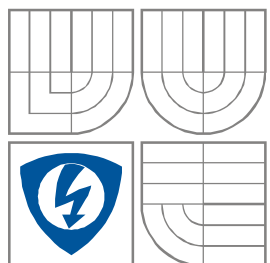


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ

ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND  
COMMUNICATION

DEPARTMENT OF RADIO ELECTRONICS

## Zobrazování černobílých snímků v nepravých barvách

Pseudo- colour paging of the monochromatic pictures

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

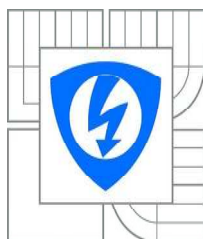
Bc. Libor Boleček

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. Václav Říčný, CSc.

BRNO, 2010



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav radioelektroniky

# Diplomová práce

magisterský navazující studijní obor

Elektronika a sdělovací technika

*Student:* Bc. Libor Boleček

*ID:* 78548

*Ročník:* 2

*Akademický rok:* 2009/2010

**NÁZEV TÉMATU:**

**Zobrazování černobílých snímků v nepravých barvách**

## POKYNY PRO VYPRACOVÁNÍ:

Na základě analýzy metod zobrazování v nepravých barvách vytvořte program umožňující převod a zobrazení monochromatického snímku v běžných datových formátech na snímek zobrazený ve zvolených nepravých barvách. Program musí umožňovat změnu přiřazení barev sledující zvýšení subjektivního rozlišení jasových rozdílů v původním snímku.

Diplomová práce má obsahovat zejména:

- volbu vhodného programovacího prostředí a návrh programového vybavení umožňujícího vstup digitálních obrazových dat v nejčastěji používaných obrazových formátech,
- vývojový diagram a detailní návrh vlastního programu s důrazem na interaktivní a intuitivní ovládání,
- odladění a funkční ověření programu,
- aplikační příklady funkce programu.

## DOPORUČENÁ LITERATURA:

[1] POYNTON, CH. Digital Video and HDTV. San Francisco: MKP, 2005.

[2] BLAŽEK, R. Číslicový systém pro zobrazování v nepravých barvách. Diplomová práce. VUT, 1988.

*Termín zadání:* 8.2.2010

*Termín odevzdání:* 21.5.2010

*Vedoucí práce:* prof. Ing. Václav Říčný, CSc.

**prof. Dr. Ing. Zbyněk Raida**

*Předseda oborové rady*

## UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové 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.

# LICENČNÍ SMLOUVA

## POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

### 1. Pan/paní

Jméno a příjmení: Bc. Libor Boleček  
Bytem: Sídliště 552, Štěpánov, 783 13  
Narozen/a (datum a místo): 23. dubna 1985 ve Šternberku

(dále jen „autor“)

a

### 2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií  
se sídlem Údolní 53, Brno, 602 00  
jejímž jménem jedná na základě písemného pověření děkanem fakulty:  
prof. Dr. Ing. Zbyněk Raida, předseda rady oboru Elektronika a sdělovací technika  
(dále jen „nabyvatel“)

## Čl. 1

### Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- ☒ disertační práce
  - ☒ diplomová práce
  - ☒ bakalářská práce
  - ☒ jiná práce, jejíž druh je specifikován jako .....
- (dále jen VŠKP nebo dílo)

Název VŠKP: Zobrazování černobílých snímků v nepravých barvách

Vedoucí/ školitel VŠKP: prof. Ing. Václav Říčný, CSc.

Ústav: Ústav radioelektroniky

Datum obhajoby VŠKP: \_\_\_\_\_

VŠKP odevzdal autor nabyvateli\*:

v tištěné formě – počet exemplářů: 2

v elektronické formě – počet exemplářů: 2

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

---

\* hodící se zaškrtněte

## Článek 2

### Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
  - ihned po uzavření této smlouvy
  - ☒ 1 rok po uzavření této smlouvy
  - ☒ 3 roky po uzavření této smlouvy
  - ☒ 5 let po uzavření této smlouvy
  - ☒ 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

## Článek 3

### Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 21. května 2010

.....  
Nabyvatel

.....  
Autor

## Anotace

Diplomová práce zobrazování černobílých snímků v nepravých barvách se zabývá prostudováním metodiky zobrazení černobílých snímků v nepravých barvách a aplikací zjištěných poznatků při vytvoření počítačového softwaru. Hlavní funkcí programu je převedení monochromatického snímku do nepravých barev. Tímto postupem je docíleno zvýšení diagnostické výtěžnosti snímku. Největší uplatnění může program najít v medicínské oblasti. K pseudobarvení je použito několik různých metod, které jsou pro získání co nejlepších výsledků kombinované s úpravami jasové stupnice snímku zvyšujícími kontrast obrazů.

Program je vytvořen v programovacím prostředí Visual Studio 2008 a napsán programovacím jazykem C#. Vstupem programu je monochromatický obraz v různých formátech (JPEG, BMP). Výstupem programu může být vytisknutý snímek v nepravých barvách nebo uložený obrázek opět v několika různých formátech.

## Klíčová slova

software , monochromatický snímek, falešné barvy, diagnostika, jasová stupnice

## Annotation

Diploma thesis displaying black and white image in false color includes a study of methods and creation of computer software. The main function of the program is to convert the monochrome image into the pseudo colors. This procedure is achieved increasing the diagnostic yield of image. The program may find application in medical field. The pseudocoloring is used several different methods for obtaining the best results, combined with modifications increasing brightness scale image contrast images. The program is created in the programming environment of Visual Studio 2008 and written in C # programming language. Entry program is monochrome images in various formats (JPEG, BMP). The output of the program can be printed in a false-color image or stored image again in several different formats

## Keywords

software, monochromatic image, false color, diagnosis, luminance scale

BOLEČEK, L. *Zobrazování černobílých snímků v nepravých barvách*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 60 s. Vedoucí semestrální práce prof. Ing. Václav Říčný, CSc.

# Prohlášení

Prohlašuji, že svůj semestrální projekt na téma Zobrazování černobílých snímků v nepravých barvách jsem vypracoval samostatně pod vedením vedoucího semestrálního projektu 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 uvedeného semestrálního projektu dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 21. května 2010

.....  
podpis autora

# Poděkování

Děkuji vedoucímu semestrálního projektu Prof. Ing. Václavu Říčnému, CSc. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mého semestrálního projektu.

