

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

## INTERAKTIVNÍ WEBOVÉ APLIKACE PRO PODPORU VÝUKY ZPRACOVÁNÍ OBRAZU

INTERACTIVE WEB APPLICATIONS FOR IMAGE PROCESSING EDUCATION

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Filip Krolop

### VEDOUCÍ PRÁCE

SUPERVISOR

doc. Mgr. Pavel Rajmic, Ph.D.

BRNO 2020

# Bakalářská práce

bakalářský studijní program **Telekomunikační a informační systémy**

Ústav telekomunikací

**Student:** Filip Krolp

**ID:** 195372

**Ročník:** 3

**Akademický rok:** 2019/20

**NÁZEV TÉMATU:**

## **Interaktivní webové aplikace pro podporu výuky zpracování obrazu**

**POKYNY PRO VYPRACOVÁNÍ:**

Nastudujte problematiku vztahující se k výukovým apletům, navrhnete jejich funkcionalitu a ovládání, implementujte a odladte je. Půjde o interaktivní podání algoritmů používaných v oblasti zpracování obrazů, konkrétně budou aplety zaměřeny na barevné modely, jejich vztahy a převody: 1/ Paletová (indexová) reprezentace obrazu, 2/ Míchání barev v různých modelech, 3/ Převod barevného obrazu na šedotónový, 4/ Převod šedotónového obrazu na barevný (tzv. nepravé barvy). Aplety implementujte pomocí HTML a JavaScriptu tak, aby demonstrovaly příslušnou teorii a vybízely studenty k interakci.

**DOPORUČENÁ LITERATURA:**

[1] Beneš, B.; Sochor, J.; Felkel, P.; Žára, J.: Moderní počítačová grafika. Computer Press, Brno, 2005.

[2] Gonzalez, R.C.; Woods, R.E.: Digital Image Processing. Třetí vydání. Pearson; 2007. ISBN 978-0131687288

**Termín zadání:** 3.2.2020

**Termín odevzdání:** 8.6.2020

**Vedoucí práce:** doc. Mgr. Pavel Rajmic, Ph.D.

**prof. Ing. Jiří Mišurec, CSc.**  
předseda rady studijního programu

**UPOZORNĚNÍ:**

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Uvedení do problematiky vztahující se k výukovým appletům, návrh jejich funkcionality a ovládání, následovaný vlastní implementací a odladěním. Jedná se o interaktivní podání algoritmů používaných v oblasti zpracování obrazů, konkrétně jsou applety zaměřeny na barevné modely, jejich vztahy a převody: 1/ Paletová (indexová) reprezentace obrazu, 2/ Míchání barev v různých modelech, 3/ Převod barevného obrazu na šedotónový, 4/ Převod šedotónového obrazu na barevný (tzv. nepravé barvy). Applety implementovány pomocí HTML a JavaScriptu tak, aby demonstrovaly příslušnou teorii a vybízely studenty k interakci.

## KLÍČOVÁ SLOVA

Základy počítačové grafiky, barevné modely, barevné prostory, míchání barev, RGB, CMY, CMYK, CIEXYZ, Lab, HSB, YCbCr, umělé obarvení obrazu, nepravé barvy, parametrická křivka, Schrödingerova krychle, barevná paleta, barvicí paleta, indexová paleta, gamma korekce.

## ABSTRACT

Introduction to theory related to applets that will be used for educational purposes, creation of custom design, draft of their basic functionality and control elements, followed by own implementation and optimization. This is an interactive submission of image processing algorithms specifically, applets are focused on color models, their relationships and calculations between each other: 1 / Pallet (index) image representation, 2 / Color mixing in different models, 3 / Color image conversion to grayscale image, 4 / Conversion of a grayscale image to color image (false colors). Applets implemented using HTML and JavaScript to demonstrate the theory and encourage students to interact.

## KEYWORDS

Basics of computer graphics, color models, color spaces, color mixing, RGB, CMYK, CIEXYZ, Lab, HSB, YCbCr, artificial coloring, false colors, parametric curve, Schrödinger cube, color palette, index palette, gamma correction.

KROLOP, Filip. *Interaktivní webové aplikace pro podporu výuky zpracování obrazu*. Brno, Rok, 43 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: doc. Mgr Pavel Rajmic, Ph.D.



## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Interaktivní webové aplikace pro podporu výuky zpracování obrazu“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu doc. Mgr. Pavlu Rajmicovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno .....

.....

podpis autora

# Obsah

Úvod	8
<b>1 Teoretický úvod</b>	<b>9</b>
1.1 Úvod k programovací části práce	9
1.1.1 HTML	9
1.1.2 JavaScript	10
1.1.3 CSS	10
1.2 Základní pojmy v počítačové grafice	11
1.2.1 Světlo, jeho povaha a barva	11
1.2.2 Barevné prostory, modely a jejich vzájemné převody	13
1.2.3 RGB a CMY(K)	13
1.2.4 CIEXYZ	14
1.2.5 L*a*b (CIELAB)	15
1.2.6 Převod z RGB do Lab	16
1.2.7 Průměrování RGB kanálů	18
1.2.8 Selektivní míchání RGB kanálů	19
1.2.9 HSB	20
1.2.10 YCbCr	21
1.3 Metody umělého obarvení obrazu	22
1.3.1 Metoda parametrické křivky	23
1.3.2 Metoda Schrödingerovy krychle	24
1.3.3 Metoda barevné palety	24
1.3.4 Další metody umělého obarvení obrazu	27
1.3.5 Gamma korekce	29
<b>2 Rozbor práce</b>	<b>30</b>
2.1 Paletová (indexová) reprezentace obrazu	30
2.2 Míchání barev v různých modelech	33
2.3 Převod šedotónového obrazu na barevný (nepravé barvy)	36
<b>3 Závěr</b>	<b>38</b>
<b>Literatura</b>	<b>39</b>
<b>A Obsah přiloženého CD</b>	<b>43</b>

# Seznam obrázků

1.1	Lidské oko . . . . .	12
1.2	Citlivost čípků dle vlnové délky . . . . .	12
1.3	Model RGB . . . . .	14
1.4	Srovnání RGB a CMY modelu . . . . .	15
1.5	Chromatický diagram CIE XYZ . . . . .	15
1.6	Prostor Lab - osy . . . . .	16
1.7	Převod RGB -> XYZ . . . . .	17
1.8	Převod XYZ -> Lab . . . . .	18
1.9	Průměrování RGB . . . . .	19
1.10	Náhled appletu 1 . . . . .	20
1.11	HSB Color Model . . . . .	21
1.12	YCbCr - poměr složek dle ITU-T . . . . .	22
1.13	Obr: Převod RGB -> YCbCr . . . . .	22
1.14	RGB krychle . . . . .	23
1.15	Schrödingerovy krychle . . . . .	24
1.16	Implementace LUT . . . . .	26
1.17	Barevné palety . . . . .	27
2.1	Náhled appletu 2 . . . . .	31
2.2	Náhled appletu 2 . . . . .	32
2.3	Náhled appletu 2 . . . . .	33
2.4	Náhled appletu 2 . . . . .	34
2.5	Náhled appletu 3 . . . . .	35
2.6	Náhled appletu 3 . . . . .	36
2.7	Náhled appletu 3 . . . . .	36
2.8	Náhled appletu 4 . . . . .	37

# Úvod

Hlavním úkolem této bakalářské práce bylo vytvořit grafické návrhy a posléze implementovat funkcionalitu jednotlivých appletů sloužících pro podporu výuky zpracování obrazu. Tyto mnou vytvořené webové aplikace mohou být rovněž využité jako pomůcka studentům pro osvojení teorie probírané na přednáškách, blíže popsané ve skriptech k předmětu, a jejich domácí přípravě k úspěšnému absolvování předmětu Základy počítačové sazby a grafiky BZSG. Applety, které jsem v rámci zadaného tématu pro mou bakalářskou práci tvořil, se zabývají problematikou převodu barevného obrázku na šedotónový, vztahy a přepočty při míchání barev napříč různými barevnými modely a prostory, principem paletové, neboli indexové reprezentace bitmapového obrazu a převodu šedotónového obrazu do nepravých barev.

Svou bakalářskou práci jsem rozčlenil do několika základních kapitol.

- V první části teoretického úvodu se krátce zmiňuji k oběma použitým jazykům, programovacímu (JavaScript) i značkovacímu jazyku (HTML), ale i ke kaskádovému stylům CSS, které byly ve vytvořených appletech rovněž využity.
- Druhá část teoretického úvodu se zabývá odborným výkladem základů počítačové grafiky, nejvýznamnějšími pojmy dále rozvinutými a vysvětlenými v následujících kapitolách. Zmiňuji se zde také o barevných modelech a prostorech, které jsou stěžejním pojmem v mé práci.
- Třetí kapitola se týká rozboru práce, zaměřuji se zde na několik detailů, které se týkají spíše realizace zadání a implementace algoritmů, případně popisu funkce a významu daného appletu jako takového. Rovněž uvádím úryvky ze zdrojových kódů jednotlivých programů a snažím se o stručná objasnění konkrétních úseků mého programu, zejména jejich funkci v daném appletu.

# 1 Teoretický úvod

## 1.1 Úvod k programovací části práce

Každý z nás pracuje prakticky denně s rozličnými aplikacemi běžících v moderních internetových prohlížečích. A protože právě podpora multimediálního obsahu na webu za dlouhá léta prošla a stále prochází významným vývojem, nebylo nikterak špatným nápadem podobné téma práce zpracovat s využitím výhradně webových technologií a jazyků užívaných na webu. Jejich nespornou výhodou je především poměrně jednoduché použití, díky spustitelnosti na prakticky libovolném zařízení a nezávislosti na použitém operačním systému mohou být takové programy přístupné všem běžným uživatelům s internetovým připojením. Přitom také odpadá nutnost instalace, případně konfigurace jakéhokoli přídatného softwaru nebo jeho závislých součástí.

V několika následujících odstavcích bych Vám rád přiblížil základní technologie a jazyky, s jakými se v dnešním světě na webu setkáte nejčastěji a na kterých je také postavena má bakalářská práce.

### 1.1.1 HTML

HTML pochází z anglického názvu Hypertext Markup Language a jedná se o značkovací jazyk hojně užívaný pro tvorbu webových stránek. Raná verze vznikla již roku 1990 a pracovali na ní Tim Berners-Lee a Robert Cailliau v rámci projektu WWW pro organizaci CERN (Evropské centrum jaderného výzkumu)[16]. Cílem tohoto projektu bylo vyřešit problém se sdílením informací ve velké instituci. První verze HTML tak zdaleka neobsahovala tolik funkcí jako její současná verze a umožňovala prakticky pouze členění textu do logických úrovní, užití zvýraznění textu, vložení obrázků a provázání stránek s dalšími pomocí tzv. hypertextových odkazů.

HTML často bývá chybně označován jako programovací jazyk, avšak HTML pouze popisuje stránku z hlediska velmi zjednodušené vizuální podoby, nedává nám prostor implementovat úplný design či vlastní funkcionalitu námi umístěných komponent na webové stránce. K tomu slouží v současné době kaskádové styly CSS (pro ladění vzhledu) a JavaScript (pro implementaci interaktivních prvků a scriptů).

Jazyk HTML je charakterizován množinou značek (tzv. tagů) a jejich vlastností, ty jsou součástí každého zdrojového kódu webové stránky. Značky jsou obvykle párové, přičemž obsah vložený mezi ně definuje přímo konkrétní umístění v dokumentu (např. hlavička, tělo stránky), nebo jejich vlastnost (odstavec, nadpis určité úrovně apod.). Validní HTML dokument by tak měl obsahovat specifikaci standardu pomocí tagu `!doctype`, párové značky pro začátek a konec HTML dokumentu, hlavičku

a tělo stránky.

Současná verze HTML5, jehož finální podoba byla ustanovena v roce 2014, přináší podstatné změny, mezi nejvýznamnější patří přímá podpora přehrávání multimédií v prohlížeči a podpora pro aplikace, jež pro svůj běh nevyžadují připojení k Internetu[17].

