modeli pohybuje v intervale  $\langle 0,1 \rangle$ . Pri technickej implementácii je tento rozsah prevádzaný do digitálnej formy. Najčastejšie je kódovaný na 8-bitov (t.j. 256 dielov). [5]

Ak zhrnieme vyššie popísané, RGB model udáva silu pôvodného svetla rozloženého na silu jeho troch farebných kanálov. Rovnaká farba sa dá teda jednoducho obnoviť, pokiaľ zaistíme tri zdroje svetla presných farieb - červenej, zelenej a modrej a ich silu zregulujeme podľa RGB hodnôt (napr. 0 nesvieti, 255 svieti na maximum). Ich vzájomným zložením je obnovená pôvodná farba – RGB je aditívny farebný model. Hodí sa pre zariadenia, ktoré do tmavého podkladu pridávajú svetlo, a tým vytvárajú farby. [1]



Obr. 1. Aditívne miešanie farieb

Farebný model RGB sa svojim rozsahom najčastejšie reprezentuje ako jednotková kocka umiestnená v osách r, g, b. Z toho vyplýva, že množina základných farieb obsahuje osem farieb. Vrchol  $[0,0,0]$  (t.j. stred súradnicového systému) odpovedá čiernej farbe (Black). Naproti tomu vrchol  $[1,1,1]$  odpovedá bielej farbe (White). Farby ležiace na diagonále medzi týmito vrcholmi odpovedajú odtieňom šedej (Gray). [5]



Obr. 2. Reprezentácia farebného modelu RGB

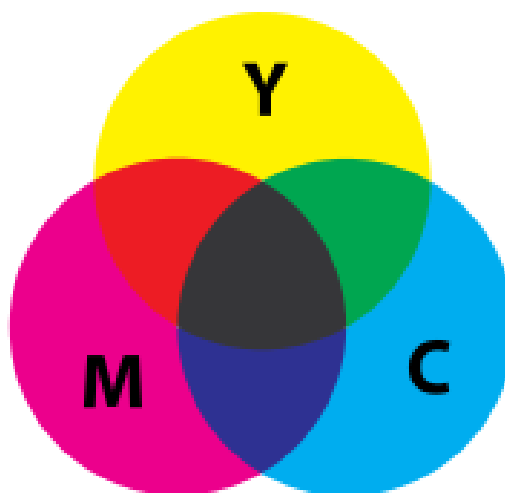
Bohužiaľ sám model RGB nemá žiadnu presnú špecifikáciu svojich základných farieb - červenej, zelenej a modrej - a tak vzniklo viac modelov založených na RGB. Najznámejšia a najrozšírenejšia je asi varianta sRGB, ktorá je štandardom Windows. sRGB má presne definované základné farby RGB, biely bod aj gamma. Farebný model sRGB je praktický hlavne preto, že odpovedá reálnym možnostiam zobrazenia väčšiny monitorov, a používa sa preto masovo aj na Internete.

Iným zástupcom RGB modelu je AdobeRGB model, ktorý bol vytvorený firmou Adobe v roku 1998. Používa mierne odlišné základné farby a vďaka tomu obsiahne väčší rozsah farieb než model sRGB, hlavne vo farbe zeleno-modrej. Jeho nevýhodou ale je, že väčšina bežných monitorov ho nedokáže zobrazit'. [1]

## 1.2 Farebný model CMY(K)

Zatiaľ čo zhasnutý monitor je čierny a farby sa vytvárajú postupným pridávaním farieb RGB (preto je model RGB aditívny a pre monitory priamo určený), tak papier je biely a teda odráža teoreticky všetko svetlo, ktoré na neho dopadá. Všetky možné farby je teda treba dosiahnuť inak - krytím bieleho papiera atramentmi - teda subtraktívnou (odčítacou) metódou. Použitím atramentov s farbami azúrová (Cyan), purpurová (Magenta) a žltá (Yellow) sa podobného efektu dá dosiahnuť.

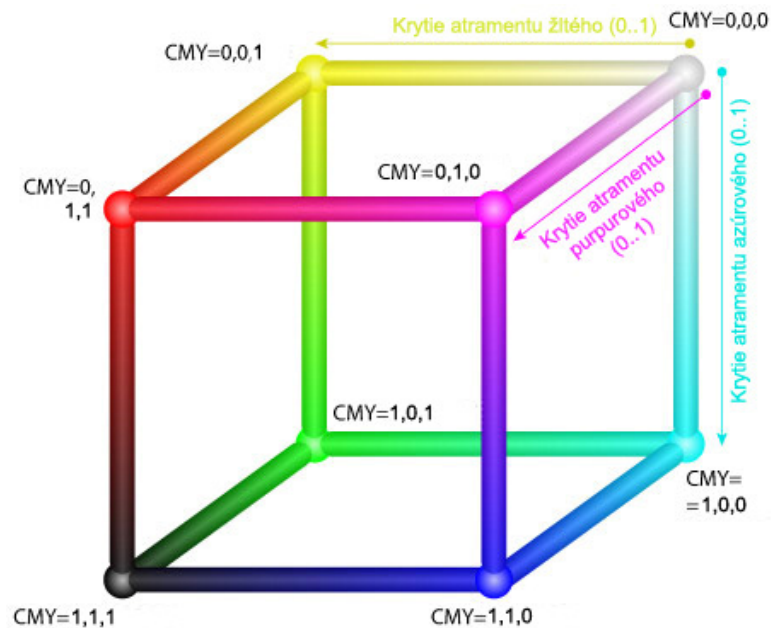
Azúrová je doplnková farba k červenej, a preto bude odrážať všetko svetlo okrem červeného. Podobne purpurová je doplnková k zelenej a žltá k modrej. Pomocou CMY farieb je tak možné riadene "uberať" RGB svetlo, a tak doceliť farieb opačne - uberaním z bielej. A to je princíp modelu CMYK, kde štvrtá čierna farba (black) je pridaná len pre jednoduchšiu realizáciu tmavých farieb. Teoreticky nie je vôbec potrebná, prakticky je ale veľmi ťažké vytvoriť tak ideálne atramenty, aby ich zmiešaním vznikla naozaj čierna, nehovoriac o ekonomickom hľadisku. [1]



Obr. 3. Subtraktívne miešanie farieb

Vďaka podobnosti modelov RGB a CMY môžeme veľmi ľahko prevádzať farby medzi oboma farebnými modelmi. Farbu vyjadrenú v modeli CMY získame odčítaním rovnakej farby vyjadrenej v RGB od jednotkovej matice. Bohužiaľ farebné gamuty oboch farebných modelov nie sú zhodné - preto sa nedajú všetky farby vyjadriteľné v RGB zobraziť v CMY a naopak. [4]

Farebný model CMY sa najčastejšie reprezentuje ako jednotková kocka umiestnená v osách c, m, y. Vrchol  $[0,0,0]$  (t.j. stred súradnicového systému) odpovedá bielej farbe. Naproti tomu vrchol  $[1,1,1]$  odpovedá čiernej farbe. Farby ležiace na diagonále medzi týmito vrcholmi, podobne ako u modelu RGB, odpovedajú odtieňom šedej s narastaním v opačnom smere. [5]



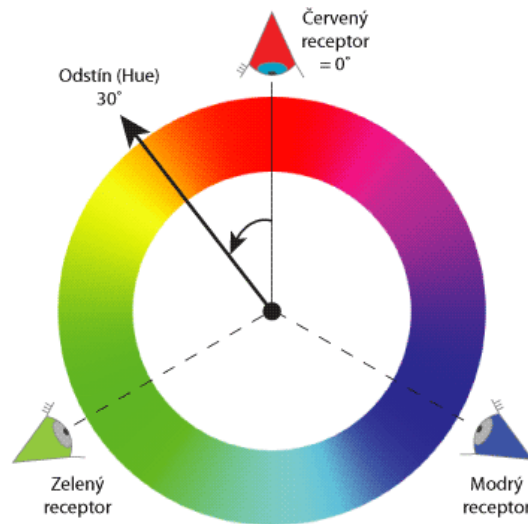
Obr. 4. Reprezentácia farebného modelu CMY

### 1.3 Farebný model HSV (HSB)

Pre niektoré situácie, hlavne pri práci vo fotoeditore, je výhodný model HSV (HSB). Jeho prínos je v tom, že odpovedá ľudskému vnímaniu (popisu) farieb. Zatiaľ čo RGB a CMYK sú modely založené na miešaní farieb, HSV model definuje farby pre človeka prirodzeným spôsobom a teda odpovedá na prirodzené otázky: Aká je to farba? Ako veľa je sýta? Ako veľa je svetlá či tmavá? Je teda veľmi intuitívny a pre niektoré prípady veľmi názorný.