V Brně dne 21. května 2010

.....  
podpis autora

# OBSAH

## Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Teorie barev</b>	<b>3</b>
2.1	Barva a barevné vnímání .....	3
2.2	Barevné modely .....	5
<b>3</b>	<b>Metody zvýrazňování obrazu</b>	<b>7</b>
3.1	Základní pojmy .....	7
3.2	Modifikace škály šedosti .....	8
3.2.1	Ostření obrazu .....	8
3.2.2	Redukce šumu .....	8
3.2.3	Pseudobarvení .....	8
3.2.4	Metoda hustotních řezů .....	9
<b>4</b>	<b>Zobrazování v nepravých barvách</b>	<b>10</b>
4.1	Jednoduché parametrické rovnice pro pseudobavení šedotónových obrazů....	10
4.2	Barevná paleta .....	13
4.3	Metoda založená na stupních jasu .....	14
4.4	Metoda založená na barevných prostorech.....	16
4.5	Pseudobarvicí krychle .....	18
4.6	Hustotní řezy .....	20
<b>5</b>	<b>Úpravy hodnot jasu</b>	<b>21</b>
5.1	Negativ .....	21
5.2	Zvýraznění části stupnice .....	22
5.3	Lineární zvýraznění kontrastu .....	23
5.4	Ekvalizace .....	24
5.5	Gama korekce.....	26
<b>6</b>	<b>Základní struktura programu Peacock</b>	<b>27</b>
6.1	Základní struktura.....	27
6.2	Grafické rozhraní.....	29
6.3	Form1.Designer.cs.....	29
6.4	Form1.cs .....	30
6.5	Helper.cs.....	36

6.5.1	Funkce transformace jasové stupnice.....	36
6.5.2	Funkce zobrazení v nepravých barvách .....	38
6.5.3	Funkce převodu bitmapa-pole.....	41
<b>7</b>	<b>Uživatelská příručka</b>	<b>43</b>
7.1	Hlavní obrazovka .....	43
7.2	Záložka tabulka .....	45
7.3	Záložka video .....	45
<b>8</b>	<b>dosažené výsledky</b>	<b>47</b>
8.1	MIPS 2.0 - Microscopy Image Processing Software.....	47
8.2	MultiSpec .....	48
8.3	Peacock.....	49
<b>9</b>	<b>Závěr</b>	<b>56</b>
	<b>Použitá literatura</b>	<b>58</b>
	<b>Seznam obrázků</b>	<b>59</b>

# 1 ÚVOD

Při měření v mnoha oborech bývají často získaná výstupní data reprezentována černobílým (monochromatickým) obrazovým snímkem – např. rentgenovým, sonografickým nebo termovizním snímkem. Hledaná obrazová informace je vyjádřena rozdílným jasnem jednotlivých bodů nebo částí daného snímku. Na základě takového obrazu například diagnostik určuje zdravotní stav pacienta. Dalšími příklady, kdy je výstupní informace vyjádřena monochromatickým snímkem, může být kromě medicínských snímků například geodézie, dálkový průzkum Země ze satelitů, hodnocení kvality tkanin, tepelného vyzařování objektu nebo kvality osvětlení prostor.

Vzhledem k vlastnostem lidského zraku a jeho omezené jasové rozlišovací schopnosti je výhodné šedotónový obraz pro vyhodnocování dále upravit, aby byla zlepšena jeho diagnostická vypovídací schopnost. K tomu slouží různé metody pro zvýrazňování obrazu. Cílem těchto metod je analýza vlastností scény a cílené zvýraznění těch parametrů obrazu, které jeho vypovídací schopnost při subjektivním hodnocení zvyšují. Metody zvýrazňování obrazu jsou popsány v kapitole 3. Jednou z velmi účinných metod je přiřazení určité barvy jednotlivým jasovým úrovním původního snímku.

Využití metody zobrazení v nepravých barvách (tzv. pseudocolor) vychází z experimentálně ověřené zkušenosti, že průměrný pozorovatel je schopen rozeznat mnohem více barevných odstínů (více jak 1000), než odstínů šedí (cca 60). Proto při přiřazení barevných odstínů původně šedotónovému obrazu (barvy s nulovou sytostí) může diagnostik získat mnohem lepší subjektivní vjem ze snímku a rozezná v něm mnohem víc detailů, přestože touto operací je zachována původní informace a žádná nová není přidána. Se zobrazením ve pseudobarvách se můžeme setkat tam, kde je požadováno převést původně monochromatický obraz na barevný obraz (např. u rentgenogramu, scintigramu, termogramu).

Nejjednodušší metodou zobrazení v nepravých barvách jsou fotografické postupy využívající pseudosolarizace, to znamená jevu, při kterém dojde k částečné inverzi obrazu, popř. k vytvoření světlých kontur na rozhraní světlých a tmavých míst, vznikajícího dodatečným osvětlením citlivé vrstvy během vyvolávání (tzv. Sabatierův jev). Problémem je ale velká pracnost, malá rychlost a malá variabilita pořizování barevných tónů. Pokud je zdrojem signálu televizní kamera, lze realizovat zařízení (hardware), které převádí monochromatický televizní signál složkové barevné signály  $U_R$ ,  $U_G$ ,  $U_B$  a synchronizační směs. Z těchto signálů se vytváří úplný barevný signál standardu PAL pro barevný televizní monitor. Další možnost představuje softwarové řešení problému – tj. převedení černobílého snímku do digitální podoby a převod do nepravých barev.