### 1.1.2 JavaScript

Javascript je interpretovaný, objektově orientovaný programovací jazyk, hojně užívaný zejména pro tvorbu scriptů ovládajících interaktivní a multimediální prvky na webové stránce. Vznikl již v roce 1995 a jeho zakladatelem je Brendan Eich z tehdejší společnosti Netscape [18]. Od jiných interpretovaných programovacích jazyků užívaných na webu se Javascript liší především tím, že nejčastěji běží na straně klienta, nikoli serveru, jako je tomu např. u jazyků PHP a ASP.

Ve své bakalářské práci jsem pro naprogramování veškerých požadovaných appletů vycházel prakticky z čistého JavaScriptu, následující ukázka kódu demonstruje vlastní implementaci metody `getImageData()`, jíž jsem použil v appletu pro převod barevného obrazu na šedotónový pro práci s obrazem načteným pomocí HTML elementu `Canvas` a to na úrovni jednotlivých pixelů [13]. Spolu s názornými ukázkami pro použití této metody snadno dohledatelnými na webu a dále citovanými v závěru práce, mi tato metoda umožnila prakticky jednoduše implementovat algoritmy pro převod obrazů do stupňů šedi.

### 1.1.3 CSS

Kaskádové styly (z angl. názvu `Cascading Style Sheets`) jsou jazykem hojně využívaným ke stylizaci a nastavování vlastností a atributů jednotlivých prvků HTML, ale i XHTML a XML dokumentu. Kaskádové styly popisují způsob zobrazení jednotlivých prvků na stránce, dokáží tedy např. nastavit barvu, způsob odsazení či obtékání prvku na stránce apod. V současné podobě rovněž umožňují vytváření efektů vyvolaných událostmi, jakými je třeba barevný přechod po najetí kurzoru na konkrétní prvek či skrytí/zobrazení elementu na stránce.

O vznik tohoto jazyka se zasloužilo standardizační mezinárodní konsorcium `World Wide Web Consortium` (ve zkratce `W3C`) již v roce 1996, zejména pak jeho prvotní návrhář `Håkon Wium Lie` [19]. K výrazné revizi tohoto jazyka došlo v roce 2011 s nástupem `CSS 2.1`, jejíž cílem mělo být oddělení vzhledu dokumentu od jeho struktury a obsahu, což do této doby zastupoval a umožňoval samotný jazyk `HTML`. S nástupem `CSS3` je možno využít nejen všech výhod a možností jazyka `HTML5`, ale i nových funkcí, jakými jsou např. animování konkrétních elementů, nastavení míry průhlednosti či `2D/3D` transformace.

## 1.2 Základní pojmy v počítačové grafice

### 1.2.1 Světlo, jeho povaha a barva

Pojem světla zasahuje do několika oblastí fyziky. Velmi závisí na tom, z jakého fyzikálního hlediska na něj nahlížíme a jakým odpovídajícím způsobem jej následně popisujeme. Světlo má povahu částicovou i vlnovou. Světlo nejčastěji definujeme jako elektromagnetické vlnění o frekvenci zhruba  $3,9 \cdot 10^{14}$  do  $7,9 \cdot 10^{14}$  Hz [8]. V oblasti viditelného spektra světla tato frekvence odpovídá vlnovým délkám v rozsahu od konce ultrafialového pásma (390 nm) po začátek pásma infračerveného (760 nm). Vlnové délky, které se nachází mimo tento rozsah lidské oko není schopno vnímat. Se světlem se velmi úzce pojí barva. Podobně jako na světlo samotné, i na barvu můžeme nahlížet z vícera hledisek. Barvou popisujeme např. nejen vlastnost konkrétního objektu nebo světla, ale také jí rozumíme vnímání z pohledu samotného pozorovatele [1]. Barva tedy také popisuje vjem vznikající na sítnici lidského oka na niž dopadá elektromagnetické záření o vlnových délkách nacházejících se ve spektru viditelného světla.

Tab. 1.1: Tabulka rozdělení barevného spektra světla. Převzato z [27].

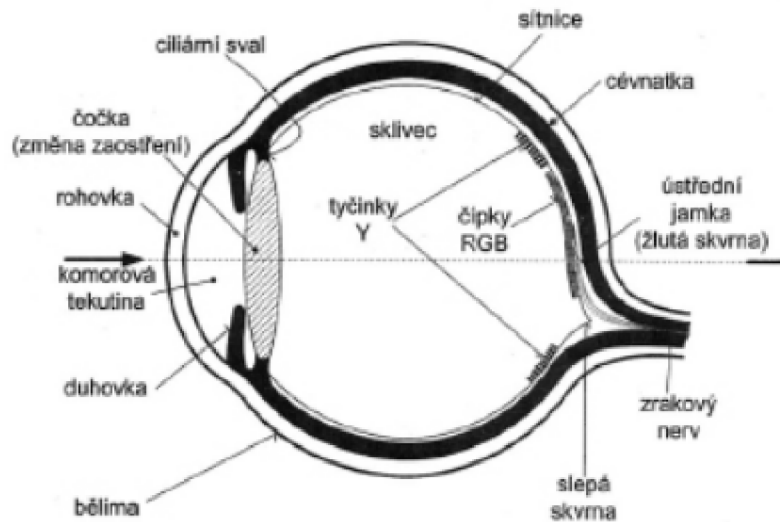
Barva	Rozsah vlnových délek	Rozsah frekvencí
červená	625–800 nm	480–375 THz
oranžová	590–625 nm	510–480 THz
žlutá	565–590 nm	530–510 THz
zelená	520–565 nm	580–530 THz
tyrkysová (azurová)	500–520 nm	600–580 THz
modrá	430–500 nm	700–600 THz
fialová (purpurová, nachová)	400–430 nm	750–700 THz

Abychom si více ujasnili zákonitosti týkající se barvy a jejího vnímání, prohlédněme si detailněji obrázek stavby lidského oka:

Oko lidského jedince je tvořeno velkým množstvím světlocitlivých buněk - tzv. čípků [27]. Existují čípky celkem tří druhů, přičemž každý z nich se specializuje na vnímání jedné ze tří základních barev - zelené, červené a modré. Jinými slovy, každý druh čípku je tedy citlivý na jinou vlnovou délku odpovídající těmto barvám. Nelze však říci, že by čípek reagoval na snímání jen jedné konkrétní barvy, pouze platí, že je pro vnímání určité frekvence světla nejcitlivější.

V případě, že dopadající elektromagnetické záření zaregistrují více druhů čípků, dochází k tzv. skládání spektrálních barev, což nám umožňuje od sebe odlišovat





Obr. 1.1: Stavba lidského oka. Obr. převzat z [5]



Obr. 1.2: Citlivost čípků na světlo podle vlnové délky. Obr. převzat z [8]

různé druhy barev a barevných odstínů. Takovým příkladem je např. vyhodnocení bílé barvy. Naopak černou barvu pozorujeme, jestliže na dopadající záření nezareaguje žádný z těchto druhů čípků. [8]

Barvu objektů můžeme posoudit na základě jejich fyzikálních vlastností, velmi však také závisí na skutečnosti, jak je vnímá samotný pozorovatel. Z fyzikálního hlediska je totiž povrch tělesa té barvy, jejíž elektromagnetické záření vyzařuje, případně odráží. Dojde-li k odrazu, v takovém případě je stěžejním hlediskem struktura spektra dopadajícího světla a také samotné zastoupení složek spektra takového světla, které jsou povrchem odraženy a které naopak pohlceny, včetně údaje o velikosti příslušné intenzity. [27]

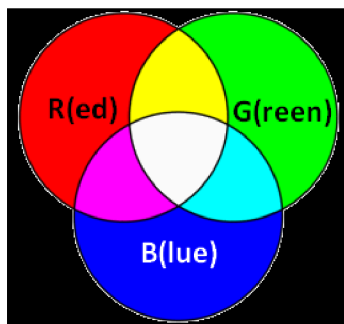
## 1.2.2 Barevné prostory, modely a jejich vzájemné převody

Barevný model nám slouží k popisu barev, vychází z předpokladu, že barva je vždy reprezentována číselnou hodnotou [1]. V nejčastějším případě je tato matematická struktura popsána uspořádanou trojicí nebo čtveřicí hodnot barevných složek, typickými zástupci jsou např. barevné modely RGB nebo CMY(K). Takto jsou definovány pouze základní barvy, z nichž barvy odvozené pak můžeme získat smícháním jednotlivých barevných složek v určitém poměru. Podle způsobu, jakým jsou barvy smíchány, můžeme dále rozdělit barevný model na aditivní (součtový) a subtraktivní (rozdílový)[20]. Hovoříme-li však o množině barev definované určitým barevným modelem, která popisuje reprezentaci jednotlivých složek, pak se jedná již o barevný prostor. Nejvýznamnější vlastnost, kterou se barevný prostor liší od barevného modelu je ta, že prostor obsahuje standardizované odstíny základních barev. U barevných prostorů též definujeme tzv. gamut, jenž by se dal zjednodušeně vyjádřit jako rozsah barev, který je schopen daný barevný prostor pojmut. Typickými příklady standardizovaných barevných prostorů jsou např. prostory sRGB nebo AdobeRGB od známé společnosti Adobe. Platí však, že každé zobrazovací či snímací zařízení (monitor, projektor či skener) má svůj vlastní gamut zobrazitelných barev. Barvy, které se nachází mimo tento rozsah, není zařízení schopno zobrazit věrně a dochází zde k jisté odchylce, kdy je zobrazena nejbližší podobná barva nacházející se v gamutu tohoto zařízení. Z této skutečnosti tedy vyplývá, že je třeba u barevných modelů a prostorů rozlišovat závislost, případně nezávislost na konkrétním zařízení. Proto je tedy barevný model RGB považován za závislý na zařízení a jakákoli dvě rozdílná zařízení mohou zobrazit odlišný výsledek odchylovající se od našeho originálního obrazu. Existují však také další prostory, jež nejsou závislé na konkrétním zařízení. Následně se budu zabývat zejména barevnými prostory XYZ (CIEXYZ) a LAB, které berou v úvahu především subjektivitu lidského vnímání. Zatímco první výše zmíněný barevný prostor, tedy prostor sRGB, byl původně zamýšlen pro vzájemnou kompatibilitu zobrazení napříč levnějšími zařízeními, nepokrývá zdaleka tak velké množství barev jako druhý zmíněný.

## 1.2.3 RGB a CMY(K)

Barevný model RGB patří, jak již bylo zmíněno, mezi modely závislé na konkrétním zařízení. Je tvořen 3 základními barevnými složkami: červenou (Red), zelenou (Green) a modrou (Blue)[1]. Tyto barevné složky se přidávají do černé barvy, proto se v grafickém znázornění tohoto modelu setkáme se 3 kruhy, které představují jednotlivé barevné složky R, G a B, umístěné na černém podkladu. Protože je tento model založen na aditivním způsobu míchání, dochází při vzájemném míchání jednotlivých kanálů v místě průniku těchto kruhů ke vzniku světlejší barvy, neboť se

složky jednotlivých barev sčítají a vytvářejí tak světlo vyšší intenzity. Z toho vyplývá, že při smíchání všech tří základních barevných kanálů o maximálních intenzitách světla vznikne bílá barva. Jelikož je tento barevný model založen na vyzařování světla, využívá se především u zobrazovacích zařízení, jakými jsou např. monitory či projektory, na stejném principu však fungují i skenery. S tímto barevným modelem se rovněž můžeme setkat i v jeho alternativní variantě RGBA, kde poslední poslední písmeno označuje tzv. Alpha kanál, stanovující míru průhlednosti.



Obr. 1.3: Schéma aditivního modelu RGB.

Prakticky opačným barevným modelem k modelu RGB je model CMY, případně CMYK [21]. V základu rovněž sestává ze 3 barev, konkrétně azurové (Cyan), purpurové (Magenta) a žluté (Yellow). Jelikož je tento model založen na principu subtraktivního míchání barev, kdy při smíchání vzniká nová, tmavší barva, je tento model výhodný zejména pro reprodukční zařízení, jakými jsou např. tiskárny. U tiskové techniky se nejčastěji setkáme s modelem CMYK, neboť obsahuje také tzv. klíčovou barvu, a to černou (blacK, ale i Key). Ta je z hlediska tisku nejekonomičtější, neboť smíchání tří výše zmíněných barev nám v praxi nezaručí stoprocentní černou a rovněž je (např. pro tisk na novinový papír) vhodnější použití jedné cartridge, než současný soutisk více barev[1].