Farebný model HSV používa podobne ako model RGB tiež tri veličiny pre popis farby, dáva im ale iný význam:

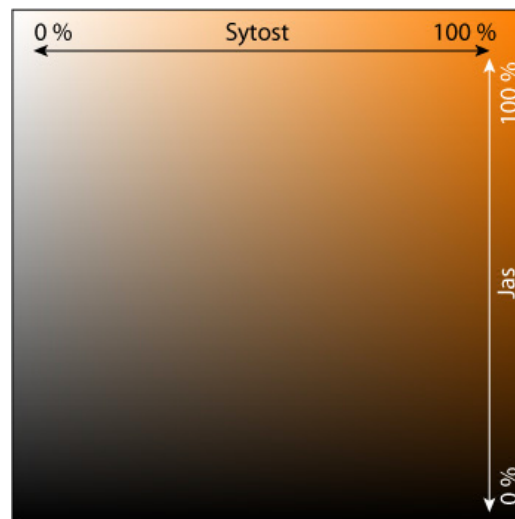
**Odtieň farby** (Hue) popisuje vlastnú čistú farbu (teda napr. červená, zelená, modrá). Asi neprekvapí, že pre popis farby sa používa uhol na farebnom kolese - teda rozsah  $0-360^\circ$ . Dohodou sa za uhol  $0^\circ$  považuje červená farba,  $120^\circ$  odpovedá zelenej a  $240^\circ$  modrej farbe,  $360^\circ$  opäť červenej, pretože sme obišli kruh dookola.



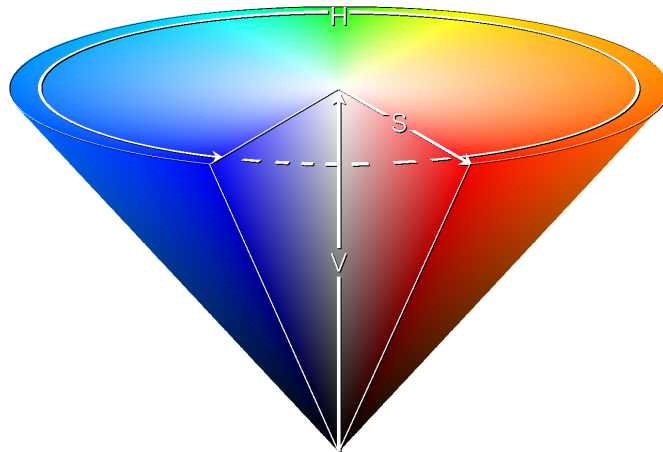
Obr. 5. Výber odtieňa farby (Hue)

**Sýtosť** či **saturácia farby** (Saturation) popisuje, ako veľa je farba "čistá" teda bez primiešania bielej (šedej). Čím viac má v sebe farba bielej, tým viac totiž jej čistota teda sýtosť klesá. Udáva sa v percentách pričom sýtosť 100 % znamená len čistú farbu, sýtosť 50 % znamená polovičnú prímies bielej a sýtosť 0 % potom znamená len odtieň šedej (od bielej po čiernu) teda už úplne bez farby.

**Jas** (Value niekedy tiež Brightness) popisuje jas farby v rozsahu 0-100 %.

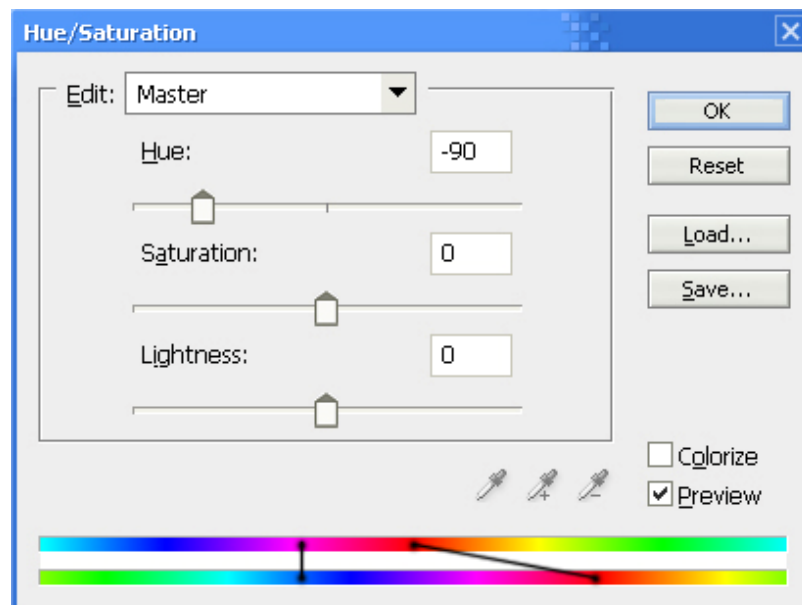


Obr. 6. Výber jasu (Value/Brightness) a sýtosti (Saturation)



Obr. 7. Reprézentácia farebného modelu HSV

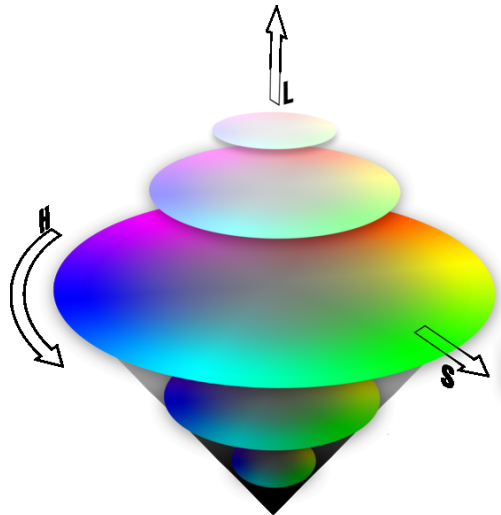
Nie je príliš typické ukladať fotografie v modeli HSV a editory to obvykle ani neponúkajú. Model HSV sa však dobre uplatní pri editácii fotografií prípadne pri grafických návrhoch. Je obvyklé zadávať pomocou modelu HSV farby v grafických editoroch, obrázky pomocou modelu HSV prefarbovať a ovládať sýtosť ich farieb. Nie je prekvapením, že pokiaľ sa sýtosť celého obrázku (teda všetkých bodov fotografie) nastaví na 0 %, obrázok prejde do svojej čiernobielej podoby. [1]



Obr. 8. Dialóg Adobe Photoshopu pre zmenu odtieňa, sýtosti a svetlosti fotografií

## 1.4 Farebný model HSL

Niektoré nedostatky modelu HSV odstránil práve tento model. Základné zložky sú: H (Hue) - farebný tón, S (Saturation) - saturácia a L (Lightness) - svetlosť. Farebný model HSL sa svojim rozsahom reprezentuje ako dvojica kužeľov, ktoré majú spoločnú podstavu. Vrchol jedného kužeľa predstavuje čiernu farbu. Biela farba je naopak vo vrchole druhého kužeľa. Model HSL asi najviac odpovedá skutočnosti, pretože najviac farieb je vnímaných práve pri strednej svetlosti (poloha spoločnej postavy kužeľov, kde  $L=0.5$ ) a vnímavosť klesá tak pri veľkom presvetlení ako aj stmavnutí. Príslušné základné farby (červená, žltá, zelená, tyrkysová, modrá a fialová) ležia opäť na obvode spoločnej podstavy kužeľov. [5]



Obr. 9. Reprezentácia farebného modelu HSL

## 1.5 Farebný model Lab (často tiež $L^*a^*b$ )

Farebný model Lab (úplne presne podľa definície CIE 1976  $L^*a^*b$ ) je model, ktorý bol navrhnutý tak, aby bol úplne nezávislý na zariadení. To ho umožňuje používať ako farebný model referenčný a napríklad Adobe Photoshop používa model Lab ako pomocný pri prevode z jedného farebného režimu do iného. Vedľa toho Photoshop vie v modeli Lab aj pracovať a snímky v modeli Lab aj ukladať.