Softwarové řešení zobrazování v nepravých barvách pro zvýšení diagnostické výtěžnosti snímku je hlavním úkolem této diplomové práce. Vstupními daty vytvořeného programu je šedotónový statický snímek v různých formátech (JPEG, BMP) nebo videosekvence ve formátu avi. Výstupem je barevný snímek ve formátech JPEG nebo BMP. Vytvořený program nabízí uživatelům převod pomocí více metod (viz. kapitola 4). Cílem aplikace není jen prosté zobrazení v nepravých barvách s využitím jedné z mnoha různých metod, ale hlavní myšlenkou je nabídnutí variability zobrazení, vedoucí k efektivnímu zlepšení diagnostické výtěžnosti. Uživatel tak může

docílit mnoha různých přidělení barevných složek vstupnímu jasů a rozhodnout se, která metoda přinese největší zvýšení diagnostické výtěžnosti. V rozhodování hraje velkou roli také psychosenzorické subjektivní vnímání barev pozorovatele. Metody zobrazení v nepravých barvách je dále možné kombinovat s úpravami jasové stupnice, které umožňují zvětšení kontrastu snímků (viz. kapitola 5). Aplikace je naprogramována v jazyce C# ve volně šiřitelném programu Visual Studio 2008. Díky použití tohoto prostředí a daného jazyku je aplikace bez problému spustitelná i na PC, kde není prostředí instalováno.

## 2 TEORIE BAREV

Cílem kapitoly je definovat pojem barevné světlo a další základní pojmy používané v této práci a stručné pojednání o vnímání barev lidským okem. V diplomové práci je slovo barva použito v zúženém významu (a zastupuje pojem barevné světlo), přestože slovo barva je obecně nezcela přesné. Výsledný barevný vjem (přeneseně barva) má psychofyzikální charakter. To znamená, že nezávisí jen na velikostech fyzikálních veličin pozorovaného barevného světla, ale i na vlastnostech pozorovatele. Barevné světlo je charakterizováno třemi fyzikálními parametry a to dominantní vlnovou délkou [m], složením spektrálních složek [%] a jasem [nt]. A těmito psychosenzorickými parametry barevný tón, barevná sytost a jasnost.

### 2.1 Barva a barevné vnímání

Teorii a popisem barev se zabývá kolorimetrie. Světlo je nositelem barevné informace. Světlo je elektromagnetické vlnění, které lze popsat frekvencí  $f$  (vlnovou délkou  $\lambda$ ), rychlostí  $v$  a několika dalšími parametry. K šíření tohoto vlnění v prostoru dochází po kvantech. Světlo se šíří ve vakuu rychlostí přibližně  $2,998 \cdot 10^8$  m/s, v ostatních látkách se světlo šíří pomaleji. Závislost mezi rychlostí šíření, vlnovou délkou a frekvencí je dán vztahem

$$\lambda = \frac{c}{f}. \quad (2.1)$$

Frekvence viditelné (člověkem vnímané) části spektra záření se pohybuje v rozmezí  $3,9 \cdot 10^{14}$  -  $7,9 \cdot 10^{14}$  Hz. Tomu odpovídá rozsah vlnových délek 379 - 769 nm (viz. vztah 2.1). Frekvence světelného záření určuje barvu světla. Fialové světlo má největší frekvenci a červené nejmenší. Aby mělo udání vlnové délky smysl, je nutné znát materiál, pro který je tato veličina uváděna. Vlnová délka barevného světla je obvykle udává pro šíření ve vakuu.

Barevný vjem člověka zajišťují čípký. V lidském oku existují tři druhy čípků, lišící se barevnými pigmenty a citlivostí k vlnovým délkám, které odpovídají jednotlivým barvám. Normální vidění je trichromatické. Nejedná se ovšem o citlivost pouze na jednu frekvenci světla, nýbrž jsou nejvíce citlivé v okolí jedné frekvence světla a čím se bude vzdalovat od této frekvence více, tím méně bude čípek reagovat. Citlivost čípků je zobrazena na obrázku 2.1



Obr. 2.1 Citlivost čípků na světlo v závislosti na vlnové délce [1].

Z hlediska barevného vjemu jsou tzv. primární zdroje záření charakterizovány chromatičností a sekundární zdroje (objekty osvětlené primárním zdrojem) koloritou. Ta je určena chromatičností primárního zdroje a spektrálním koeficientem odrazivosti  $\rho$  ( $\lambda$ ) osvětleného povrchu. Důležitým jevem z hlediska vnímání barev je odraz světla od sekundárních zdrojů. Zdroje neodráží světelné záření všech vlnových délek, ale jeho část je pohlcována. Odraz světelného záření je dán spektrálním koeficientem odrazivosti, který závisí na jeho frekvenci. Barva předmětu vnímaná pozorovatelem je tedy závislá na fyzikálních vlastnostech předmětu, konkrétně na tom jaké frekvence světla odráží. Na barvu má vliv také spektrum dopadající světla.

Mísení neboli skládání barev může být realizováno dvěma způsoby. Prvním je součtové (aditivní) skládání barev, ve kterém je požadované barevné světlo získáno mísením různého počtu složkových světel. V televizní technice a počítačové grafice jsou složkovými barvami červená, zelená a modrá. Přidáváním barevných složek, vzniknou dle rovnice 2.2 [12] všechny ostatní barvy včetně bílé

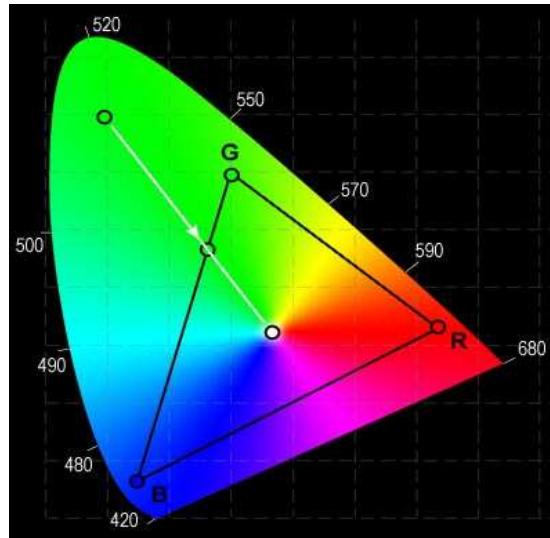
$$U_Y = 0,299 \cdot U_R + 0,587 \cdot U_G + 0,114 U_B. \quad (2.2)$$

Rovnice platí pouze pro referenční bílé světlo C. Pokud, budou všechny složky nulové, bude výsledná barva černá. Další metodou je odčítací (subtraktivní) skládání barev. V tomto případě jsou složkovými barvami žlutá, azurová a purpurová. Požadované barevné světlo se získává z bílého světla odfiltrováním určitých spektrálních složek. Subtraktivní mísení je používáno v tiskovém průmyslu. Oba způsoby skládání barev jsou zobrazeny na obrázku 2.2



Obr. 2.2 Mísení barev (převzato z [1]).