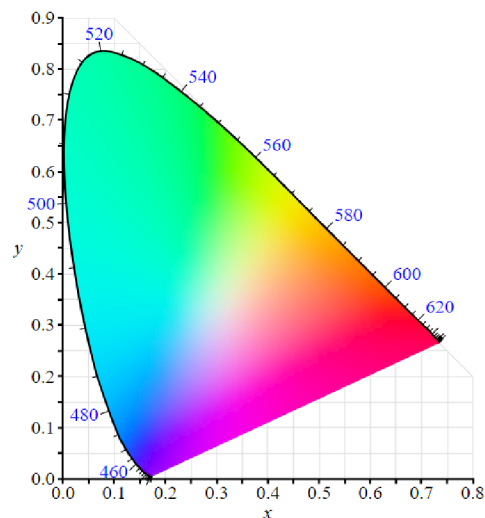
#### 1.2.4 CIEXYZ

Barevný prostor CIEXYZ spadá do kategorie prostorů nezávislých na konkrétním zařízení. Byl založen na základě výzkumů organizací CIE jako model, v němž je kladen důraz na subjektivitu lidského vnímání barvy, kdy člověk dokáže rozpoznat daleko více rozdíl v jasů, než rozdíl v odstínu barvy[1]. Proto tento prostor záměrně odděluje jasovou složku od vlastních barev, které se označují v tomto prostoru jako parametry  $x$  a  $y$ . Tyto parametry je možné vypočítat ze tří tzv. trichromatických složek  $X$ ,  $Y$  a  $Z$ , jež stanovují množství jednotlivých měrných světelných nezbytných pro simulaci konkrétní barvy.

## Srovnání barevných modelů RGB a CMY



Obr. 1.4: Srovnání RGB a CMY modelu.

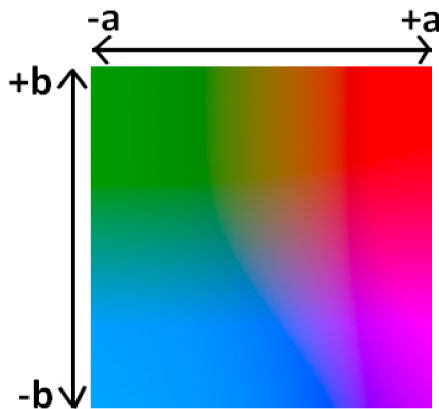


Obr. 1.5: Chromatický diagram CIE 1931 - jedná se o křivku dle standardu CIE, jež ohraničuje barvy viditelného spektra. Obrázek převzat z [28].

### 1.2.5 $L^*a^*b$ (CIELAB)

Barevný prostor  $L^*a^*b$ , též zvaný CIELAB, neboť byl definován stejnou organizací podílející se na vzniku barevného prostoru CIEXYZ, je rovněž prostor nezávislý na zařízení a prakticky z předešlého modelu CIEXYZ i vychází[23]. Z toho tedy vyplývá, že i zde je brána v úvahu subjektivita lidského vnímání barev. V základě je tento prostor založen na modelu tří os definujících konkrétní barvu. Konkrétně se jedná o osy charakterizující[22]:

- Světlost (Lightness), a to v rozsahu hodnot 0 – 100.
- Barvu v rozmezí zelené až červené (od -a do +a)
- Barvu v rozmezí modré až žluté (od -b do +b)



Obr. 1.6: Osy a, b prostoru Lab při 50% světlosti.

Na základě znalosti těchto parametrů dokážeme podle následujícího vztahu vypočítat tzv. barevnou odchylku delta E, jež nám určuje vzdálenost dvou barev při vzájemném srovnání v prostoru Lab.

$$\Delta E = \sqrt{(L - \bar{L})^2 + (a - \bar{a})^2 + (b - \bar{b})^2} \quad (1.1)$$

Tento prostor především řeší zásadní problém barevného prostoru CIEXYZ, kde kolorimetrické vzdálenosti mezi jednotlivými barvami neodpovídají vnímanému barevnému rozdílu.

Zmíněný problém názorně demonstruje předchozí obrázek, kdy rozdíl mezi zelenou a zelenavě-žlutou je poměrně velký, zatímco rozlišovací vzdálenost mezi modrou a červenou je docela malá.

Výhodu LAB prostoru, oproti prostoru RGB, oceníme zejména u nástrojů pro zpracování obrazů a editorů fotografií, kde je barevný prostor Lab využíván např. pro doostření obrazů, odstraňování artefaktů ze snímků v JPEG formátu, nebo snímků z digitálních fotoaparátů a skenerů.

## 1.2.6 Převod z RGB do Lab

Převod z RGB do barevného prostoru LAB nelze provést přímo, ale sestává ze dvou fází. Nejdříve je třeba provést přepočítání z RGB, potažmo sRGB prostoru, který je pro tyto účely využíván nejčastěji, do prostoru CIEXYZ s využitím trichromatických složek, bílého bodu D65 ve středu gamutu sRGB a transformačních funkcí. Teprve

poté můžeme hodnoty vypočtených složek převádět z prostoru XYZ do LAB. Pro přetváření lineárních hodnot do sRGB prostoru je rovněž nutné počítat s hodnotou gamma. První fázi převodu RGB do XYZ modelu znázorňují následující rovnice (ukázka pro matici o velikosti 3x3):

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = [M] \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

, kde platí:

$$[M] = \begin{bmatrix} S_r X_r & S_g X_g & S_b X_b \\ S_r Y_r & S_g Y_g & S_b Y_b \\ S_r Z_r & S_g Z_g & S_b Z_b \end{bmatrix}$$

$$\begin{array}{ll} X_r = x_r/y_r & (x_r, x_r), (x_g, y_g), (x_b, y_b) \\ Y_r = 1 & \text{představují tzv. } \mathbf{\textit{souřadnice}} \\ Z_r = (1 - x_r - y_r)/y_r & \mathbf{\textit{chromatičnosti}} \\ X_g = x_g/y_g & \\ Y_g = 1 & (X_w, Y_w, Z_w) \\ Z_g = (1 - x_g - y_g)/y_g & \text{představují hodnoty tzv.} \\ X_b = x_b/y_b & \mathbf{\textit{referenční bílé}} \\ Y_b = 1 & \\ Z_b = (1 - x_b - y_b)/y_b & \text{(sRGB prostor využívá bodu D65)} \end{array}$$

$$\begin{bmatrix} S_r \\ S_g \\ S_b \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix}^{-1} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$$

Obr. 1.7: Přepočtení hodnot RGB do systému XYZ[35].

Bohužel se mi ani v rámci této bakalářské práce nepodařilo dohledat přesný význam jednotlivých užitých konstant ve vzorcích ani dostatečně jednoduše vysvětlený popis převodu, proto jsem v této části práce vycházel především ze zdrojového kódu appletu mého předchůdce Jindřicha Hlavatého[5][12], snažil jsem se tedy ověřit správnost těchto výpočtů se zdrojovými kódy dalších autorů snadno dohledatelných na internetu (př.[30][33][34]). Odkazy na tyto zdrojové kódy jsou uvedeny v referen-

Musí být předem stanoven referenční bílý bod  $X_r, Y_r, Z_r$

$$L = 116f_y - 16$$

$$a = 500(f_x - f_y)$$

$$b = 200(f_y - f_z)$$

, kde platí:

$$f_x = \begin{cases} \sqrt[3]{x_r} & \text{pokud } x_r > \epsilon \\ \frac{\kappa x_r + 16}{116} & \text{v ostatních případech} \end{cases}$$

$$f_y = \begin{cases} \sqrt[3]{y_r} & \text{pokud } y_r > \epsilon \\ \frac{\kappa y_r + 16}{116} & \text{v ostatních případech} \end{cases}$$

$$f_z = \begin{cases} \sqrt[3]{z_r} & \text{pokud } z_r > \epsilon \\ \frac{\kappa z_r + 16}{116} & \text{v ostatních případech} \end{cases}$$

$$x_r = \frac{X}{X_r}$$

$$y_r = \frac{Y}{Y_r}$$

$$z_r = \frac{Z}{Z_r}$$

$$\epsilon = \begin{cases} 0.008856 & \text{Actual CIE standard} \\ 216/24389 & \text{Odvozené vyjádření} \end{cases}$$

$$\kappa = \begin{cases} 903.3 & \text{Actual CIE standard} \\ 24389/27 & \text{Odvozené vyjádření} \end{cases}$$

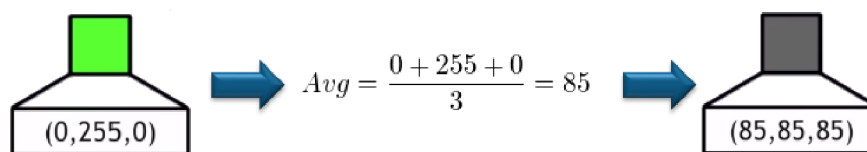
Obr. 1.8: Přepočítání hodnot XYZ do prostoru Lab, při známých hodnotách referenční bílé ( $X_r, Y_r, Z_r$ ) [36].

cích na konci práce.

### 1.2.7 Průměrování RGB kanálů

Tato metoda je prakticky založena na principu, kdy je každému barevnému kanálu R, G i B přiřazena stejná váha, odpovídající jedné třetině z celkového součtu [15]. Uvažujeme-li, že každý pixel obrazu je vyjádřen trojicí hodnot reprezentující jednotlivé barevné kanály, postupně R, G a B v tomto pořadí. Z toho každý kanál představují v 8 bitovém vyjádření hodnoty v rozmezí 0 – 255, kdy hodnota 255 značí maximální intenzitu vyzářeného světla a při hodnotě 0 daný kanál naopak

žádné světlo nevyzařuje. V případě metody průměrování se provede součet třetinových hodnot jednotlivých barevných kanálů a vzniklá výsledná hodnota jasové složky se následně nakopíruje do všech tří kanálů pro daný pixel stejně.



Obr. 1.9: Princip průměrování RGB hodnot. Na všechny tři číselné pozice pixelu, představující jednotlivé barevné kanály R, G a B jsou dosazeny stejné hodnoty, tím pixel získá šedotónový odstín na rozmezí mezi černou - bílou.

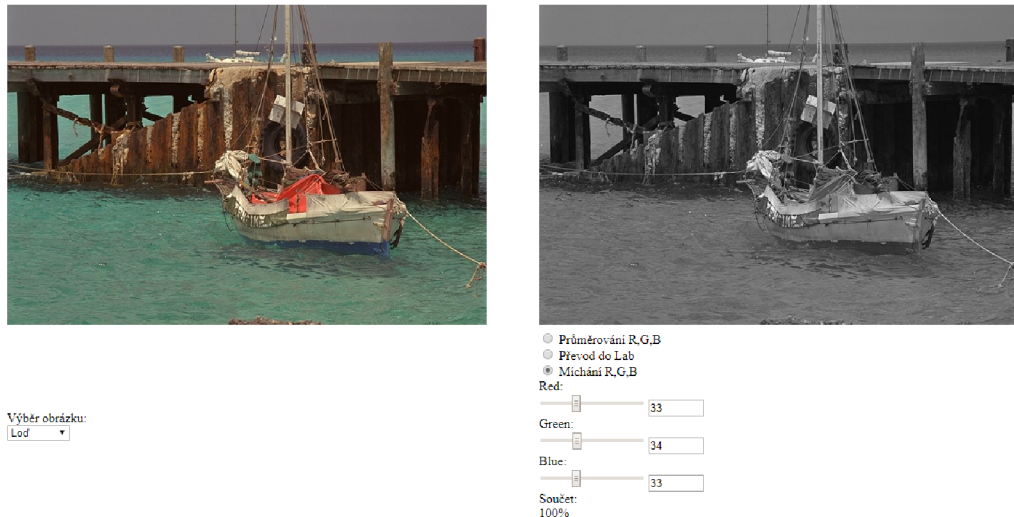
Z appletu pro převod barevného obrázku do stupňů šedi je při vybraném vstupním obrázku RGB modelu a aktivované metodě průměrování jasně vidět, že barva všech kruhů v místě jejich rozdílu je u všech barevných kanálů stejná.