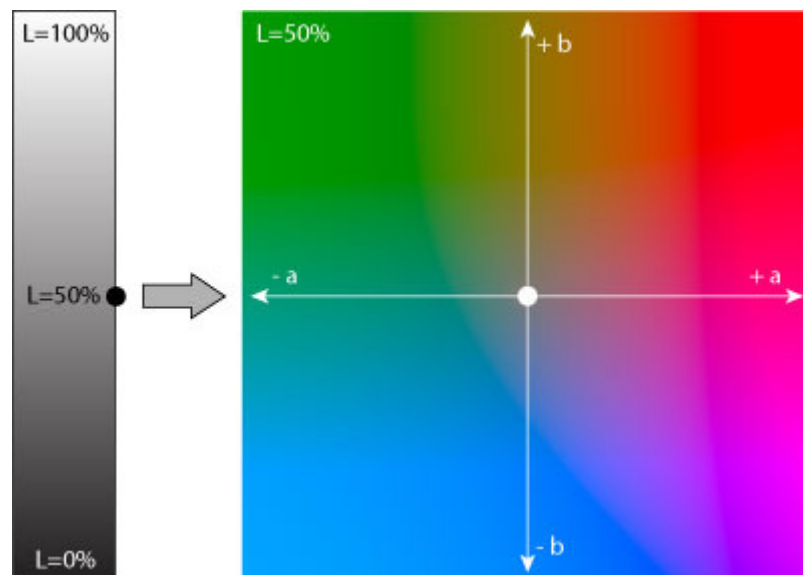


Model Lab používa opäť tri zložky pre popis farby s významom:

**Svetlosť (Lightness, L)**, ktorá v rozsahu 0 až 100 popisuje svetlosť bodu. 0 znamená čierny bod, 100 znamená biely bod.

**Zložka farby a**, ktorá popisuje farbu bodu v smere od zeleno-modrej (záporné hodnoty) po červeno-purpurovú (kladné hodnoty). Napríklad Photoshop umožňuje zadávať hodnoty v rozsahu od -128 do +127.

**Zložka farby b**, ktorá popisuje farbu bodu v smere od modro-purpurovej (záporné hodnoty) po zeleno-žltu-červenú (kladné hodnoty).



Obr. 10. Reprezentácia modelu LAB

Veľkou výhodou modelu Lab je vedľa jeho nezávislosti na zariadení aj skutočnosť, že jeho gamut je najväčší (má teda najširší rozsah zaznamenateľných farieb) a ďalej úplné oddelenie jasovej zložky L od farebných zložiek a, b. To umožňuje realizovať aj niektoré špeciálne efekty pri editácii.

### Praktické využitie modelu Lab

Na zmeny jasů je ľudské oko omnoho citlivejšie než na zmeny farieb. Ak sa prevedie obrázok do modelu Lab, je možné niektoré editácie prevádzať len v jasovom kanále L a naopak niektoré len vo farebných kanáloch a, b. Pomerne známy trik je dostriť len L kanál, čím sa obrázok síce doostří, ale úplne bez súčasného zdôraznenia farebného šumu,

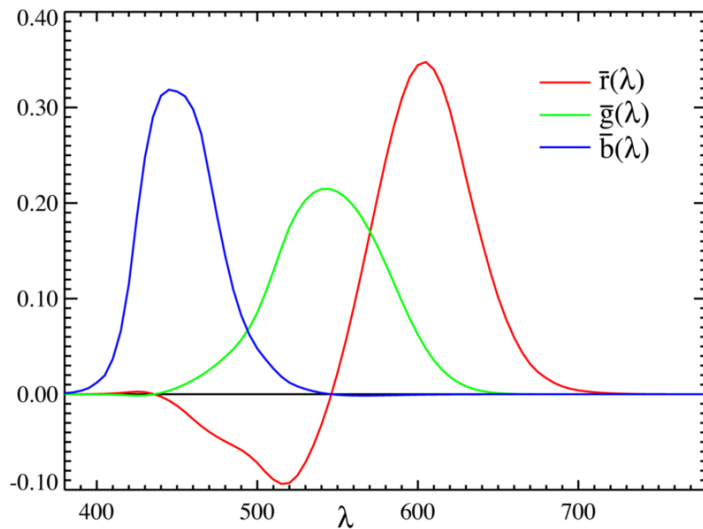
ako je to bežné pri RGB doostrení. Naopak rozostrením kanálov a, b je možné účinne potlačiť farebný šum s tým, že vďaka absencii rozostrenia v kanále L oko toto rozostrenie vôbec nevníma. Podobne na modeli Lab je založená celá rada ďalších metód editácie či zvláštnych efektov. [1]

## 1.6 Farebný priestor CIE 1931

Farebný priestor CIE 1931, zkonštruovala Medzinárodná kolorimetrická organizácia (CIE z francúzskeho Commission Internationale de l'Eclairage) v roku 1931.

Ku vzniku najviac prispeli experimenty, ktoré nezávisle na sebe v 20. rokoch 19. storočia prevádzali W. David Wright (1928) and John Guild (1931). Experimenty sa týkali hlavne javu zvaného metamerizmus.

Kruhový terčik o uhlovej veľkosti  $2^\circ$ , čo odpovedá pokrytiu žltej škvrny v ľudskom oku, v nich bol rozdelený na polovice. Na jednu z nich sa premietalo svetlo o regulovateľnej vlnovej dĺžke a konštantnej intenzite. Na druhú sa premietalo svetlo z troch monochromatických zdrojov o vlnových dĺžkach  $\lambda_R = 645 \text{ nm}$ ,  $\lambda_G = 526 \text{ nm}$  a  $\lambda_B = 444 \text{ nm}$ , ktorých intenzity bolo možné nezávisle regulovať. Intenzity troch monochromatických svetiel potrebné pre dosiahnutie optického vnemu rovnakej farby, ako z druhej polovice terčiku sa vynášali do grafov. U určitých farieb nebolo možné dosiahnuť zhody inak, než že sa naopak niektorým z troch monochromatických zdrojov dosvecovala polovica terčiku s testovanou farbou. V takomto prípade sa intenzita dosvecovacej farby vynášala ako záporné číslo. Experiment bol opakovaný s mnohými testovanými osobami a celkové výsledky boli potom spriemerované.

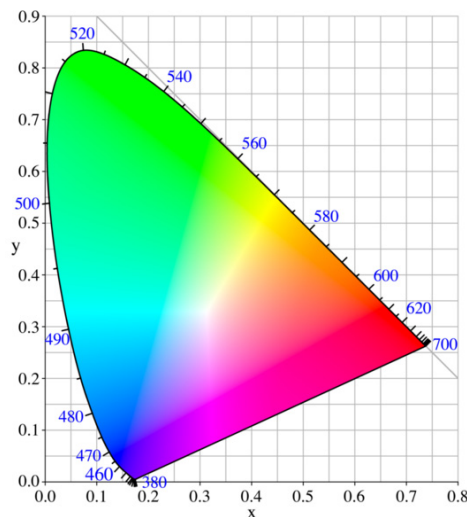


Obr. 11. Výsledné krivky z Wright - Guildovho experimentu

### Vlastnosti modelu CIE 1931

Každá farba sa dá popísať jej tónom, sýtosťou a jasom. Farebný model CIE 1931 sa znázorňuje v 2D diagrame, ktorý obsahuje len farby s plným jasom. Dá sa v ňom teda pracovať len s tónom a sýtosťou. Pre prácu s farbou vrátane jej jasnosti už nestačí 2D model a musí sa prejsť k 3D farebnému modelom.