Pro standartizované vyjádření barevného světla byl definován nejprve trojrozměrný prostor RGB (trichromatická soustava). Prostor byl dán volbou tří složkových světel R, B, G vlnových délek:  $\lambda_R = 700 \text{ nm}$ ,  $\lambda_G = 546 \text{ nm}$ ,  $\lambda_B = 435,8 \text{ nm}$ . Bylo provedeno kolorimetrické vyrovnání spektrálních barev a stanoveny kolorimetrické jednotky a průběhů trichromatických činitelů. Bylo definováno referenční bílé světlo. Tento systém má ovšem určité nedostatky, proto byl nahrazen trichromatická soustava XYZ. V systému XYZ jsou jako složková světla zvoleny fyzikálně nerealizovatelná světla. Hlavní výhodou systému XYZ jsou snazší výpočty při mísení barev a kladná hodnota trichromatických souřadnic x, y, z. Kolmým průmětem křivky spektrálních barev z jednotkové roviny kolorimetrického prostoru XYZ do roviny měrných světel (X), (Y), vznikne diagram barev MKO. Více informací o barevných systémech a diagramu barev je možné najít v [11.]



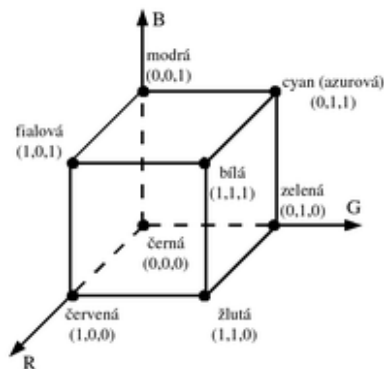
Obr. 2.3 Kolorimetrický obrazec MKO (převzato z [2])

## 2.2 Barevné modely

V programu Peacock se pracuje převážně s modelem RGB. Jedna s použitých metod je však založena na převodu s modelu RGB do modelu HSV a zpět. Informace o barevných modelech byly čerpány z Wikipedie [2].

### RGB model

RGB model je již částečně popsán při definici míchání barev, proto bude na tomto místě jen upřesněno několik jeho základních vlastností. Model RGB je možné zobrazit jako krychli zobrazenou na obrázku 2.4, ve které každá z kolmých hran udává škálu mohutností barevných složek. Potom libovolný bod se souřadnicemi (r, g, b) v této krychli udává hodnotu výsledné barvy. Každá barva je udána mohutností tří základních barev komponent (červené - **r**ed, zelené - **g**reen a modré - **b**lue, odtud RGB). Mohutnost se udává buď v procentech (dekadický způsob) nebo podle použité barevné hloubky jako určitý počet bitů vyhrazených pro barevnou komponentu (pro 8 bitů na komponentu je rozsah hodnot 0 – 255, pro 16 bitů na komponentu je rozsah hodnot 0 – 65535), přičemž čím větší je mohutnost, tím s vyšší intenzitou (světlejší) se barva komponenty zobrazuje.



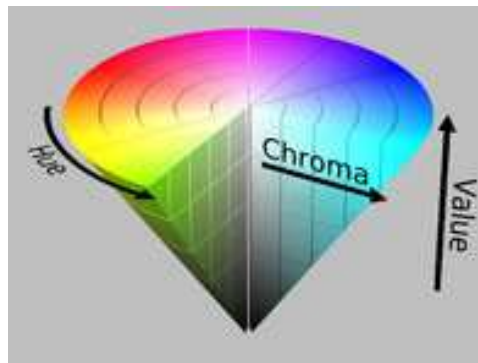
Obr. 2.4 RGB model (převzato z [2])

## HSV model

HSV (Hue, Saturation, Value), také známý jako HSB (Hue, Saturation, Brightness) či HSI (hue, saturation, intenzity), je barevný model, který vytvořil v roce 1978 Alvy Ray Smith. Tento barevný model nejvíce odpovídá lidskému vnímání barev. Model je znázorněn na obrázku 2.5. Model má opět tři složky, u nichž je nutno hlídat hodnoty (možné nesmyslné kombinace):

- **Hue** - barevný tón, převládající. Neboli odstín - barva odražená nebo procházející objektem. Měří se jako poloha na standardním barevném kole ( $0^\circ$  až  $360^\circ$ ). Obecně se odstín označuje názvem barvy.
- **Saturation (chrome)** - sytost barvy, příměs jiné barvy. Někdy též chroma, síla nebo čistota barvy, představuje množství šedi v poměru k odstínu, měří se v procentech od 0% (šedá) do 100% (plně sytá barva). Na barevném kole vzrůstá sytost od středu k okrajům. Např. červená s 50% sytostí bude růžová.
- **Value** - hodnota jasu, množství bílého světla. Relativní světlost nebo tmavost barvy. Jas vyjadřuje, kolik světla barva odráží.

Hlavním důvodem pro využití tohoto modelu je možnost ovlivnění jedné z těchto vlastností při zachování zbývajících. Což u modelu RGB není možné. V praxi to znamená, že můžeme převést RGB složky do HSI prostoru tam změnit barevný tón a převést zpět do RGB, přičemž jas obrazového bodu bude zachován.



Obr. 2.5 Model HSV (převzato z [2])

## 3 METODY ZVÝRAZŇOVÁNÍ OBRAZU

Důvodem a cílem popsaných postupů je zvýšení diagnostické výtěžnosti obrazového snímku. Všechny metody vyplývají z vlastnosti lidského zraku. K zvýraznění šedotónového obrazu používáme metody, které lze rozdělit do několika základních skupin

- modifikace škály šedosti,
- ostření obrazu,
- redukce šumu,
- barevné zvýraznění.