## 1.2.8 Selektivní míchání RGB kanálů

Tato metoda funguje obdobně jako metoda průměrování s tím rozdílem, že zde má uživatel navíc možnost ručně si upravit hodnoty pro každý ze tří již zmíněných RGB kanálů. Ve výchozím stavu jsou hodnoty nastaveny velmi podobně jako v případě metody průměrování. K nastavení požadovaných hodnot může uživatel v grafickém rozhraní appletu využít buď posuvníky, které se automaticky zobrazí při zvolení této metody, nebo lze požadovanou hodnotu vepsat do textových polí, umístěných za těmito posuvníky. Výpočet ve zdrojovém kódu této metody je založen na vynásobení hodnoty pro každý ze tří kanálů s hodnotou vycházející z aktuální polohy příslušného slideru a jejich vzájemným součtem[5]. Protože hodnota vyčtená z nastavení slideru vychází z procentuálního vyjádření v grafickém rozhraní a nachází se tedy v rozsahu hodnot od nuly do jedné, bylo též třeba přepočíst hodnoty jednotlivých barevných kanálů, vyjádřených původně osmi bity, na hodnoty v rozsahu 0 – 1. Hodnota výsledného vypočteného jasu je opět nakopírována do všech tří hodnot barevných kanálů R,G a B.

V průběhu tvoření ovládacích prvků této metody jsem si kladl za cíl, provázat ovládací prvky mezi sebou tak, aby např. při nastavení požadované hodnoty posuvníkem došlo k aktualizaci hodnoty v poli a naopak. Bohužel, zde narážím na nekompatibilitu se staršími verzemi Internet Exploreru, kde právě tyto ovládací prvky nefungují zcela korektně, proto doporučuji pro používání appletu využívat aktuální verze prohlížečů, nejlépe pak přímo prohlížeče Chrome a Firefox.





Obr. 1.10: Ukázka vytvořeného appletu, který znázorňuje některé z možností převodu barevného obrázku na šedotónový.

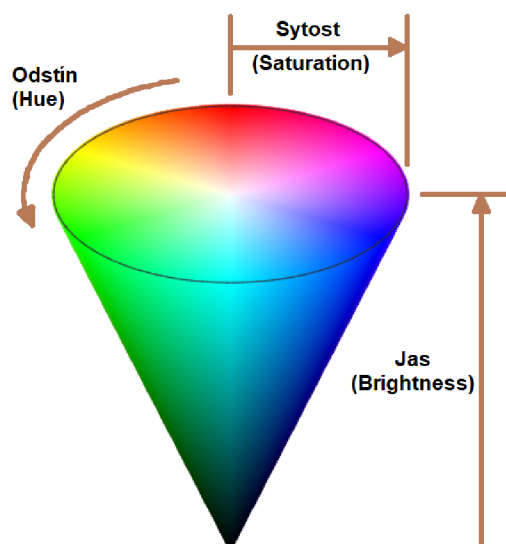
### 1.2.9 HSB

Barevný model HSB je alternativním barevným modelem k výše zmíněným modelům. Vznikl s cílem klást důraz zejména na názornost, intuici lidského vnímání barev a barevnou představivost člověka[1]. Bylo prokázáno, že lidé mají daleko větší představu o vlastnostech konkrétní barvy, než o výsledku vzájemného smíchání většího množství barev. Tento barevný model je, podobně jako model RGB, reprezentován třemi barevnými složkami, nyní se ale nejedná o poměr zastoupení konkrétní barevné složky, ale o parametry definující barevný odstín (Hue), sytost konkrétní barvy (Saturation) a hodnotu jasu (Brightness)[25]. V některé literatuře se tento barevný model vyskytuje i pod názvem HSV, kdy poslední písmeno značí též jasovou složku nazvanou Value, jedná se však prakticky o totožný barevný model.

Bližší význam parametrů HSB/HSV modelu:

- Odstín - tento parametr představuje úhel naměřený na barevném kotouči s rozsahem měřitelných hodnot 0 - 360°, kdy dle dohody prochází nultý a 360tý stupeň 100% červenou, 120tý stupeň zelenou a 240° modrou barvou. Jelikož je barevný odstín představován barevným kruhem, který vychází i vstupuje (po objetí kruhu) do stejné barvy, větší hodnoty odstínu jak 360st postrádají smysl, neboť představují nový oběh kruhu.
- Sytost - jedná se o zastoupení konkrétní barvy ve vztahu k šedé, vyjadřuje se v procentech v rozmezí 0% (čistě šedá) po 100% (plně sytá barva, bez zastoupení šedé).
- Hodnota jasu (B/V) - vyjadřuje poměrnou světlost či tmavost konkrétní barvy.

Graficky se barevný model HSB znázorňuje jako rotační kužel, v němž jsou zastoupeny všechny barvy, které lze vyjádřit pomocí výše zmíněných parametrů, tedy odstínu, sytosti a stupně jasu. Následující obrázek popisuje, jakým způsobem je možné tyto parametry z grafického modelu HSB modelu vyčíst. Rotací kolem středu podstavy lze tak snadno získat hodnotu odstínu, poloměr od středu podstavy k jejím okrajům představuje parametr sytosti (saturace) a vzdálenost od vrcholu kuželu k podstavě pak značí nastavený jas.



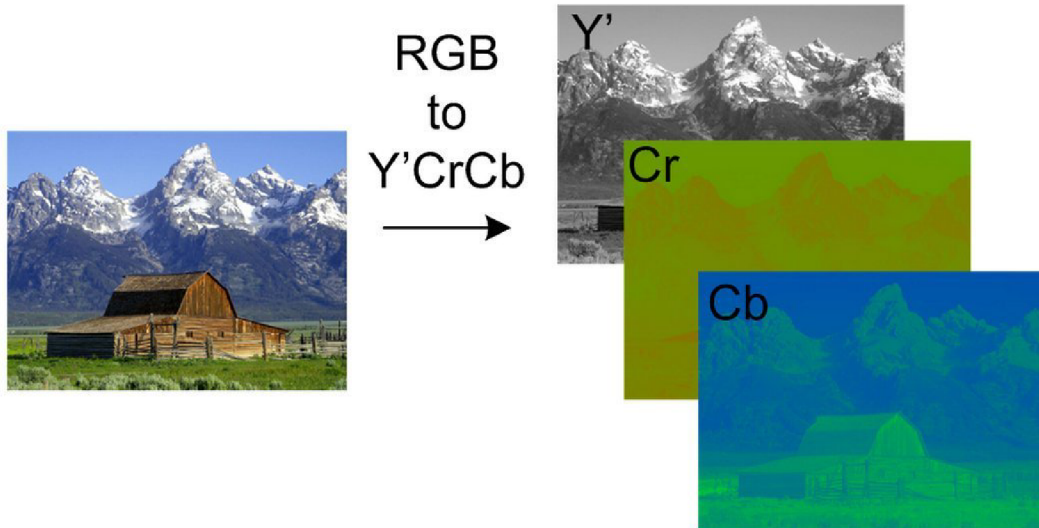
Obr. 1.11: Grafické znázornění modelu HSB pomocí rotačního kuželu.

### 1.2.10 YCbCr

S tímto barevným modelem se setkáme spíše u televizní techniky, zpracování videa či digitálních fotografií. Základem je zde opět složka jasová (Y), kterou doplňují dvě barvosné monochromatické složky Cb (Blue - Modrá) a Cr (Red - Červená). Toto oddělení jasové složky od barvosných má svůj význam - v dobách minulých tak bylo dosaženo vzájemné kompatibility barevného televizního vysílání se staršími černobílými televizory. Navíc tak mohl být jasový signál vysílán ve výrazně větším rozlišení a mohla být pro něj rezervována i větší šířka pásma, než pro barvosné složky, jejichž zastoupení není ve výsledném obrazu prioritní. Existuje více předpisů, v jakých poměrech mohou být jednotlivé složky do celkového obrazu zakomponovány, jedno z nejznámějších a nejuniverzálnějších rozvržení vychází ze standardu ITU-T a stanovuje poměry následovně:

$$\begin{bmatrix} Y' \\ C_B \\ C_R \end{bmatrix} = \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ -0,168736 & -0,331264 & 0,5 \\ 0,5 & -0,418688 & -0,081312 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \quad (1.2)$$

Obr. 1.12: YCbCr - poměr složek dle ITU-T



Obr. 1.13: Ukázka převodu fotografie z RGB modelu do prostoru YCbCr. Obrázek převzat z [26].

### 1.3 Metody umělého obarvení obrazu

V následující podkapitole popisují možnosti a způsoby umělého obarvení obrazu. Těchto principů se dnes v praxi hojně využívá např. při termovizním měření tepelného záření u termometrů, avšak termovizní snímky dnes naleznou využití také např. v armádě v naváděcích systémech raket, v civilním nasazení pak k detekci míst, ve kterých dochází k úniku tepla z budov. Hlavním důvodem k takovému využití obarvení původně šedotónových snímků je zejména jejich lepší čitelnost, kdy lidský pozorovatel provádějící diagnostiku takto obarveného obrazu dokáže v obrazu rozpoznat daleko více detailů, oproti případu, kdy by měl k dispozici pouze šedotónový snímek ve stupních šedi [3].

Možností, jakými můžeme dosáhnout převodu šedotónového obrazu v úrovních jasu do obrazu o nepravých barvách je skutečně mnoho, mezi nejpoužívanější způsoby v praxi se řadí např.[29]:

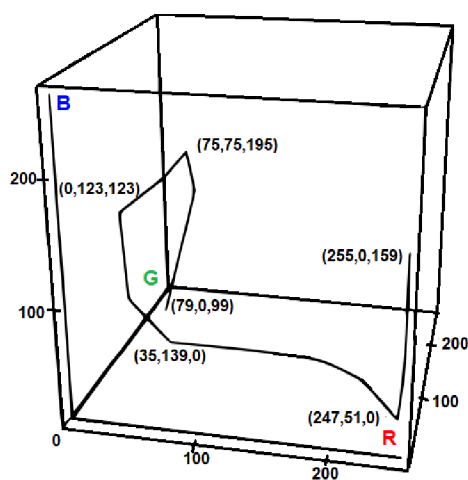
- využití jevu pseudosolarizace u fotografických výstupů

- hardwarové řešení pomocí programovatelných logických obvodů
- případně softwarová řešení, která mohou být založena na využití rozličných metod, případně jejich kombinací

V této práci, respektive v následujících odstavcích se budu zabývat primárně softwarovým řešením, neboť právě tyto metody jsou využity i v mnou vytvořeném appletu.

### 1.3.1 Metoda parametrické křivky

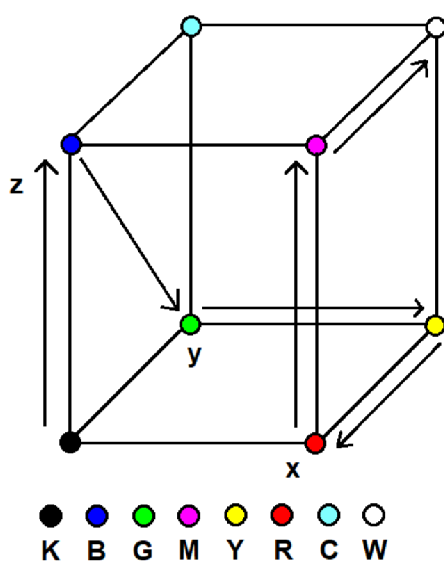
Metoda parametrické křivky vychází z předpokladu, že RGB model vyjádříme schematicky jako krychli reprezentující v jednotlivých svých rozích všechny základní barvy RGB a CMYK modelu, tedy barvy vyjádřené osmi bity o hodnotách buď 0 nebo 255 a to v jakékoli kombinaci pro jednotlivé barevné kanály [3]. Pro obraz, který je představován 24 bity je základem následující schematické znázornění krychle, neboť 24 rohů je reprezentováno počtem možných kombinací  $2^3$  právě proto, že barevný model RGB tvoří tři základní barevné složky. V takto definovaném prostoru je potom možné popsat proces barvení křivkou, jež bude vytvořena z tzv. ekvidistantně rozložených bodů. Souřadnice pro každý konkrétní bod se poté následně vztahují ke stupňům šedi, kterými je tvořen původní obrázek. Spirálovitý tvar této Beziérový křivky je však nutný pro zachování informace o kontinuálních změnách v jasů. Zároveň musí platit, že oba konce této křivky sahají až do jednotlivých rohů tohoto modelu, které značí bílou a černou barvu.