Na krivke podkovovitého tvaru sú vynesené tzv. spektrálne teda prirodzené farby. Na ľavom konci je vlnová dĺžka 380nm odpovedajúca fialovej farbe, napravo potom dĺžka 700nm odpovedajúca červenej farbe. Na vrchole podkovy je vlnová dĺžka 520nm odpovedajúca zelenej farbe. Spektrálne sa týmto farbám hovorí preto, že ich rozloženie po podkove odpovedá rozloženiu vo farebnom spektre po rozložení svetla hranolom.



Obr. 12. Štandardný kolorimetrický trojuholník CIE 1931

Na priamke, ktorá spojuje konce podkovy potom ležia takzvané nespektrálne teda neprirodzené farby. Vznikajú miešaním koncových farieb spektra a preto ich nenájdeme medzi farbami hranolom rozloženého spektra.

Na súradniciach  $x=1/3$ ,  $y=1/3$  leží farba biela označovaná W (White).

Diagram CIE 1931 má jednu dôležitú vlastnosť. Všetky farby vzniknuté zmiešaním ľubovoľných dvoch nájdeme v diagrame na úsečke, ktorá spojuje práve tieto dve miešané farby.

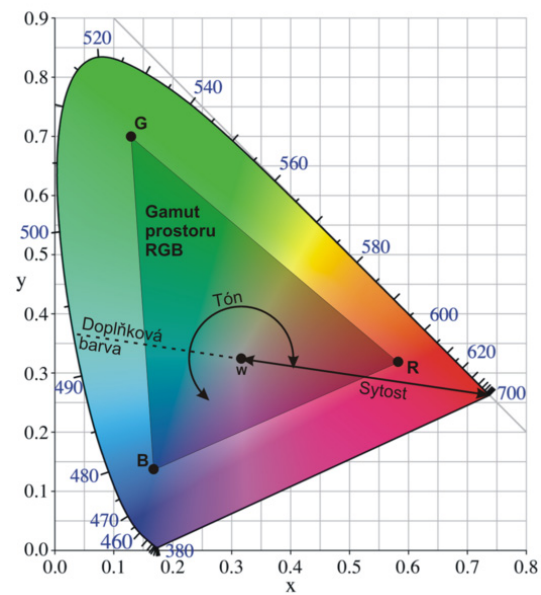
Sýtosť farby je daná pomerom čistej spektrálnej alebo nespektrálnej farby a farby bielej. Preto všetky farby určitého tónu a rôznej sýtosti nájdeme v diagrame na spojnici bielej a spektrálnej alebo nespektrálnej farby daného tónu.

Tón farby teda v diagrame hľadáme otáčaním takejto spojnice okolo bielej farby. Preto sa tiež tón farieb vyjadruje v uhlových stupňoch.

Doplňková farba k ľubovoľnej farbe je taká farba, ktorej tón leží na opačnej polpriamke priamky danej farby prechádzajúcej bielou. Jej tón má teda hodnotu o  $180^\circ$  rozdielnu.

Pravidlo, že farby vzniknuté zmiešaním ľubovoľných dvoch leží na ich spojnici sa dá rozšíriť aj pre viac farieb. Ak spojíme napríklad tri ľubovoľné farby čiarami, vznikne trojuholník obsahujúci všetky farebné kombinácie, ktoré môžu vzniknúť zmiešaním týchto troch farieb. Takémuto farebnému priestoru hovoríme farebný **gamut**.

Čím je farebný gamut väčší, tím viac farieb môžeme použitím troch základných namiešať. Najväčšieho gamutu sa dosahuje použitím farieb červenej, zelenej a modrej, ktoré používa práve model RGB. [2]



Obr. 13. Farebný rozsah modelu RGB

## 2 FAREBNÉ MODELY PRE TELEVÍZNU A VIDEO TECHNIKU

S doposiaľ uvedenými farebnými modelmi sa stretávame vo väčšine programov spracovávajúcich farebné rastrové obrázky. Nasledujúca skupina modelov je menej známa. Sú používané hlavne tam, kde sa stretáva počítačová grafika s televíznou technikou, tj. v oblasti animácií, filmových trikov, multimédií apod.

Medzi tieto modely patrí model YUV, ktorý sa používa pre prenos televíznych signálov v norme PAL. Obdobne existujú aj modely YIQ pre americkú normu NTSC a model YCbCr pre normu SECAM. Ich spoločným rysom je oddelenie jasovej zložky Y od farebných informácií tak, aby mohli byť používané na čiernobielych aj farebných televízoroch. Zostávajúce dva signály nesú informácie o veľkosti farebných zložiek obrazu.

Ďalšou výhodou je, že časť informácií môže byť vynechaná z dôvodu redukcie prenášaných údajov. Ľudské oko je oveľa citlivejšie na zmeny jasů, ako na farebné zmeny. Niektoré televízne normy, ako napríklad NTSC toto využíva a prenáša len 11% pôvodnej modrej a 30% pôvodnej červenej. Informácia o zelenej farbe je obvykle zachovaná v kanáli Y.

Informácia o jase má v televíznych modeloch rozsah  $\langle 0,1 \rangle$ , pre farebnú informáciu sa používa rozsah  $\langle -0.5, +0.5 \rangle$ . Ak chceme jednotlivé zložky pri počítačovom spracovaní ukladať ako celé čísla v rozsahu 0-255 musíme ich najskôr posunúť do kladnej polosi pričítaním konštanty.

Model YCbCr sa okrem televíznej techniky používa pri zápise rastrových dát do formátu jpeg. [6]

## **II. PRAKTICKÁ ČASŤ**

### 3 KONVERZNÉ ALGORITMY MEDZI FAREBNÝMI MODELMI

V praktickej časti bolo mojou úlohou vyhľadať a zostaviť algoritmy pre vzájomné prevody medzi farebnými modelmi. Použité algoritmy sú vytvorené v jazyku C/C++.

Všetky konverzie medzi farebnými modelmi sú prevádzané cez model RGB, ku ktorému sú voľne dostupné prevodné vzorce pre každý z vyššie uvedených farebných modelov.

#### 3.1 CMYK

##### 3.1.1 Konverzia CMYK → RGB

CMYK vstupné hodnoty - rozsah 0 ÷ 255

RGB výstupné hodnoty - rozsah 0 ÷ 255

```
R = ( 1 - C ) * ( 1 - K );
G = ( 1 - M ) * ( 1 - K );
B = ( 1 - Y ) * ( 1 - K );
```

##### 3.1.2 Konverzia RGB → CMYK

RGB vstupné hodnoty - rozsah 0 ÷ 255

CMYK výstupné hodnoty - rozsah 0 ÷ 255

```
var_R = 255 - R;
var_G = 255 - G;
var_B = 255 - B;
Min = Minimum( var_R , var_G , var_B );
if( Min==255 ) { C = M = Y = 0; K = 255; }
else {
C = ( var_R - Min ) / ( 255 - Min );
M = ( var_G - Min ) / ( 255 - Min );
Y = ( var_B - Min ) / ( 255 - Min );
K = Min; }
```