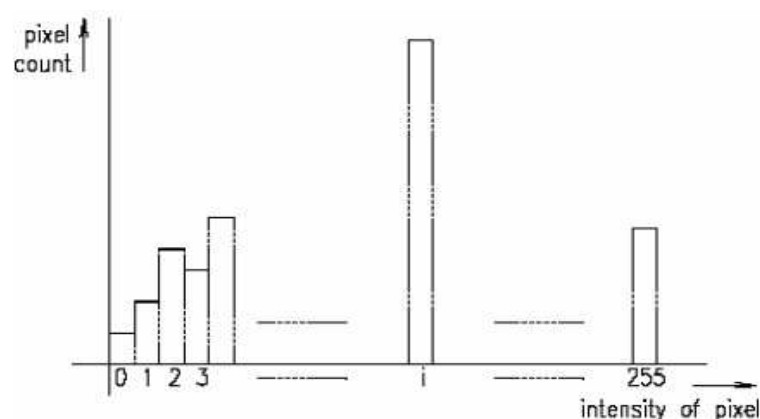
Všechny zmíněné metody budou v následujících kapitolách stručně popsány. Speciální důraz bude kladen na barevné zvýraznění, které je předmětem diplomové práce. Informace v této kapitole byly čerpány z literatury [4], [5] a předmětu Multimediální signály a data.

### 3.1 Základní pojmy

Před vlastním popisem jednotlivých metod je třeba vysvětlit několik pojmů s nimi souvisejících.

Prvním z pojmů je *šedotonový histogram*, který lze definovat jako vektor počtu pixelů (obrazový bod, jejichž maticí je tvořen obraz) na určitých hladinách šedi. Tento pojem lze nejlépe pochopit prohlédnutím obrázku č. 1. Na ose x je vynášena hodnota jasu a na ose y počet pixelů o daném jasu. Součet sloupců musí být roven celkovému počtu pixelů v obraze. Histogram umožňuje udělat si představu o vzhledu snímku a o rozložení jasu jeho pixelů. Tedy zda je obraz spíše tmavý či světlý nebo úrovní jeho kontrastu.

Rozložení histogramu a tím jasy jednotlivých pixelů lze ovlivnit bodovou, lokální nebo globální operací. *Bodová operace* je taková operace, při které má na výstupní pixel vliv jen odpovídající vstupní obrazový bod. Pokud má na hodnotu výstupního pixelu vliv více pixelů vstupního obrazu, pak mluvíme o *lokální operaci*. Vstupní body, které mají vliv na výstup, se nazývají okolím bodu a to se uvažuje zpravidla čtvercové. Situace, kdy má každý bod vstupního obrazu vliv na každý bod výstupu představuje tzv. *globální operaci*.



Obr. 3.1 Histogram (převzato z [5])

## 3.2 Modifikace škály šedosti

Základní myšlenkou metody je zvýšení kontrastu v některé části stupnici. Zvýšením kontrastu samozřejmě dojde k zvýraznění obrazu. Vzhledem k omezenému dynamickému rozsahu hodnot obrazových elementů taková úprava současně vede ke snížení kontrastu v jiné části stupnice. Proto je důležitá správná volba části stupnice, kde bude kontrast zvýšen. K nejlepší možnosti se dochází zpravidla zkušebně, tedy heuristicky. Uživatel má také možnost měnit oblast průběžně, což mu umožní prohlížet různé části obrazu s různě nastaveným kontrastem. Další možností je závislost označovaná jako „zebra“ nebo pilová funkce, skládající se z několika na sebe navazujících strmých úseků, z nichž každý plně využívá přípustného rozsahu výstupních hodnot – tak se dosáhne mimořádného zvýšení kontrastu za cenu vytvoření falešných hran v místech obrazu, kde se stýkají sousední jasové hodnoty dvou takto zdůrazněných původních intervalů jasu. Úpravy jasové stupnice zvyšující kontrast v monochromatickém obraze jsou ve vytvořené aplikaci použity jako přípravné před samotným pseudobarvením. Konkrétní metody implementované v programu jsou popsány v kapitole 3.

### 3.2.1 Ostření obrazu

Na základě výsledků neurofyziologických a psychologických výzkumů je založena metoda ostření obrazu, kterou můžeme nazvat také hledání hran. Výzkumy ukazují, že pro vnímání člověka jsou ve snímku zejména důležitá místa, kde se náhle mění hodnota jasu. Což prakticky znamená, že jsou to místa, kde je velký rozdíl mezi jasy sousedních vektorů. Tyto místa jsou nazývány hrany.

Hledání hran si lze představit jako filtraci, konkrétně jako použití horní propusti. Operace je realizována v prostorové oblasti použitím vhodné lokální operace, což znamená, že je použit vhodný maskový (lokální) operátor, jenž představuje impulzní charakteristiku filtru. Ekvivalentní výsledek je získán také převodem do frekvenční oblasti a v ní použití vhodného filtru typu horní propust. Dalšími úpravami (například prahováním), je možné získat hrubou hranovou reprezentaci obrazu. Hrubá hranová reprezentace obrazu je černobílý obraz, který vznikne z původně šedotónového obrazu a zachycuje pouze hrany obrazu.

### 3.2.2 Redukce šumu

Operace redukce šumu je operací protichůdnou k ostření obrazu. Proces je opět často realizován lokální operací, tentokrát ovšem představuje filtraci použitím dolní propusti. Základní způsob použitím masky představující dolní propust je ovšem jen jednou z možností a pomůže nám při redukce Gausovského šumu (šum s nulovou střední hodnotou). Tento způsob ovšem není účinný, pokud se v obraze objeví impulzní šum tzv. typ sůl a pepř. V takové situaci je nutné použít jiný typ filtru a to mediánový filtr, který je nelineární a pracuje na tom principu, že seřadí pixely spadající pod jeho masku od největšího po nejmenší a reprezentativní pixel nahradí pixelem, který bude ležet uprostřed uspořádané řady.