Obr. 1.14: Schematické znázornění RGB krychlového prostoru proloženého parametrickou křivkou.

### 1.3.2 Metoda Schrödingerovy krychle

Metoda Schrödingerovy krychle, stejně jako metoda parametrických křivek popísaná v předchozí podkapitole, pracuje s krychlovým RGB modelem. Hlavním rozdílem oproti způsobu popsaném v předchozí podkapitole je skutečnost, že rozmezí jasových hodnot na vstupu je rozčleněno do šesti skupin, kdy trojice výstupních RGB složek pro každý segment buďto roste, klesá nebo stagnuje na své minimální, případně maximální hodnotě [3]. Tento způsob chování je možné v krychli znázornit několika možnými směry: buď směrem z černé do modré, z modré do zelené, ze zelené do žluté, ze žluté do červené, z červené do purpurové a na závěr z purpurové do bílé barvy. Popsaný způsob jasně vysvětluje následující obrázek i tabulka Schrödingerova principu tvorby barev.



Obr. 1.15: Princip metody Schrödingerovy barvicí krychle.

### 1.3.3 Metoda barevné palety

Základem pro tuto metodu je vytvoření většího množství barvicích palet o maticovém uspořádání. Tato matice je trojřádková, neboť každý tento řádek matice odpovídá jednotlivé barevné složce z RGB modelu, jednotlivé sloupce poté představují vstupní hodnoty pro jasovou složku[3]. Zpracování obrazu probíhá pixel za pixelem a algoritmus programu zajišťuje vyhledání a zejména výběr příslušné barevné složky ve zvolené barevné paletě, a to dle zadaných vstupních hodnot, které slouží pro sestavení výstupní barvy. Tato metoda patří, ze způsobů výše uvedených, mezi jednodušší a byla použita i v mém appletu.

Tab. 1.2: Tabulka přiřazení barevných složek v metodě Schrödingerovy krychle. Sloupec rozsah jasů představuje jasové hodnoty pixelů vstupního obrazu, sloupce R, G, B charakterizují změny barevných složek během rozsahu (rostoucí nebo klesající průběh), poslední sloupec vyjadřuje finální barvu na konci rozsahu (v rozích Schrödingerovy pseudobarvicí krychle). Převzato z [29].

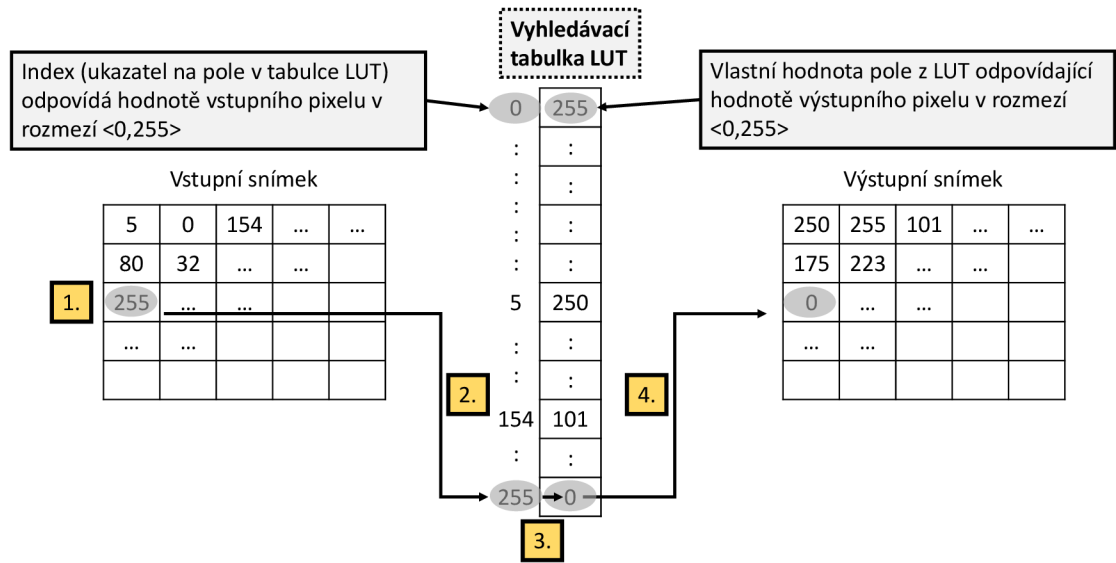
(Y) Rozsah jasů	(R) Červená	(G) Zelená	(B) Modrá	Výsledná barva
0	0	0	0	Černá
1 - 42	0	0	↗	Modrá
43 - 84	0	↗	↘	Zelená
85 - 128	↗	max	0	Žlutá
129 - 170	max	↘	0	Červená
171 - 212	max	0	↗	Růžová
213 - 255	max	↗	max	Bílá

Schematicky bychom tuhle metodu mohli vyjádřit pomocí tzv. převodní Look Up tabulky (v anglické literatuře hojně označované jako LUT). Tato tabulka charakterizuje závislost mezi pixely vstupního a výstupního obrazu. Funguje na principu vyhledávací tabulky, ve které jasová hodnota pixelu obsaženého ve vstupním monochromatickém obraze odpovídá řádku tabulky (potažmo paměťové adrese) a hodnota tohoto konkrétního řádku tabulky, respektive hodnota z odpovídající adresy v paměťovém prostoru, charakterizuje jasovou složku výstupního obrazu. [8]

Metoda tedy funguje poměrně jednoduše a z hlediska jejího principu je velmi dobře čitelná a srozumitelná. Jelikož je každý digitální snímek tvořen miniaturními obrazovými body a každý z těchto pixelů nabývá jisté číselné hodnoty, celkový snímek pak představuje číselná matice, jejíž rozměry jsou určeny rozměry právě tohoto celkového obrazu. Uvažujeme-li matici šedotónového obrazu, pak tato matice bude pouze jednorozměrná stupnice odstínů šedi v rozsahu hodnot 0 - 255.

V případě barevného obrazu je ale situace poněkud odlišná. Takový obraz totiž charakterizuje trojice matic, kdy každá z nich zastupuje jednu ze základních barevných složek RGB. Podle obsahu jednotlivých matic tedy můžeme poměrně snadno vyčíst zastoupení těchto základních barevných složek v daném obrazovém bodu. Na takovém principu funguje i mnou vytvořený applet demonstrující principy umělého obarvení obrazu, kdy nejdříve je pro každý odstín šedi šedotónového snímku přiřazen odpovídající odstín barevný. Z toho však vyplývá, že nyní dochází k situaci, kdy z původní jednorozměrné matice představující odstíny šedi vzniká na základě konkrétní tabulky trojice matic představující barevný odstín. [8]

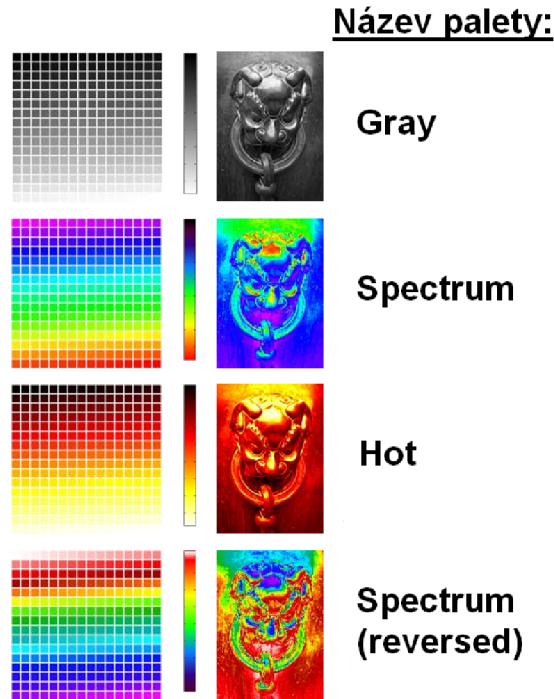




Obr. 1.16: Implementace LUT tabulky. Inspirace z [11]

Existence již dříve zmíněné Look-Up Table tabulky je základním předpokladem pro použití metody využívající barevnou paletu. Hlavní předností této metody je absence doplňkových parametrů, tudíž si tato metoda dokáže vystačit pouze s jasovou hodnotou pixelu ve vstupním snímku. K tomu je potřebná tabulka čítající takový počet řádků, z jakého počtu diskretních hodnot jasu dokáže sestávat obrazový bod snímku na vstupu. Za předpokladu, že definujeme počet bitů měřicího zařízení s pomocí proměnné  $n$ , tabulka pak obsahuje celkem  $2^n$  řádků. [8] Nejčastěji pracujeme s osmibitovými hodnotami, což nám umožňuje pracovat na 256 řádcích v tabulce s 256 stupni jasových hodnot. Tabulka obsahuje také tři sloupce, přičemž každý z těchto sloupců je vyhrazen pro jednu z barevných složek základu RGB. Vzhledem k použití osmibitového vyjádření hodnot může tabulka obsahovat pouze hodnoty z rozsahu 0-255. Rovněž platí, že data obsažená v tabulce musí být dopředu známa, jelikož je tabulka reprezentována 2D polem. Metoda pro svou funkci využívá výběru řádku z tabulky, který koresponduje s barvou pixelu na výstupu a to podle hodnoty jasové složky vstupního pixelu.

Data obsažená v tabulce mohou být poměrně různorodá, s čím souvisí také skutečnost, že je možné využít neomezené množství barevných palet. [8] V praktickém využití se však nesetkáme s přílišnou variabilitou zmíněné metody. Převážně proto ji řadíme převážně mezi způsoby, v důsledku jejichž využívání nedochází tak snadno k



Obr. 1.17: Ukázka vybraných variant barevných palet a jejich vlivu na podobu výstupního obrazu. Obr. převzaty z [10] s drobnými úpravami.

modifikacím barevné palety a následnému dopadu na podobu výsledného obrazu. A to zejména z toho důvodu, že k modifikaci barevné palety by bylo nezbytné udržovat větší množství LUT tabulek. Potíž ale nastává v situaci, kdy se pokusíme paletu co nejpřesněji definovat. Z důvodu přílišné časové náročnosti a obtížné přístupnosti samotnému uživateli je taková definice poměrně nesnadná.

### 1.3.4 Další metody umělého obarvení obrazu

Metod pro umělé obarvení obrazu, potažmo zobrazování snímku v nepravých barvách existuje celá řada, a jejich výčet by zdaleka nepokrýval rozsah praktické části mé práce a jejich podrobný výklad by až příliš odbočoval od původní zaměření práce. Proto mi dovoluňte, abych se zde pouze v krátkosti zmínil o některých dalších pokročilejších technikách využitelných v praxi. Základní principy některých metod se od sebe příliš neliší, což mnohdy usnadňuje aplikaci těchto metod v praxi.

#### Metoda jednoduché parametrické rovnice

Metoda byla poprvé popsána jako velmi dobře využitelná pro zpracování rentgenových snímků. Vyznačuje se svou typickou vlastností udržet progres jasové složky a



současně generuje nesčetné množství změn v barvách snímku, jež si kladou za cíl napomoci při vyhodnocování snímku lidským okem. Metoda využívá schematického zobrazení RGB prostoru v podobě pomyslné krychle, v níž se nachází tzv. ekvidistančně rozmístěné body tvořící křivku. Počet bodů se shoduje se stupni šedi ze vstupního obrazu, přičemž dochází k transformaci jednotlivých hodnot ze vstupní stupnice šedi na předem stanovenou barvu vyjádřitelnou složkami RGB modelu. Tato barva je tak určena koordináty samotného vzorku v krychlovém vyjádření barevného prostoru. V případě že využijeme modelu RGB, získáme výhodu v podobě použití těchto schémat coby LUT tabulek. Intenzitu jasové složky umožňuje regulovat hlavní diagonála v této RGB krychli, která zde představuje spojnicí mezi černou a bílou barvou, tedy šedotónovou stupnicí. Chceme-li tedy dosáhnout obarvení snímku a současně zachovat souvislé změny v jasové složce, pak je nezbytné a žádoucí, aby transformační křivka nabývala spirálovitého tvaru a rotovala okolo zmíněné diagonály.