#### 3.2 HSV

##### 3.2.1 Konverzia HSV → RGB

HSV vstupné hodnoty - rozsah 0 ÷ 1

RGB výstupné hodnoty - rozsah 0 ÷ 255

```
if ( S == 0 )
{
R = V * 255;
G = V * 255;
B = V * 255;
}
else
{
var_h = H * 6;
if ( var_h == 6 ) var_h = 0;
var_i = int( var_h ); // var_i = floor( var_h )
```



```

var_1 = V * ( 1 - S );
var_2 = V * ( 1 - S * ( var_h - var_i ) );
var_3 = V * ( 1 - S * ( 1 - ( var_h - var_i ) ) );

if ( var_i == 0 ) { var_r = V ; var_g = var_3 ; var_b = var_1; }
else if ( var_i == 1 ) { var_r = var_2 ; var_g = V ; var_b = var_1; }
else if ( var_i == 2 ) { var_r = var_1 ; var_g = V ; var_b = var_3; }
else if ( var_i == 3 ) { var_r = var_1 ; var_g = var_2 ; var_b = V; }
else if ( var_i == 4 ) { var_r = var_3 ; var_g = var_1 ; var_b = V; }
else { var_r = V ; var_g = var_1 ; var_b = var_2; }

R = var_r * 255;
G = var_g * 255;
B = var_b * 255;
}

```

### 3.2.2 Konverzia RGB → HSV

RGB vstupné hodnoty - rozsah 0 ÷ 255

HSV výstupné hodnoty - rozsah 0 ÷ 1

```

var_R = ( R / 255 );
var_G = ( G / 255 );
var_B = ( B / 255 );

var_Min = min( var_R, var_G, var_B ); //Min. hodnota RGB
var_Max = max( var_R, var_G, var_B ); //Max. hodnota RGB
del_Max = var_Max - var_Min; //Delta RGB hodnota

V = var_Max;

if ( del_Max == 0 ) //šedá, žiadny jas...
{
    H = 0;
    S = 0;
}
else //výpočet Hue...
{
    S = del_Max / var_Max;

    del_R = ( ( ( var_Max - var_R ) / 6 ) + ( del_Max / 2 ) ) / del_Max;
    del_G = ( ( ( var_Max - var_G ) / 6 ) + ( del_Max / 2 ) ) / del_Max;
    del_B = ( ( ( var_Max - var_B ) / 6 ) + ( del_Max / 2 ) ) / del_Max;

    if ( var_R == var_Max ) H = del_B - del_G;
    else if ( var_G == var_Max ) H = ( 1 / 3 ) + del_R - del_B;
    else if ( var_B == var_Max ) H = ( 2 / 3 ) + del_G - del_R;

    if ( H < 0 ) ; H += 1;
    if ( H > 1 ) ; H -= 1;
}

```

## 3.3 HSL

### 3.3.1 Konverzia HSL → RGB

HSL vstupné hodnoty - rozsah 0 ÷ 1

RGB výstupné hodnoty - rozsah 0 ÷ 255

```

if ( S == 0 )
{
    R = L * 255;
    G = L * 255;
    B = L * 255;
}
else
{
    if ( L < 0.5 ) var_2 = L * ( 1 + S );
    else          var_2 = ( L + S ) - ( S * L );
    var_1 = 2 * L - var_2;

    R = 255 * Hue_2_RGB( var_1, var_2, H + ( 1 / 3 ) );
    G = 255 * Hue_2_RGB( var_1, var_2, H );
    B = 255 * Hue_2_RGB( var_1, var_2, H - ( 1 / 3 ) );
}

double Hue_2_RGB(double v1, double v2, double vH ) //Funkcia Hue_2_RGB
{
    if ( vH < 0 ) vH += 1;
    if ( vH > 1 ) vH -= 1;
    if ( ( 6 * vH ) < 1 ) return ( v1 + ( v2 - v1 ) * 6 * vH );
    if ( ( 2 * vH ) < 1 ) return ( v2 );
    if ( ( 3 * vH ) < 2 ) return ( v1 + ( v2 - v1 ) * (( 2/3 )-vH) * 6 );
    return ( v1 );
}

```

### 3.3.2 Konverzia RGB → HSL

RGB vstupné hodnoty - rozsah 0 ÷ 255

HSL výstupné hodnoty - rozsah 0 ÷ 1

```

var_R = ( R / 255 );
var_G = ( G / 255 );
var_B = ( B / 255 );

var_Min = min( var_R, var_G, var_B ); //Min. hodnota RGB
var_Max = max( var_R, var_G, var_B ); //Max. hodnota RGB
del_Max = var_Max - var_Min; //Delta RGB hodnota

L = ( var_Max + var_Min ) / 2;

if ( del_Max == 0 ) //šedá, žiadny jas...
{
    H = 0;
    S = 0;
}
else //výpočet Hue...
{
    if ( L < 0.5 ) S = del_Max / ( var_Max + var_Min );
    else          S = del_Max / ( 2 - var_Max - var_Min );

    del_R = ( ( ( var_Max - var_R ) / 6 ) + ( del_Max / 2 ) ) / del_Max;
    del_G = ( ( ( var_Max - var_G ) / 6 ) + ( del_Max / 2 ) ) / del_Max;
    del_B = ( ( ( var_Max - var_B ) / 6 ) + ( del_Max / 2 ) ) / del_Max;

    if ( var_R == var_Max ) H = del_B - del_G;
    else if ( var_G == var_Max ) H = ( 1 / 3 ) + del_R - del_B;
    else if ( var_B == var_Max ) H = ( 2 / 3 ) + del_G - del_R;

    if ( H < 0 ) ; H += 1;
}

```

```

    if ( H > 1 ) ; H -= 1;
}

```

### 3.4 YUV

Farebný model YUV má väčší farebný rozsah ako model RGB, preto je nutné zabezpečiť aby výsledné hodnoty tento rozsah nepresiahli. Rozsah hodnôt modelu YUV pre farebné informácie je od -0.5 do +0.5, preto sa pri prevode prevádza posun hodnôt do kladnej časti(RGB→YUV) a naopak pri opačnom prevode.

#### 3.4.1 Konverzia YUV → RGB

YUV vstupné hodnoty - rozsah 0 ÷ 1

RGB výstupné hodnoty - rozsah 0 ÷ 255

```

R = ( Y + 1.13983 * ( V - 0.5 ) ) * 255;
G = ( Y - 0.39466 * ( U - 0.5 ) - 0.58060 * ( V - 0.5 ) ) * 255;
B = ( Y + 2.03211 * ( U - 0.5 ) ) * 255;
R = MimoRozsah( R );
G = MimoRozsah( G );
B = MimoRozsah( B );

```

```

double MimoRozsah(double val) // funkcia mimo rozsah
{
    if( val < 0 ) return 0;
    else if ( val > 255 ) return 255;
    else return val;
}

```

#### 3.4.2 Konverzia RGB → YUV

RGB vstupné hodnoty - rozsah 0 ÷ 255

YUV výstupné hodnoty - rozsah 0 ÷ 1

```

varR = R / 255;
varG = G / 255;
varB = B / 255;

Y = 0.299 * varR + 0.587 * varG + 0.114 * varB;
U = - 0.14713 * varR - 0.28886 * varG + 0.436 * varB + 0.5;
V = 0.615 * varR - 0.51499 * varG - 0.10001 * varB + 0.5;

```

### 3.5 YIQ

Farebný rozsah modelu YIQ je podobne ako pri modeli YUV väčší ako rozsah modelu RGB a musí sa použiť rovnaká korekcia hodnôt, ako aj posun hodnôt do kladnej časti.

#### 3.5.1 Konverzia YIQ → RGB

YIQ vstupné hodnoty - rozsah 0 ÷ 1

RGB výstupné hodnoty - rozsah 0 ÷ 255

```

R = ( Y + 0.9563 * ( I - 0.5 ) + 0.6210 * ( Q - 0.5 ) ) * 255;
G = ( Y - 0.2721 * ( I - 0.5 ) - 0.6473 * ( Q - 0.5 ) ) * 255;

```

```

B = ( Y + 1.1070 * ( I - 0.5 ) + 1.7046 * ( Q - 0.5 ) ) * 255;
R = MimoRozsah( R );
G = MimoRozsah( G );
B = MimoRozsah( B );

```

### 3.5.2 Konverzia RGB → YIQ

RGB vstupné hodnoty - rozsah 0 ÷ 255

YIQ výstupné hodnoty - rozsah 0 ÷ 1

```

varR = R / 255;
varG = G / 255;
varB = B / 255;

```

```

Y = 0.299 * varR + 0.587 * varG + 0.114 * varB;
I = 0.5957 * varR - 0.2744 * varG - 0.3212 * varB + 0.5;
Q = 0.2114 * varR - 0.5226 * varG + 0.3111 * varB + 0.5;

```

## 3.6 YCbCr

Pri tomto modeli nastáva rovnaká situácia ako pri predchádzajúcich modeloch pre televíznu techniku a preto sú použité rovnaké korekčné postupy.