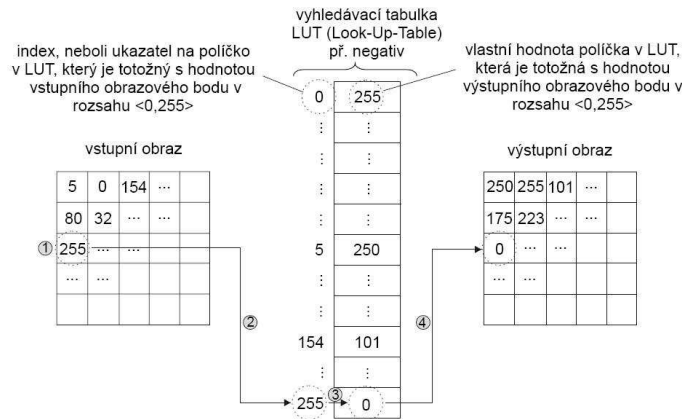
### 3.2.3 Pseudobarvení

Pseudobarvení je účinnou metodou zvýraznění obrazu vedoucí podstatnému zvýšení diagnostické výtěžnosti obrazu. Jedná se zvláštní typ bodového zpracování šedotónového obrazu, kdy převodní funkce nabývá vektorových hodnot a její složky vyjadřují hodnoty barevných komponent. Smyslem této operace je vyjádřit odstíny šedi ve stupnici barev.

Využití této operace vychází z vlastností lidského oka a je založena na skutečnosti, že člověk rozeznává mnohem větší počet barevných odstínů než stupňů šedi. Mimo tuto aplikaci a převod šedotónového obrazu na barevný existuje i příbuzná operace a to takzvaný převod do nepravých barev. Nepravé barvy se u medicínských obrazů také mohou používat pro lepší využití zraku pozorovatele. Protože citlivost čípků lidské sítnice je maximální v zelené oblasti

viditelného záření, je optimální při pozorování např. červeného objektu převést tuto barvu na nepravou zelenou barvu.

Převod do nepravých barev je proveden bodovým operátorem (stejně jako změna škály šedosti) a při stejné hodnotě jasu bude vždy stejná i výstupní hodnota, proto je možné převod prakticky realizovat pomocí převodní LUT tabulky. LUT je zkratkou pro anglický výraz Look Up Table. Tabulka vyjadřuje požadovanou závislost. Jedná se v podstatě o vyhledávací tabulku, kde hodnota jasu obrazového bodu ve vstupním monochromatickém obraze je řádkem tabulky (adresou v paměti) a obsah daného řádku tabulky (hodnota uložená na dané adrese v paměti) je jasnem (parametrem) výstupního obrazu. Implementace LUT v HW i SW je zobrazen na obrázku.



Obr. 3. 1 Implementace LUT tabulky ( převzato z[5])

Samotný princip metody je relativně jednoduchý a srozumitelný. Každý digitální snímek se skládá z tzv. obrazových prvků (pixelů), přičemž každý pixel má určitou číselnou hodnotu. Obraz je tedy reprezentován maticí čísel, přičemž rozměry matice jsou dány jeho rozměry. V případě šedotónového snímku se jedná o jedinou matici a její obsah prezentujeme jako odstíny šedi ve většině případů v rozsahu 0-255. Barevný obraz je však reprezentován trojicí matic, přičemž každá z matic reprezentuje jednu základní barevnou složku (červenou-zelenou-modrou). Přesněji řečeno obsah jednotlivých matic udává zastoupení základních barevných složek v konkrétním pixelu. Program pro pseudobarvení tedy přiřadí každému odstínu šedi v černobílém snímku barevný odstín. Na základě předešlého textu je patrné, že z jedné původní matice reprezentující odstín šedi jsou podle určité tabulky vytvořeny tři matice reprezentující barevný odstín. Konkrétní metody pseudobarvení jsou popsány v kapitole 4.

### 3.2.4 Metoda hustotních řezů

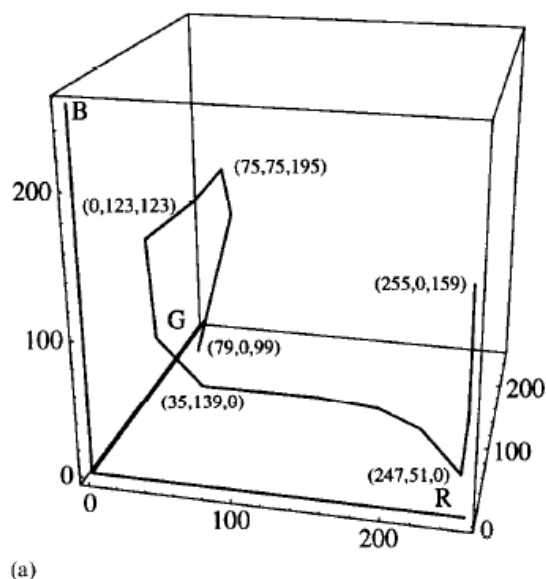
Hustotní řezy (density slicing) patří mezi velmi jednoduché metody zvýraznění obrazu. Jedná se o metodu představující rozšíření koncepce práhování. Z toho vylivají další skutečnosti. Například to, že provedením této metody dojde k zmenšení počtu hodnot v původním obraze do menšího počtu tříd ve výstupním obraze. Dojde tedy k potlačení rozdílů uvnitř definovaných intervalů a zdůrazní se rozdíly mezi intervaly. Počet tříd a určení prahových hodnot vychází z prozkoumání histogramu. Modifikací může vzniknout metoda pseudobarvící. Místo omezení na menší počet jasů může, být pomocí práhování vytvořen barevný snímek s malým počtem barev. V podstatě se jedná o nastavení vhodné tabulky barev. Interval kanálu je rozdělen na podintervaly a mezní hodnoty takto vzniklých intervalů přiřazeny RGB kanálům.

## 4 ZOBRAZOVÁNÍ V NEPRAVÝCH BARVÁCH

Pokud v předchozích kapitolách bylo uvedeno několik různých přístupů a možností k zvýraznění obrazu (hledání hran, jasové transformace, redukce šumu, pseudobarvení). Tak v této kapitole bude ze zmíněných metod vyčleněna metoda zobrazování v nepravých barvách. A budou podrobněji popsány a rozebrány postupy, které lze k převedení snímku do nepravých barev použít. Těchto metod je velké množství, i když některé přístupy se od sebe až tak neliší a jsou postaveny na stejných základech. V následujících kapitolách se zaměříme na metody, které jsou využity v programu Peacock.