### **Metody založené na jasové stupnici**

Tyto metody se hojně vyskytují v anglické literatuře pod názvem Gray-level based methods. Jejich charakteristickou vlastností je zejména požadavek na konstrukci transformační funkce, což znamená transformaci šedotónové stupnice, gradientů případně zcela odlišných šedotónových funkcí do libovolných barev. S využitím těchto technik dochází ke konverzi kompletní šedotónové stupnice na barevnou sadu přesně dle stanoveného pravidla, neuvažují se zde žádné doplňkové informace. Z toho důvodu tyto metody trpí zásadním neduhem, kdy dochází k případům, že je některá informace v obraze přehlédnuta. Metoda vychází z principu, kdy pro jednotlivé barevné složky RGB modelu vznikají tři na sobě nezávislé transformace. [8]

### **Metoda využívající barevných prostorů**

Jedná se o další z metod využitelných k umělému obarvení obrazu. Její principy využívají krom stupnice šedi také např. sytost, ale i jiné doplňující parametry. Barevný prostor je zde chápán jako interpretace zrakového vjemu člověka. Jeden z nejvýznamnějších a v praxi nejužívanějších barevných prostorů pro pseudobarvení snímku je prostor HSI (v anglické terminologii představují iniciály postupně barevný odstín (Hue), sytost neboli saturaci (Saturation) a intenzitu (Intensity)). Model HSI umožňuje postupnou konfiguraci svých parametrů, každý z nich lze nastavovat zvlášť. Získat tyto parametry můžeme z jasu vstupního šedotónového snímku. Metoda není příliš vhodná k použití u snímku s nízkou saturací barev, neboť u nich může docházet k nesprávné detekci barevného odstínu. [8]

### 1.3.5 Gamma korekce

Jedná se o nelineární transformaci sloužící k transformaci jasové stupnice. Této vlastnosti se využívá k úpravě jasové dynamiky snímku, přesněji ke zvětšení využitelné části rozsahu celkového jasu snímku. V praxi je tato metoda účinná např. při úpravách špatné expozice, kdy se část objektu jeví na snímku buďto příliš přepálená, nebo naopak nedostatečně osvětlená[3]. Praktické využití této metody nalezneme, stejně jako se snaží nastínit vytvořený applet, zejména v lékařství, přesněji pak při zkoumání nepříliš zdárně exponovaných rentgenových snímků.

## 2 Rozbor práce

V této kapitole bych se rád zaměřil na podrobnější vysvětlení implementace jednotlivých metod ve zdrojových kódech appletů. Všechny applety byly napsány v jazyce HTML s využitím kaskádových stylů a čistého JavaScriptu, neboť využití dalších rozšiřujících knihoven se v mém případě neprojevovalo jako nezbytné.

Na první pohled mohou vytvořené applety působit trochu obstarožně, jelikož jsem preferoval funkčnost, nenáročnost a kompatibilitu napříč prohlížeči nad samotným designem. U většiny appletů, které jsem tvořil, jsem se snažil využít k zarovnání jednotlivých prvků na stránce způsob zarovnání do buněk pomyslné tabulky. V současné době s tímto řešením na webu již příliš nesetkáme a na moderních webových stránkách se k těmto účelům využívá spíše vymezení pomyslných oblastí elementy `<div>` a užití vybraných vlastností kaskádových stylů, jež poskytují široké možnosti zarovnání prvků na stránce. Pro mou práci se však tento způsob ukázal jako velmi užitečný, neboť v grafickém rozhraní appletů se často nachází více interaktivních prvků vedle sebe (např. prostor pro vykreslení originálního obrázku a zvláště prostor pro překreslený obrázek na základě naprogramovaného algoritmu), z toho důvodu se mi tato cesta jevila jako nejschůdnější pro mou situaci.

V následujících odstavcích se pokusím nejen o něco více nahlédnout pod pokličku provedné implementaci užitých principů a algoritmů, ale také podrobněji a zároveň co nejsrozumitelněji osvětlit principy jejich fungování v této dokumentaci, a to jak z praktického, tak teoretického hlediska.

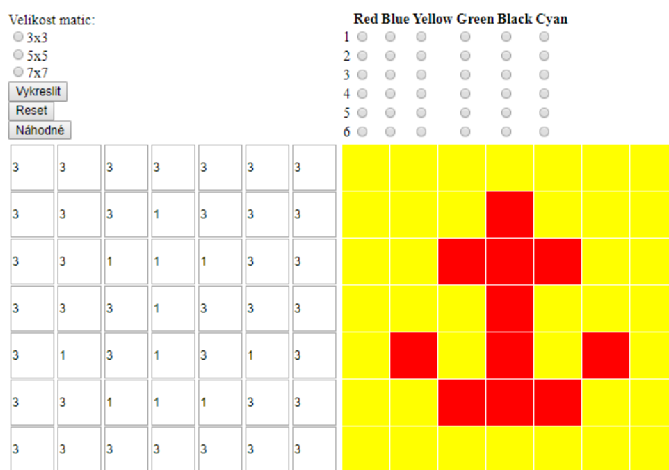
### 2.1 Paletová (indexová) reprezentace obrazu

Zabýváme se oblastí bitmapové (rastrové) grafiky, kde základním stavebním kamenem každého obrazu je pixel. Jak již bylo zmíněno v předchozích kapitolách, tento pixel je v RGB modelu vyjádřen třemi hodnotami barevných kanálů, a současně platí, že všechny tyto tři hodnoty určují jako celek výslednou barvu daného pixelu. Existují různé formáty pro ukládání bitmapové grafiky, mezi nejznámější patří např. JPG (JPEG), PNG, BMP nebo GIF[1]. Všechny tyto formáty se navzájem mezi sebou liší ve způsobu zpracování obrazu i jeho komprese, mohou být uloženy v barvách RGB s určitou barevnou hloubkou, ale také s omezeným počtem barev nebo v tzv. indexovém módu s paletou barev.

Paletová reprezentace tedy funguje na principu existující převodní tabulky, kde hodnota konkrétního pixelu neznačí jeho přesnou barvu, ale je pouze ukazatelem a odkazuje tak na konkrétní pozici v paletě barev. Tato tabulka je při uložení automaticky vytvořena a je součástí každého obrazu, který je uložen ve formátu pracujícím

s paletou barev. Z těchto poznatků plyne, že tento grafický formát je vhodný především pro ukládání obrazů, které neobsahují příliš velké množství barev (např. screenshot obrazovky, na které běží jednoduchý textový editor či webová stránka) a nehodí se příliš pro fotografie s velkým množstvím detailů a širokou škálou barev a barevných přechodů.

Applet prakticky zjednodušeně znázorňuje strukturu takového obrazového formátu. Vpravo nahoře se nachází sloupec s možností výběru barvy do převodní tabulky. Vlevo dole pak nevidíme přímo RGB hodnoty představující barvu pixelů, ale pouze hodnoty odkazující na převodní tabulku, ve které se již nachází barva přidružená právě této hodnotě. V pravé dolní části se poté vykreslí výsledek, jenž bychom viděli např. po uložení a následném zobrazení takového souboru. Vlevo nahoře se ještě nachází možnost zvolit velikost zobrazené matice, tlačítko pro obnovení appletu do výchozího nastavení a pro náhodné vygenerování hodnot v naší matici. V rámci této bakalářské práce bylo mým cílem implementovat funkcionalitu appletu a odladit jej tak, aby korektně znázorňoval probíranou teorii studentům a bylo jej možné použít pro podporu výuky.



Obr. 2.1: Ukázka appletu znázorňujícího princip paletové (indexové) reprezentace obrazu.

Po technické stránce je tento applet znázorňující principy paletové (indexové) reprezentace obrazu naprogramován pouze v čistém JavaScriptu, s využitím HTML a kaskádových stylů a nevyužívá žádných externích knihoven třetích stran, stejně jako applet předchozí. Tato skutečnost by sama o sobě mělo zajistit velmi dobrou kompatibilitu napříč různými webovými prohlížeči i mobilními zařízeními. Protože applet umožňuje pro demonstraci principů vybírat jeden ze tří typů přednastavených matic (веду, že se konkrétně jedná o rozměry 3x3, 5x5 a 7x7), samotného přepínání jsem docílil tím způsobem, že na samotné HTML stránce jsou načteny všechny tyto

typy matic, ale v daný okamžik mají nastaven atribut viditelnosti pouze některé, a to v závislosti na zaškrťávacím prvku typu radio button. Ostatní matice tedy zůstávají načteny v paměti, ale jsou schovány pomocí CSS atributu display: hidden;. Přesně tuto funkci zajišťuje metoda changeMatrixSize(), které se jako vstupní parametr předává velikost aktuálně zvolené matice.

```
function changeMatrixSize(type) {
  switch (type) {
    case "3x3":
      document.getElementById('input5x5').style.display = "none";
      document.getElementById('output5x5').style.display = "none";
      document.getElementById('input7x7').style.display = "none";
      document.getElementById('output7x7').style.display = "none";
      document.getElementById('input3x3').style.display = "block";
      document.getElementById('output3x3').style.display = "block";
      break;
    case "5x5":
      document.getElementById('input3x3').style.display = "none";
      document.getElementById('output3x3').style.display = "none";
      document.getElementById('input7x7').style.display = "none";
      document.getElementById('output7x7').style.display = "none";
      document.getElementById('input5x5').style.display = "block";
      document.getElementById('output5x5').style.display = "block";
      break;
    case "7x7":
      document.getElementById('input3x3').style.display = "none";
      document.getElementById('output3x3').style.display = "none";
      document.getElementById('input5x5').style.display = "none";
      document.getElementById('output5x5').style.display = "none";
      document.getElementById('input7x7').style.display = "block";
      document.getElementById('output7x7').style.display = "block";
      break;
    default:
      break;
  }
}
```

Obr. 2.2: Palet. repr. obrazu - metoda changeMatrixSize()

Grafické uživatelské rozhraní této výukové aplikace obsahuje prakticky 2 viditelné matice, z toho jednu matici vstupní (umístěna nalevo, obsahuje hodnoty indexů 1 - 6 podle přiřazených barev v pravém horním rohu) a druhou matici výstupní (umístěna vpravo, sem se vykreslují změny jednotlivých pixelů). Bylo tedy nutné zajistit vazbu mezi těmito maticemi, aby při úpravě parametrů ve vstupní části se mohly změny přenášet na výstup. Pro řešení a úspěšné vyřešení tohoto problému jsem využil atribut třídy u prvků HTML jazyka, a záměrně jsem zvolil ve zdrojovém kódu tohoto appletu podobné jména tříd pro vstupní i výstupní matice. Tyto jména tříd se tak liší pouze v jednom jediném písmenku. Následně je tak možné, a velice snadno realizovatelné za pomoci práce s řetězci toto písmenko změnit a tím zajistit reakci výstupní matice na provedené změny ve vstupní matici a to v reálném čase, bez nutnosti využívat tlačítko pro manuální překreslení výstupního okna.