### 3.6.1 Konverzia YCbCr → RGB

YCbCr vstupné hodnoty - rozsah 0 ÷ 1

RGB výstupné hodnoty - rozsah 0 ÷ 255

```

R = ( Y + 1.402 * ( Cr - 0.5 ) ) * 255;
G = ( Y - 0.34414 * ( Cb - 0.5 ) - 0.71414 * ( Cr - 0.5 ) ) * 255;
B = ( Y + 1.772 * ( Cb - 0.5 ) ) * 255;
R = MimoRozsah( R );
G = MimoRozsah( G );
B = MimoRozsah( B );

```

### 3.6.2 Konverzia RGB → YCbCr

RGB vstupné hodnoty - rozsah 0 ÷ 255

YCbCr výstupné hodnoty - rozsah 0 ÷ 1

```

varR = R / 255;
varG = G / 255;
varB = B / 255;

```

```

Y = 0.299 * varR + 0.587 * varG + 0.114 * varB;
Cb = - 0.1687 * varR - 0.3313 * varG + 0.5 * varB + 0.5;
Cr = 0.5 * varR - 0.4187 * varG - 0.0813 * varB + 0.5;

```

## 4 TRIEDA cColorConversion

Na základe vyššie uvedených algoritmov som v jazyku C++ vytvoril triedu, ktorá tieto algoritmy využíva. Triedu je možné použiť prakticky v ľubovoľnej aplikácii, ktorá potrebuje prevádzať farby medzi farebnými modelmi.

### 4.1 Popis triedy

Definícia:

```
class cColorConversion
{
private:
    enum eInType InType;
    double Hue_2_RGB(double v1, double v2, double vH);
    void DisplayError(const char *message);
    double Max3(double in1, double in2, double in3);
    double Min3(double in1, double in2, double in3);
    double OutOfRange(double val);
public:
    tRGB RGB;
    tCMYK CMYK;
    tHSL HSL;
    tHSV HSV;
    tYUV YUV;
    tYIQ YIQ;
    tYCBCR YCBCR;
    cColorConversion();
    bool LoadColor(enum eInType IT, double in1, double in2, double
in3, double in4);
    void Convert();
};
```

V privátnej časti obsahuje trieda premennú výčtového typu eInType, ktorý je definovaný ako:

```
enum eInType { undefined, eRGB, eCMYK, eHSL, eHSV, eYUV, eYIQ, eYCBCR };
```

Táto premenná predstavuje typ vstupného farebného modelu.

Ďalej nasledujú privátne funkcie potrebné pre chod niektorých konverzných algoritmov, ako napríklad funkcie vracajúce minimum a maximum z troch hodnôt, funkciu na ošetrovanie prekročenia rozsahu hodnôt. Ďalej funkciu DisplayError, ktorá môže zobrazit' chybu podľa typu vytváratej aplikácie(konzola, alebo Windows aplikácia).

Vo verejnej časti triedy sa nachádzajú štruktúry jednotlivých farebných modelov definované ako:

```
typedef struct {
    double R, G, B;
} tRGB;

typedef struct {
    double C,M,Y,K;
} tCMYK;

typedef struct {
    double H,S,L;
} tHSL;

typedef struct {
    double H,S,V;
} tHSV;

typedef struct {
    double Y,U,V;
} tYUV;

typedef struct {
    double Y,I,Q;
} tYIQ;

typedef struct {
    double Y,CB,CR;
} tYCBCR;
```

Premenné obsiahnuté v týchto štruktúrach sú typu double v rozsahu 0-1.

Ďalej sa tu nachádza konštruktor, funkcia LoadColor a funkcia Convert.

## 4.2 Práca s triedou

Po vytvorení inštancie triedy priradí konštruktor do premennej InType hodnotu undefined, značiacu že nie je zvolený žiadny vstupný farebný model, čo zabráni prípadnému volaniu metódy Convert.

Následne sa očakáva, že pomocou metódy LoadColor „nahráme“ určitú farbu v niektorom z dostupných farebných modelov. Definícia metódy je nasledovná:

```
bool LoadColor(enum eInType IT, double in1, double in2, double in3, double in4);
```

Sú očakávané 4 parametre, piaty parameter je nepovinný a používa sa len pri farebných modeloch s počtom zložiek > 3, napríklad CMYK.

Prvý parameter určuje typ zadávaného farebného modelu, ďalšie parametre obsahujú hodnoty jednotlivých zložiek daného farebného modelu v rozsahu 0-1.

Príklad „nahratia“ farebného modelu HSV:

```
cColorConversion farba;  
farba.LoadColor(eHSV, H_zlozka, S_zlozka, V_zlozka);
```

V prípade úspechu vracia funkcia hodnotu `true`, v opačnom prípade hodnotu `false`.

Funkcia prevedie farbu zo zadaného farebného modelu do modelu RGB a výsledok uloží do štruktúry RGB. Premenná `InType` obsahuje typ vstupného farebného modelu (vo vyššie uvedenom príklade `eHSV`).

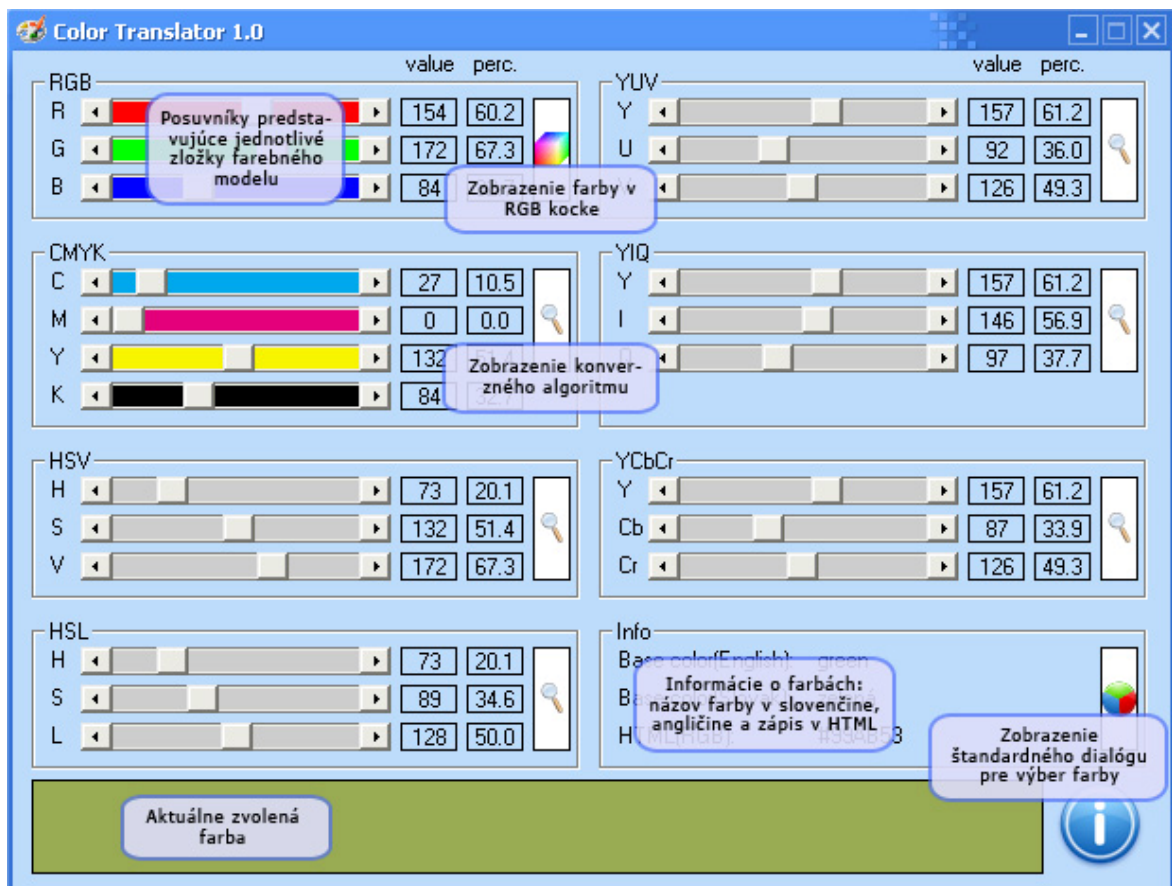
V ďalšom kroku voláme metódu `Convert()`, ktorá prevedie farbu uloženú v štruktúre RGB do všetkých dostupných farebných modelov a týmito hodnotami naplní štruktúry jednotlivých farebných modelov. Podľa hodnoty premennej `InType` sa zaistí, aby sa nespracoval prevod späť do vstupného farebného modelu, ktorý by mohol vzhľadom na zaokrúhľovanie a prevádzané matematické operácie mierne pozmeniť vstupnú hodnotu a tým zabráni prípadnému zacykleniu programu.