### 4.1 Jednoduché parametrické rovnice pro pseudobarvení šedotónových obrazů

Tato metoda na jedné straně zachovává progres jasu, ale přitom vytváří mnoho barevných změn, které pomohou lidskému oku vyhodnotit snímek. Metoda je podrobně popsána v [6].



Obr. 4.1 RGB prostor se zobrazenou parametrickou křivkou (převzato z [6])

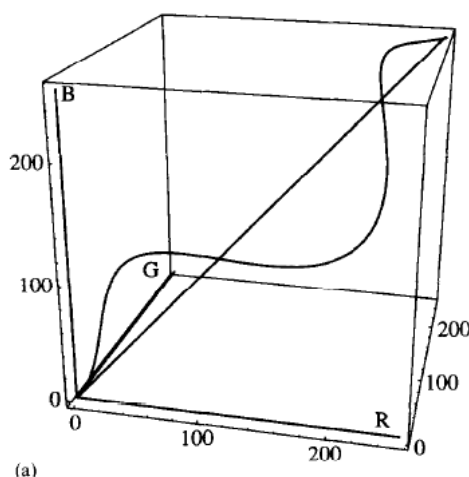
Spirála je formalismus. Pseudobarvení můžeme matematicky popsat pomocí transformace křivky v barevném prostoru (viz obrázek Obr. 4.1). Křivka je vytvořena z ekvidistantně rozložených bodů. Bodů je stejný počet, jako je počet stupňů šedi ve vstupním snímku. Každá hodnota ze vstupní šedotónové stupnice je transformována na konkrétní RGB barvu definovanou souřadnicemi vzorku v barevném prostoru (krychli). Užitím RGB modelu mohou být tyto schémata využity jako LUT tabulky.

Hlavní úhlopříčka v RGB krychli je vlastně spojnicí bílé a černé barvy a proto představuje stupnici šedi (intenzitu jasu). Z toho vyplývá, že pokud chceme realizovat obarvení a přitom zachování kontinuální změny jasu, musí transformační křivka představovat spirálu točící se kolem této diagonály. V třírozměrném spojitém prostoru je spirála kolem osy z, s parametrem t dána vektorem v [6]

$$\vec{v}(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = \begin{pmatrix} r(t) \cdot \sin(\omega t + \varphi) \\ r(t) \cdot \cos(\omega t + \varphi) \\ z(t) \end{pmatrix}. \quad (4.1)$$

Fáze  $\varphi$  definuje směr a frekvence  $\omega$  souvisí s počtem otáček kolem osy z. Funkce  $r(t)$  určuje tvar spirály a funkce  $z(t)$  její vývoj. Pro použití této rovnice pro pseudobarvení musí být mapovány kolem hlavní diagonály RGB krychle (viz. obr 4.2), což může být provedeno pomocí následující matice [6]

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = M \cdot \vec{v}. \quad (4.2)$$



Obr. 4.2 Křivka vyhovující rovnici (převzato z [6])

V další fázi je třeba stanovenit matici M. Jakákoliv rotace kartouzského systému je dána [6]

$$M = \begin{pmatrix} \cos\beta + a_x^2 \cdot (1 - \cos\beta) & a_z \cdot \sin\beta + a_x \cdot a_y \cdot (1 - \cos\beta) & -a_y \cdot \sin\beta + a_x \cdot a_z \cdot (1 - \cos\beta) \\ -a_z \cdot \sin\beta + a_y \cdot a_x \cdot (1 - \cos\beta) & \cos\beta + a_y^2 \cdot (1 - \cos\beta) & a_x \cdot \sin\beta + a_y \cdot a_z \cdot (1 - \cos\beta) \\ a_y \cdot \sin\beta + a_z \cdot a_x \cdot (1 - \cos\beta) & -a_x \cdot \sin\beta + a_z \cdot a_y \cdot (1 - \cos\beta) & \cos\beta + a_z^2 \cdot (1 - \cos\beta) \end{pmatrix}, \quad (4.3)$$

$\beta$  určuje úhel natočení a  $a_x$ ,  $a_y$  a  $a_z$  představují cosinus pevného úhlu mezi osou rotace g a osami x, y, z. Určit tyto cosiny není těžké, protože osa g musí zůstat v rovině (x, y). Viz obrázek 4. 2.

$$a_x = \cos 45^\circ = \frac{1}{\sqrt{2}}, \quad a_y = \cos 135^\circ = -\frac{1}{\sqrt{2}}, \quad a_z = \cos 90^\circ = 0. \quad (4.4)$$

Velikost délky hrany krychle RGB je nastavena na 1. Jednoduchou aplikací Pythagorovy věty

$$a^2 + b^2 = c^2, \quad (4.5)$$

kde  $a$ ,  $b$  jsou délky odvěsen a  $c$  délka přepony. Označíme hranu RGB krychle a stěnovou úhlopříčku  $u$  a objektovou úhlopříčku  $v$ . Pak můžeme počítat jejich délky.

$$\begin{aligned} u &= \sqrt{a^2 + a^2} & v &= \sqrt{a^2 + u^2} \\ u &= \sqrt{1^2 + 1^2} & v &= \sqrt{1^2 + \sqrt{2}^2} \\ u &= \underline{\underline{\sqrt{2}}} & v &= \underline{\underline{\sqrt{3}}} \end{aligned} \quad (4.6)$$

V dalším postupu počítáme s využitím znalosti  $a$ ,  $u$  a  $v$  a Kosinova teorému  $\cos(\beta)$ .

$$\begin{aligned} \cos(\beta) &= \frac{a^2 + v^2 - u^2}{2 \cdot a \cdot v} \\ \cos(\beta) &= \frac{1^2 + \sqrt{3}^2 - \sqrt{2}^2}{2 \cdot 1 \cdot \sqrt{3}} \\ \cos(\beta) &= \frac{1}{\sqrt{3}} \longrightarrow \sin(\beta) = \sqrt{\frac{2}{3}} \end{aligned} \quad (4.7)$$