K tomu účelu slouží ve zdrojovém kódu metody radioChangeColor() a changeIndexNumber(). První z nich se stará o přiřazení indexu ke konkrétní barvě (pomocí tabulky vpravo nahoře), zatímco druhá metoda changeIndexNumber na základě znalosti hodnoty řádku vstupní matice se změnou znaku přesměruje na výstup a nastaví novou barvu pro všechny shodné hodnoty indexů ze vstupní matice do matice výstupní.

```

function radioChangeColor(color,row) {
  var x, i, a;
  switch (row) {
    case "row1":
      x = document.getElementsByName('inMtx');
      for (i = 0; i < x.length; i++) {
        if (x[i].value == "1") {
          a = x[i].getAttribute('id')
          a = a.replace("i", "o");
          document.getElementById(a).style.backgroundColor = color;
        }
      }
      break;
    case "row2":
      x = document.getElementsByName('inMtx');
      for (i = 0; i < x.length; i++) {
        if (x[i].value == "2") {
          a = x[i].getAttribute('id')
          a = a.replace("i", "o");
          document.getElementById(a).style.backgroundColor = color;
        }
      }
      break;
    case "row3":
      x = document.getElementsByName('inMtx');
      for (i = 0; i < x.length; i++) {
        if (x[i].value == "3") {
          a = x[i].getAttribute('id')
          a = a.replace("i", "o");
          document.getElementById(a).style.backgroundColor = color;
        }
      }
      break;
    ...
    ...
    ...
    ...
    case "row6":
      x = document.getElementsByName('inMtx');
      for (i = 0; i < x.length; i++) {
        if (x[i].value == "6") {
          a = x[i].getAttribute('id')
          a = a.replace("i", "o");
          document.getElementById(a).style.backgroundColor = color;
        }
      }
      break;
    default:
      break;
  }
}

```

Obr. 2.3: Palet. repr. obrazu - metoda radioChangeColor()

## 2.2 Míchání barev v různých modelech

Tento applet má za cíl názorně demonstrovat přepočty mezi jednotlivými barevnými modely a prostory. Základními ovládacími prvky jsou zde trojice posuvníků, pro každý barevný model či prostor jeden, nad posuvníky se nachází aktuální hodnoty nastavovaných parametrů. Při uživatelském nastavení určité barvy pomocí jednoho barevného modelu dojde současně ke změně nastavených parametrů i u modelů a prostorů dalších, neboť jsou tyto prvky vzájemně propojeny na základě přepočtů probíhajících na pozadí. Pokud tedy např. nastavíme parametry v prostoru Lab, tedy Světlost a parametry a, b na námi požadované hodnoty, můžeme tak snadno

```

function changeIndexNumber(cId) {
    var a, getVal, getRow, getRadio, getRadioId;

    a = cId;
    a = a.replace("1", "0");

    getVal = document.getElementById(cId).value;
    getRow = "row" + getVal;
    getRadio = document.getElementsByName(getRow);

    for (i = 0; i < getRadio.length; i++) {
        if (getRadio[i].checked) {
            getRadioId = getRadio[i].getAttribute("id");
        }
    }

    switch (getRadioId.charAt(0)) {
        case "r":
            document.getElementById(a).style.backgroundColor = "red";
            break;
        case "b":
            document.getElementById(a).style.backgroundColor = "blue";
            break;
        case "y":
            document.getElementById(a).style.backgroundColor = "yellow";
            break;
        case "g":
            document.getElementById(a).style.backgroundColor = "green";
            break;
        case "k":
            document.getElementById(a).style.backgroundColor = "black";
            break;
        case "c":
            document.getElementById(a).style.backgroundColor = "cyan";
            break;
        default:
            break;
    }
}

```

Obr. 2.4: Palet. repr. obrazu - metoda changeIndexNumber()

zjistit, jaké hodnoty bychom museli nastavit v modelu RGB, CMY, HSB a YCbCr, abychom získali totožnou barvu. Od pravého horního rohu do zhruba středu stránky se pak nachází oblast, v níž je další ohraničený prostor čtvercového rozměru. Vnější a větší z nich nám zobrazuje aktuální nastavenou barvu, menší prostor potom vyznačuje barvu inverzní k námi zvolené barvě. Máme-li tedy zvolenou červenou barvu o souřadnicích 255 0 0 v RGB modelu, ve větším čtverci tak bude zobrazena právě tato barva a v menší oblasti její doplněk, tedy barva tyrkysová (0 255 255 v RGB).

Podobně jako již předchozí zmíněné applety, i tento jsem se pokusil realizovat bez využití jakékoli externí knihovny, vše se tedy odehrává v klasickém čistém JavaScriptu a v základních funkcích HTML jazyka a kaskádových stylů. Tento applet se mi zdál mírně obtížnější na návrh a přestože základní funkcionality se mi nejspíše docílit podařilo, s největší pravděpodobností by se ale hlavní struktura programu dala zapsat o něco lépe a elegantněji.

Zdrojový kód stránky využívá tabulkového layoutu, který by se, jak jsem již zmínil, dal považovat na dnešním webu za zastaralý, nicméně pro případ realizace tohoto appletu se s ním velmi dobře pracovalo.

Celý program vypadá a funguje následovně: Hlavními a prakticky jedinými ovládacími prvky jsou posuvníky náležící konkrétním barevným modelům a prostorům. Větší část okna poté zabírají dvě soustředné čtvercové plochy, z nichž větší znázorňuje aktuální nastavenou barvu v RGB modelu a malý čtverec uvnitř značí barvu doplňkovou, inverzní či převrácenou k modelu RGB, tedy barvu v modelu CMY. Nad každým posuvníkem v jednotlivé buňce představující konkrétní barevný model

či prostor se nachází údaj o aktuální nastavené hodnotě, buď v dekadickém formátu, anebo v procentuálním vyjádření. Měníme-li polohu některého z posuvníků, můžeme si všimnout, že se mění polohy i některých dalších posuvníků z kategorie odlišného barevného modelu či prostoru. Tyto posuvníky jsou si vzájemně svázány a to zcela úmyslně, pro lepší názornost a představu při převodu hodnot mezi odlišnými modely a prostory.

Každý z nabízených modelů či prostorů má svou vlastní nastavovací metodu `set()`, která poté, co je zavolána, přečte hodnoty všech sliderů v dané skupině a provede nastavení příslušné barvy na hlavních zobrazovacích čtvercových plochách programu. Po úspěšném provedení této operace je volána metoda `autoConv()` se vstupními parametry vypočtených hodnot v RGB souřadnicích 0-255. Tato metoda obsahuje klasickou konstrukci přepínače (Switch), který se na základě poslední nastavované hodnoty přepne na příslušné návěští konkrétního barevného modelu nebo prostoru a provede aktualizaci hodnot všech ostatních posuvníků. Zjednodušeně se dá tedy říci, že metoda `autoConv()` slouží výhradně pro přepočítání hodnot ve všech ostatních modelech a prostorech, tedy kromě toho, co celou tuto akci vyvolal.

```
function setRGB() {
    var rgb = "rgb";
    var r_rgb = document.getElementById('sliderr').value;
    var g_rgb = document.getElementById('sliderg').value;
    var b_rgb = document.getElementById('sliderb').value;

    document.getElementById("outer-div").style.backgroundColor = rgb2hex(r_rgb, g_rgb, b_rgb);
    document.getElementById("inner-div").style.backgroundColor = rgb2hex(255-r_rgb, 255-g_rgb, 255-b_rgb);

    autoConv(rgb, r_rgb, g_rgb, b_rgb);
}

function setCMY() {
    var cmy = "cmy";
    var c_cmy = document.getElementById('sliderc').value;
    var m_cmy = document.getElementById('sliderm').value;
    var y_cmy = document.getElementById('slidery').value;

    document.getElementById("outer-div").style.backgroundColor = rgb2hex(255-c_cmy, 255-m_cmy, 255-y_cmy);
    document.getElementById("inner-div").style.backgroundColor = rgb2hex(c_cmy, m_cmy, y_cmy);

    autoConv(cmy, c_cmy, m_cmy, y_cmy);
}

function setHSB() {
    var hsb = "hsb";
    var h_hsb = document.getElementById('sliderh').value;
    var s_hsb = document.getElementById('sliders').value;
    var b_hsb = document.getElementById('sliderb').value;

    s_hsb/=100;
    b_hsb/=100;

    var C = b_hsb*s_hsb;

    var hh = h_hsb/60;

    var X = C*(1-Math.abs(hh*2-1));
```

Obr. 2.5: Míchání barev - Nastavovací funkce.

Protože i zde jsem řešil obtíže při snaze o pochopení teorie vzájemných převodů mezi barevnými modely a prostory, snažil jsem se vycházet z již funkčních a ověřených postupů již napsaných přímo v jazyce JavaScript. Většina výpočtů týkajících se vztahů mezi barevnými modely a prostory je proto s drobnými úpravami převzatá z online kalkulačků na serveru [rapidtables.com](http://rapidtables.com)[37].



```

function rgb2hex(red, green, blue) {
    var rgb = blue | (green << 8) | (red << 16);
    return '#' + (0x1000000 + rgb).toString(16).slice(1)
}

function autoConv(colorModel, val1, val2, val3) {
    switch (colorModel) {
        case "rgb":
            /*to CMY (value 1,2,3 = r,g,b)*/
            var c_cmy = 255 - val1;
            var m_cmy = 255 - val2;
            var y_cmy = 255 - val3;

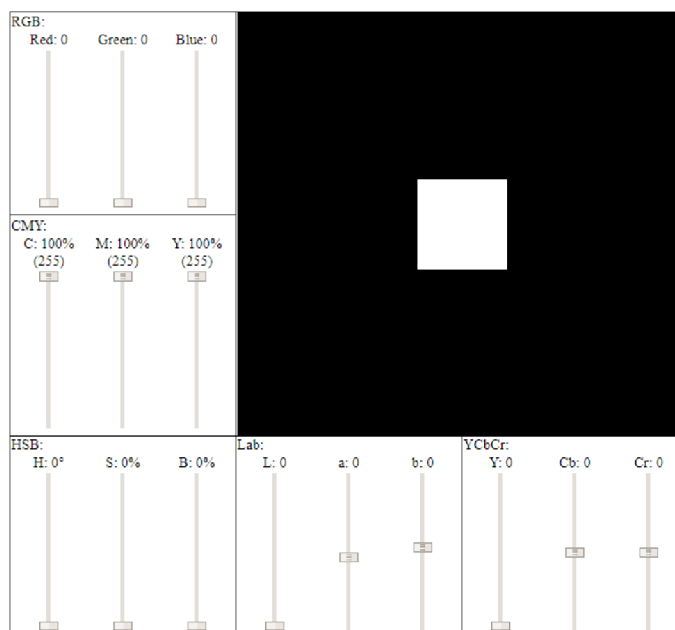
            document.getElementById("sliderc").value = c_cmy;
            document.getElementById("sliderm").value = m_cmy;
            document.getElementById("slidery").value = y_cmy;

            document.getElementById("cmycval").value = c_cmy;
            document.getElementById("cmycperc").value = Math.round((c_cmy/255)*100);
            document.getElementById("cmymval").value = m_cmy;
            document.getElementById("cmymperc").value = Math.round((m_cmy/255)*100);
            document.getElementById("cmyyval").value = y_cmy;
            document.getElementById("cmyyperc").value = Math.round((y_cmy/255)*100);

            /*HSB*/
            var r = val1;
            var g = val2;
            var b = val3;
            z/=255;
            g/=255;
            b/=255;
            var M = Math.max(z,g,b);
            var m = Math.min(z,g,b);
            var C = M-m;
            var h;
            if( C==0 ) h=0;
    }
}

```

Obr. 2.6: Míchání barev - funkce autoConv();

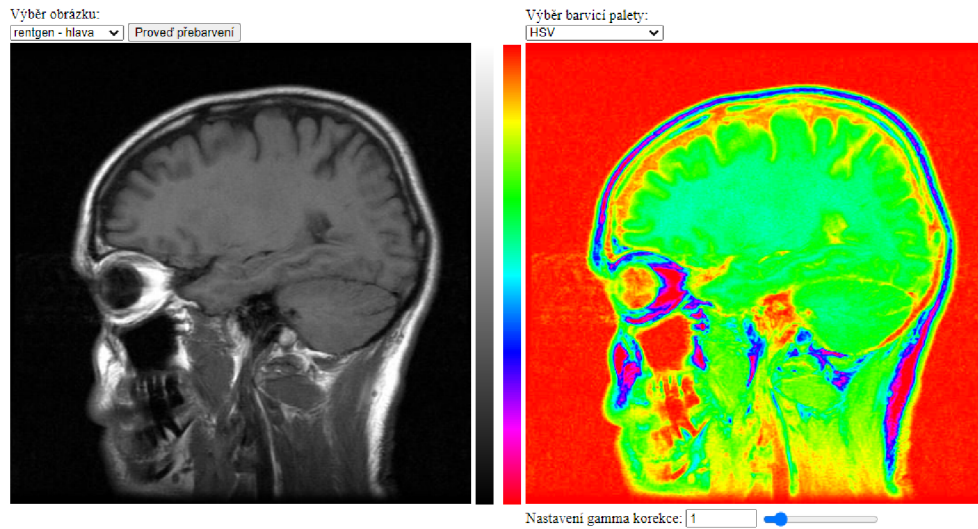


Obr. 2.7: Ukázka appletu pro míchání barev v různých modelech.