## 5 Aplikácia ColorTranslator

Mojou ďalšou úlohou bolo zostaviť program pre Windows, ktorý by využíval vyššie popísané prevodné algoritmy a teda aj triedu cColorConversion.

Pre vytvorenie aplikácie som si vybral jazyk C/C++ a prostredie WIN32 API, ktoré je pri menších aplikáciách relatívne jednoduché.

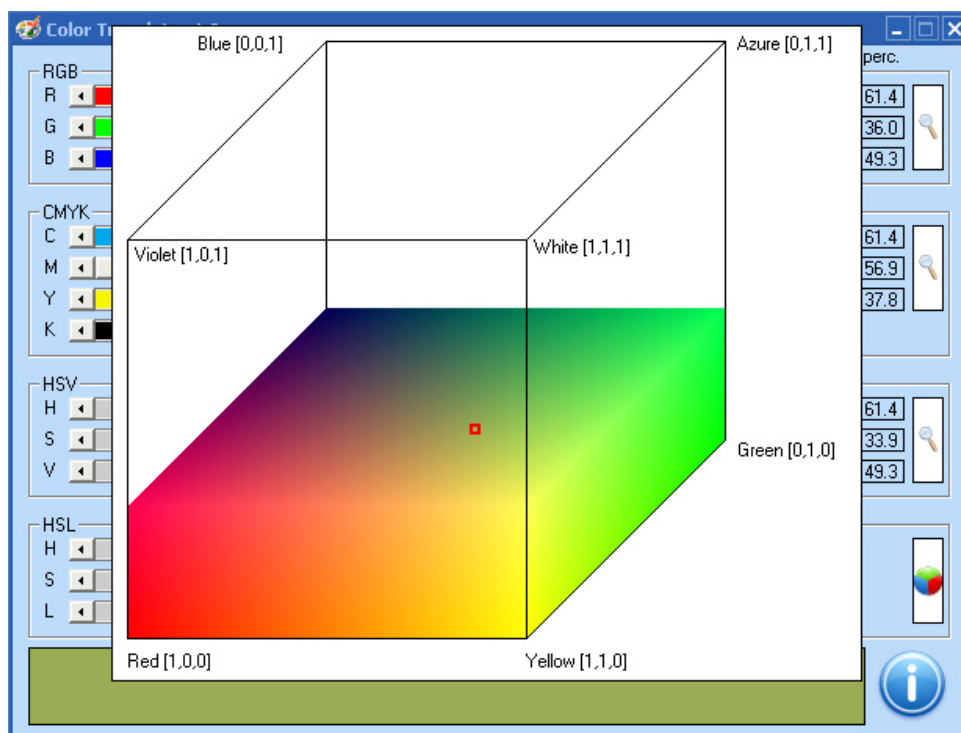
Vzhľadom na to, že aplikácia môže byť používaná k výuke, snažil som sa aby prevody medzi farebnými modelmi boli čo najviac názorné a bolo hneď vidieť, ako ktorá zložka súvisí s ostatnými pri prevodoch. Z tohto dôvodu som namiesto číselného zadávania hodnôt vytvoril aplikáciu, ktorá pozostáva z rady posuvníkov, z ktorých každý patrí jednej zložke farebného modelu. Pri posúvaní je farba vybraného farebného modelu prevedená do ostatných farebných modelov a sú nastavené pozície ostatných posuvníkov podľa výsledku konverzie pre každý farebný model.



Obr. 14. Okno aplikácie ColorTranslator



Pomocou tlačidiel nachádzajúcich sa napravo v bloku každého farebného modelu je možné zobrazit' konverzný algoritmus pre prevod daného farebného modelu do a z modelu RGB. Výnimkou je len model RGB, kde stlačenie tlačidla spôsobí zobrazenie kocky RGB s vyznačením umiestnenia práve vybranej farby.



Obr. 15. Zobrazenie práve vybranej farby v kocke RGB

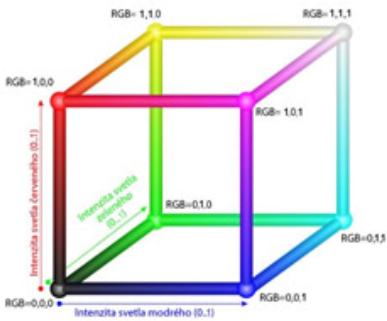
Blok Info zobrazuje názov vybranej základnej farby, prípadne text biela, šedá, alebo čierna v slovenskom a anglickom jazyku. Ďalej je tu možné nájsť zápis farby v HTML kóde a tlačidlo pre zobrazenie štandardného dialógu Windows pre výber farby.

## 6 PREZENTÁCIA K VÝUKE

Na základe teoretickej časti som vytvoril prezentáciu v Powerpointe, ktorú je možné využívať k výuke predmetu Počítačová grafika. Prezentácia popisuje vlastnosti jednotlivých farebných modelov, spôsoby ich použitia, výhody a nevýhody.

**Barevný model RGB**

- Barevný model RGB se svým rozsahem nejčastěji reprezentuje jako jednotková kostka umístěná v osách r, g, b. Z toho vyplývá, že množina základních barev obsahuje osm barev. Vrchol  $[0,0,0]$  (t.j. střed souřadnicového systému) odpovídá černé barvě (Black). Naproti tomu vrchol  $[1,1,1]$  odpovídá bílé barvě (White). Barvy ležící na diagonále mezi těmito vrcholy odpovídají odstínům šedé (Gray).

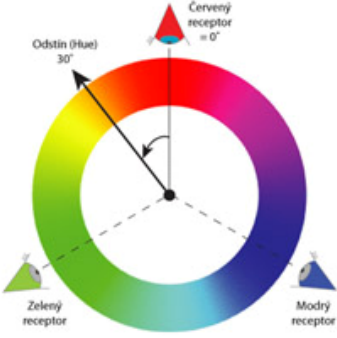


Obr. 16. Snímok prezentácie

**Barevný model HSV (HSB)**

Barevný model HSV používá podobně jako model RGB též tři hodnoty pro popis barvy, dává jim ale jiný význam:

- Odstín barvy (Hue)** popisuje vlastní čistou barvu (tedy např. červená, zelená, modrá). Asi nepřekvapí, že pro popis barvy se používá úhel na barevném kole - tedy rozsah  $0 - 360^\circ$ . Dohodou se za úhel  $0^\circ$  považuje červená barva,  $120^\circ$  odpovídá zelené a  $240^\circ$  modré barvě,  $360^\circ$  opět červené, protože jsme obešli kruh dokola.



Obr. 17. Snímok prezentácie

## ZÁVER

Hlavným cieľom tejto bakalárskej práce bolo vytvorenie teoretického zhrnutia na tému farebných modelov. V teoretickej časti som sa zameril hlavne na popis vlastností, použitie, výhody a nevýhody vybraných farebných modelov.

V praktickej časti som zostavil algoritmy pre vzájomné prevody medzi farebnými modelmi, vytvoril triedu v jazyku C++ ktorá tieto algoritmy využíva. V ďalšej časti som sa zameril na vytvorenie aplikácie pre Windows ktorá využíva spomínanú triedu na vizuálne znázornenie prevodných vzťahov medzi farebnými modelmi. V záverečnej časti sa práca zameriava na vytvorenie prezentácie, ktorú je možné použiť pri výuke predmetu Počítačová grafika.

## ZÁVER V ANGLIČTINE

The main aim of the work was creation of theoretical summary in the field of color models. In the theoretical part I focused mainly on description of qualities, applications, advantages and disadvantages of chosen color models.

In the practical part I made up algorithmus for the mutual conversions among the color models, and designed the class in the C++ language which uses these algorithms. In the next part I focused on designing Windows application, which uses the class for visualization of the transfer relations among the color models. The final part is focused on designing the presentation, which can be used for education in the subject Počítačová grafika

**ZOZNAM POUŽITEJ LITERATÚRY**

- [1] Pihan Roman, fotografovani.cz : Vše o světle [online]. [cit. 5.23.2007]. Dostupný z WWW: [http://www.fotografovani.cz/art/fozak\\_df/rom\\_1\\_01\\_cojetosvetlo.html](http://www.fotografovani.cz/art/fozak_df/rom_1_01_cojetosvetlo.html)
- [2] PanWiki [online]. [cit. 5.23.2007]. Dostupný z WWW: [http://panwiki.panska.cz/index.php/Barevný\\_prostor\\_CIE\\_1931](http://panwiki.panska.cz/index.php/Barevný_prostor_CIE_1931)
- [3] Wikipedia – Internetová encyklopédia [online]. [cit. 5.23.2007]. Dostupný z WWW: <http://wikipedia.org>
- [4] Otakar Čerba : Barvy v počítači a v kartografii [online]. [cit. 5.23.2007]. Dostupný z WWW: <http://gis.zcu.cz/studium/pok/Materialy/Book/ar03.html>
- [5] Výuka počítačovej grafiky na Technickej univerzite v Košiciach [online]. [cit. 5.23.2007]. Dostupný z WWW: <http://hornad.fei.tuke.sk/predmety/pg/index.php?action=zaklad/farby.htm>
- [6] Žára, J. a kol. : Moderní počítačová grafika. Computer Press, 2005
- [7] Charles Petzold : Programování ve Windows. Computer Press, 1999

**ZOZNAM POUŽITÝCH SYMBOLOV A ZKRATIEK**

**RGB** Red, Green, Blue (červená, zelená, modrá )

**CMYK** Cyan, Magenta, Yellow, Key (blacK) (azúrová, purpurová, žltá, čierna)

**HSV** Hue, Saturation, Value (odtieň farby, saturácia/sýtosť, jas)

**HSL** Hue, Saturation, Lightness (odtieň farby, saturácia/sýtosť, svetlosť)

**CIE** Commission Internationale de l'Eclairage (Medzinárodná kolorimetrická organizácia)

**ZOZNAM OBRÁZKOV**

Obr. 1. Aditívne miešanie farieb.....	10
Obr. 2. Reprezentácia farebného modelu RGB .....	11
Obr. 3. Subtraktívne miešanie farieb .....	12
Obr. 4. Reprezentácia farebného modelu CMY .....	13
Obr. 5. Výber odtieňa farby (Hue).....	14
Obr. 6. Výber jasú (Value/Brightness) a sýtosti (Saturation).....	14
Obr. 7. Reprezentácia farebného modelu HSV.....	15
Obr. 8. Dialóg Adobe Photoshopu pre zmenu odtieňa, sýtosti a svetlosti fotografií .....	15
Obr. 9. Reprezentácia farebného modelu HSL .....	16
Obr. 10. Reprezentácia modelu LAB.....	17
Obr. 11. Výsledné krivky z Wright - Guildovho experimentu .....	19
Obr. 12. Štandardný kolorimetrický trojuholník CIE 1931 .....	20
Obr. 13. Farebný rozsah modelu RGB .....	21
Obr. 14. Okno aplikácie ColorTranslator .....	32
Obr. 15. Zobrazenie práve vybranej farby v kočke RGB.....	33
Obr. 16. Snímok prezentácie.....	34
Obr. 17. Snímok prezentácie.....	34

## ZOZNAM PRÍLOH

P I Príklady rozkladu obrazu na jednotlivé farebné zložky

P II CD obsahujúce zdrojové kódy k triede `cColorConversion` a aplikácii `ColorTranslator`, prezentáciu v Powerpointe na tému farebných modelov a túto prácu v elektronickej podobe.



## PRÍLOHA P I: PRÍKLADY ROZKLADU OBRAZU NA JEDNOTLIVÉ FAREBNÉ ZLOŽKY

Obrázok rozložený na zložky RGB



Pôvodný obrázok



R - Zložka



G - Zložka



B - Zložka

Obrázok rozložený na zložky CMYK



C - Zložka



M - Zložka



Y - Zložka

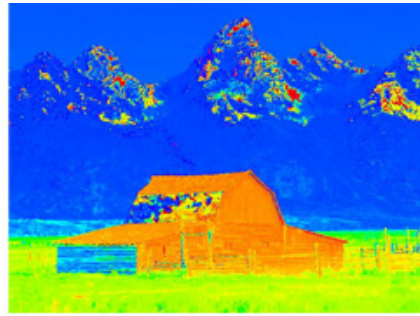


K - Zložka

Obrázok rozložený na zložky HSV



Pôvodný obrázok



H - Zložka



S - Zložka



V - Zložka

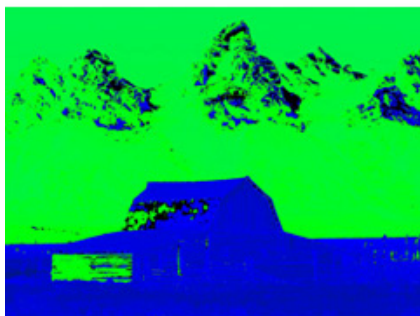
Obrázok rozložený na zložky YUV



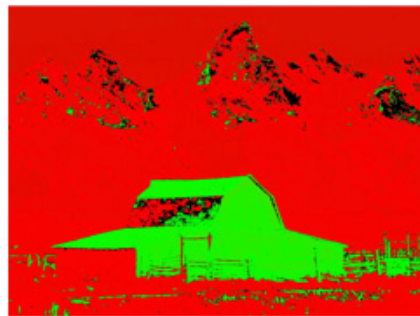
Pôvodný obrázok



Y - Zložka



U - Zložka



V - Zložka

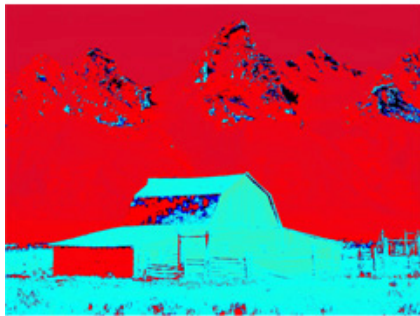
Obrázok rozložený na zložky YIQ



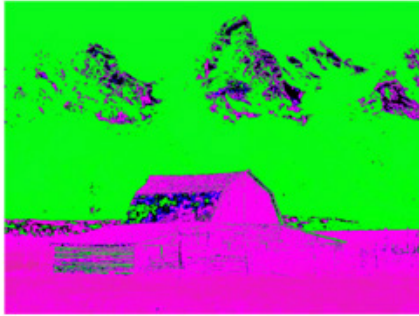
Pôvodný obrázok



Y - Zložka



I - Zložka



Q - Zložka

Obrázok rozložený na zložky YCbCr



Pôvodný obrázok



Y - Zložka



Cb - Zložka



Cr - Zložka