## 2.3 Převod šedotónového obrazu na barevný (nepravé barvy)

Cílem tohoto appletu je znázornění možností umělého obarvení vstupního obrazu (černobílá fotografie, rentgenový snímek, případně šedotónový vzor) volbou z ně-

kolika druhů barvicí palety. Mým úkolem zde rovněž bylo implementovat možnost uživatelské změny nastavení gamma korekce výsledného obarveného obrazu. Grafické rozhraní appletu jsem původně uvažoval využít z části z mého appletu pro převod barevného obrázku na šedotónový, přičemž jsem měl v plánu jej rozšířit o další prvky, např. nastavení barvy koncových bodů, uživatelské nastavení hodnoty gamma pomocí posuvníku a možnost výběru ze čtyř druhů barvicích palet.



Obr. 2.8: Ukázka appletu pro převod obrazu do nepravých barev.

## 3 Závěr

V rámci bakalářské práce byla doplněna a notně rozšířena nejen tato dokumentace, ale hlavním požadavkem bylo zejména dokončení všech zadaných appletů tak, aby správně fungovaly alespoň se základními funkcemi. Domnívám se, že tento požadavek byl splněn a applety mohou sloužit k zamýšlenému účelu.

# Literatura

- [1] RAJMIC, P. *Základy počítačové sazby a grafiky*. Brno: Vysoké učení technické v Brně, 2017. ISBN: 978-80-214-4451- 5.
- [2] Gonzalez, R.C.; Woods, R.E.: *Digital Image Processing*. Třetí vydání. Pearson; 2007. ISBN 978-0131687288
- [3] NOVOTNÝ, L. *Vytvoření programů pro podporu výuky zpracování signálů a obrazů* [online]. Brno: Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. 2014. Dostupné z:  
<<https://www.vutbr.cz/studenti/zav-prace/detail/73812>>
- [4] NOVOTNÝ, L. *Barvení šedotónových obrazů (nepravé barvy) – applet*. [online]. Brno: Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. 2014. Dostupné z:  
<<https://www.vutbr.cz/studenti/zav-prace/detail/73812>>
- [5] HLAVATÝ, J. *Soubor programů pro podporu výuky zpracování obrazů* [online]. Brno: Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. 2013. Dostupné z:  
<<https://www.vutbr.cz/studenti/zav-prace/detail/66567>>
- [6] HLAVATÝ, J. *Převod barevného obrazu na šedotónový – applet. Míchání barev v různých modelech - applet*. [online]. Brno: Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. 2013. Dostupné z:  
<<https://www.vutbr.cz/studenti/zav-prace/detail/66567>>
- [7] RIŠKO, M. *Paletová (indexovaná) reprezentace obrazu*. [online]. Brno: Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. 2014. Dostupné z:  
<<https://www.vutbr.cz/studenti/zav-prace/detail/77455>>
- [8] BOLEČEK, L. *Zobrazování černobílých snímků v nepravých barvách* [online]. Brno: Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. 2010. Dostupné z:  
<<https://www.vutbr.cz/studenti/zav-prace/detail/31105>>
- [9] LEHMAN, Thomas, KASER, Andreas, REPGES, Rudolf *A simple parametric equation for pseudocoloring grey scale images keeping their original brightness progression* [online]. 1993-1996 [cit. 2020-06-01]. Dostupné z:  
<<https://www.sciencedirect.com/science/article/pii/S0262885696011201>>

- [10] *SCOTT, Ed . Masking Spectrum 5 - photo.net* [online]. 1997 [cit. 2020-05-25]. Dostupné z:  
<<http://web.archive.org/web/20170119160339/http://photo.net/learn/optics/edscott/ms000050.htm>>
- [11] *HOZMAN, J. Základní metody předzpracování obrazu* [online]. 2003 [cit. 2020-05-25]. Dostupné z:  
<[http://webzam.fbmi.cvut.cz/hozman/Zprac\\_obr\\_prisp\\_kurz\\_UEM\\_3\\_2003.pdf](http://webzam.fbmi.cvut.cz/hozman/Zprac_obr_prisp_kurz_UEM_3_2003.pdf)>
- [12] *Interaktivní applety pro výuku – rozcestník* [online]. [cit. 2018-12-14]. Dostupné z:  
<<http://www.utko.feec.vutbr.cz/~rajmic/applets/>>
- [13] *HTML canvas getImageData() Method* [online]. [cit. 2018-12-14]. Dostupné z:  
<[https://www.w3schools.com/tags/canvas\\_getimagedata.asp](https://www.w3schools.com/tags/canvas_getimagedata.asp)>
- [14] *HTML5 Canvas — the Basics* [online]. [cit. 2018-12-14]. Dostupné z:  
<<https://dev.opera.com/articles/html5-canvasbasics/#pixelbasedmanipulation>>
- [15] *Convert Image to Grayscale* [online]. [cit. 2018-12-14]. Dostupné z:  
<<https://www.coursera.org/lecture/duke-programming-web/convert-imageto-grayscale-yYRvg>>
- [16] *Hypertext Markup Language - Wikipedie* [online]. [cit. 2018-12-14]. Dostupné z:  
<[https://cs.wikipedia.org/wiki/Hypertext\\_Markup\\_Language](https://cs.wikipedia.org/wiki/Hypertext_Markup_Language)>
- [17] *HTML5 - Wikipedie* [online]. [cit. 2018-12-14]. Dostupné z:  
<<https://cs.wikipedia.org/wiki/HTML5>>
- [18] *JavaScript* [online]. [cit. 2018-12-14]. Dostupné z:  
<<https://developer.mozilla.org/cs/docs/Web/JavaScript>>
- [19] *Kaskádové styly - Wikipedie* [online]. [cit. 2018-12-14]. Dostupné z:  
<[https://cs.wikipedia.org/wiki/Kaskádové\\_styly](https://cs.wikipedia.org/wiki/Kaskádové_styly)>
- [20] *RGB - Wikipedie* [online]. [cit. 2018-12-14]. Dostupné z:  
<<https://cs.wikipedia.org/wiki/RGB>>
- [21] *CMYK - Wikipedie* [online]. [cit. 2018-12-14]. Dostupné z:  
<<https://cs.wikipedia.org/wiki/CMYK>>
- [22] *Lab - Wikipedie* [online]. [cit. 2018-12-14]. Dostupné z:  
<<https://cs.wikipedia.org/wiki/Lab>>

- [23] *CIE XYZ - Wikipedie* [online]. [cit. 2018-12-14]. Dostupné z:  
<[https://cs.wikipedia.org/wiki/CIE\\_XYZ](https://cs.wikipedia.org/wiki/CIE_XYZ)>
- [24] *sRGB - Wikipedie* [online]. [cit. 2018-12-14]. Dostupné z:  
<<https://cs.wikipedia.org/wiki/SRGB>>
- [25] *Kaskádové styly* [online]. [cit. 2018-12-14]. Dostupné z:  
<[https://cs.wikipedia.org/wiki/Kaskádové\\_styly](https://cs.wikipedia.org/wiki/Kaskádové_styly)>
- [26] *YCbCr color spaces* [online]. [cit. 2019-05-12]. Dostupné z:  
<<https://www.hisour.com/ycbcr-color-spaces-26075/>>
- [27] *Barva - Wikipedie* [online]. [cit. 2019-05-12]. Dostupné z:  
<<https://cs.wikipedia.org/wiki/Barva>>
- [28] *Chromatický diagram CIE 1931* [online]. [cit. 2019-05-12]. Dostupné z:  
<[https://cs.wikipedia.org/wiki/CIE\\_XYZ#/media/File:CIE1931xy\\_blank.svg](https://cs.wikipedia.org/wiki/CIE_XYZ#/media/File:CIE1931xy_blank.svg)>
- [29] BOLEČEK, Libor. *Program pro zobrazení černobílých snímků v nepravých barvách (Pseudo-colour Paging of the Monochromatic Picture)*. *Elektrorevue* [online]. 2010, 20.12.2010, 2010(125), 9 [cit. 2018-05-28]. Dostupné z:  
<<http://www.elektrorevue.cz/cz/clanky/zpracovani-signalu/15/program-prozobrazeni-cernobilych-snimku-v-nepravych-barvach/>>
- [30] *Color.h* [online]. [cit. 2018-12-14]. Dostupné z:  
<[http://www.miraproject.org/MIRA-doc/Color\\_8h\\_source.html](http://www.miraproject.org/MIRA-doc/Color_8h_source.html)>
- [31] *ColorSpaceConverter.java* [online]. [cit. 2018-12-14]. Dostupné z:  
<[https://imagej.nih.gov/ij/plugins/download/Color\\_Space\\_Converter.java](https://imagej.nih.gov/ij/plugins/download/Color_Space_Converter.java)>
- [32] *ColorConverter.java* [online]. [cit. 2018-12-14]. Dostupné z:  
<<https://github.com/nekdenis/AndroidColorDiff/blob/master/app/src/main/java/com/github/nekdenis/util/ColorConverter.java>>
- [33] *ColorSpaceConverter.java* [online]. [cit. 2018-12-14]. Dostupné z:  
<<https://imagej.nih.gov/ij/source/ij/process/ColorSpaceConverter.java>>
- [34] *Rgb2lab.c* [online]. [cit. 2018-12-14]. Dostupné z:  
<<https://www.csie.ntu.edu.tw/~b92069/HWs/rgb2lab.c>>

- [35] *RGB to XYZ* [online]. [cit. 2018-12-14]. Dostupné z:  
<[http://www.bruceindbloom.com/index.html?Eqn\\_RGB\\_XYZ\\_Matrix.html](http://www.bruceindbloom.com/index.html?Eqn_RGB_XYZ_Matrix.html)>
- [36] *XYZ to LAB* [online]. [cit. 2018-12-14]. Dostupné z:  
<[http://www.bruceindbloom.com/index.html?Eqn\\_XYZ\\_to\\_Lab.html](http://www.bruceindbloom.com/index.html?Eqn_XYZ_to_Lab.html)>
- [37] *RGB to HSV* [online]. [cit. 2018-12-14]. Dostupné z:  
<<https://www.rapidtables.com/convert/color/rgb-to-hsv.html>>

# A Obsah přiloženého CD

Převod barevného obrazu na šedotónový:

- Applet je nutné umístit na webhosting (FTP), lokálně jej nelze spustit z důvodu přístupu k obrázkům (Cross-origin resource sharing)
- /PrevodBarObrSedot/index.html
- /PrevodBarObrSedot/images/
- /PrevodBarObrSedot/images/Lod.jpg
- /PrevodBarObrSedot/images/Motorky.jpg
- /PrevodBarObrSedot/images/Paleta.jpg
- /PrevodBarObrSedot/images/Papousci.jpg
- /PrevodBarObrSedot/images/RGB.jpg

Míchání barev v barevných modelech:

- /MichaniBarev/index.html

Paletová (indexová) reprezentace obrazu:

- /PaletReprezObr/index.html

Nepravé barvy:

- /NepraveBarvy/index.html
- /NepraveBarvy/images/
- /NepraveBarvy/images/1.png
- /NepraveBarvy/images/2.png
- /NepraveBarvy/images/3.png
- /NepraveBarvy/images/4.png
- /NepraveBarvy/images/5.png
- /NepraveBarvy/images/6.png
- /NepraveBarvy/images/7.png
- /NepraveBarvy/images/9.png
- /NepraveBarvy/images/10.png
- /NepraveBarvy/images/11.png
- /NepraveBarvy/images/12.png
- /NepraveBarvy/images/placeholder.png