Dosažením vztahu (4.4) a (4.6) do rovnice (4.3) získáme:

$$M = \begin{pmatrix} \cos\beta + a_x^2 \cdot (1 - \cos\beta) & a_x a_y \cdot (1 - \cos\beta) & -a_y \cdot \sin\beta \\ a_y a_x \cdot (1 - \cos\beta) & \cos\beta + a_y^2 \cdot (1 - \cos\beta) & a_x \cdot \sin\beta \\ a_y \cdot \sin\beta & -a_x \cdot \sin\beta & \cos\beta \end{pmatrix} = \frac{1}{2 \cdot \sqrt{3}} \begin{pmatrix} 1 + \sqrt{3} & 1 - \sqrt{3} & 2 \\ 1 - \sqrt{3} & 1 + \sqrt{3} & 2 \\ -2 & -2 & 2 \end{pmatrix} \quad (4.8)$$

Na závěr předchozí matici dosadíme do vztahu (4.2) za "M" a za "v" dosadíme (4.1) a získáme finální podobu parametrů pro pseudobarvení

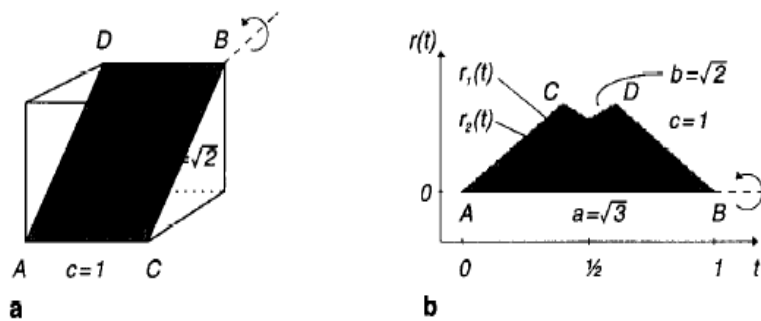
$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \frac{1}{2 \cdot \sqrt{3}} \begin{pmatrix} 1 + \sqrt{3} & 1 - \sqrt{3} & 2 \\ 1 - \sqrt{3} & 1 + \sqrt{3} & 2 \\ -2 & -2 & 2 \end{pmatrix} \cdot \begin{pmatrix} r(t) \cdot \sin(\omega \cdot t + \varphi) \\ r(t) \cdot \cos(\omega \cdot t + \varphi) \\ z(t) \end{pmatrix} \quad (4.9)$$

Kde  $0 < t < 1$ . V zájmu matematické jednoduchosti jsou barevné osy normalizovány k jedné, zatímco na obrázcích jsou v rozsahu 8bit tedy 0 až 255. Ve vztahu (4.9) fáze  $\varphi$  definuje počáteční barvu (barvu, kterou bude nabývat pixel o nulovém jasu). Zatímco frekvence  $\omega$  určuje dynamiku barevné změny. Funkce  $r(t)$  a  $z(t)$  jsou získány pomocí úvahy, která není pro účel této práce zásadní.

$$r(t) = \sqrt{\frac{3}{2}} \cdot \begin{cases} t & 0 \leq t \leq 0,5 \\ 1-t & \text{jindy} \end{cases}, \quad z(t) = \sqrt{3} \cdot t \quad (4.10)$$

Posledními neznámými ve vztahu tedy zůstávají fáze a frekvence. Variace těchto dvou parametrů vede ke vzniku velkého množství barevných spekter.

V poslední fázi je potřeba určit funkce  $r(t)$  a  $z(t)$  Pokud je hlavní diagonála považována za osu otáčení, potom největší vzdálenost  $r(t)$  je omezena zbývajícím objemem rotující RGB krychle. Vzhledem k symetrii krychle je zbývajícím objem tvořen dvěma rovinnými trojúhelníky. Situaci ukazuje obrázek č. 4.3. Vzdálenost  $r(t)$  musí zůstat uvnitř obou trojúhelníků. Z tohoto důvodu je největší sklon  $dr/dt$  ekvivalentní tangentě uhlu  $\gamma$  trojúhelníku ABC. Proto opět použijeme cosinuv teorém (4.7).

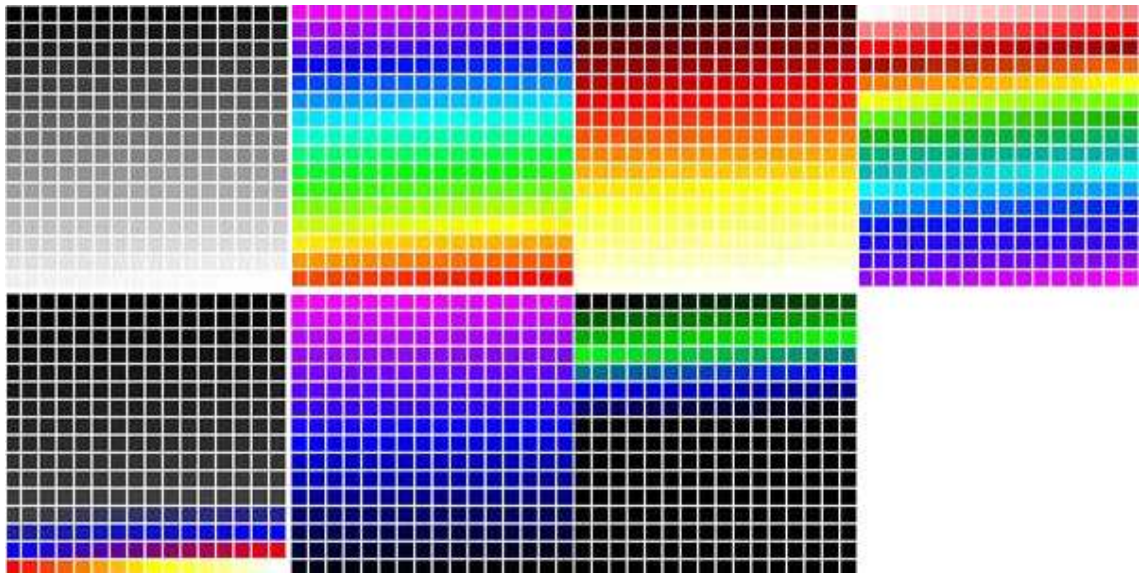


Obr. 4.3 Určení funkcí  $r(t)$  a  $z(t)$  (převzato z [6])

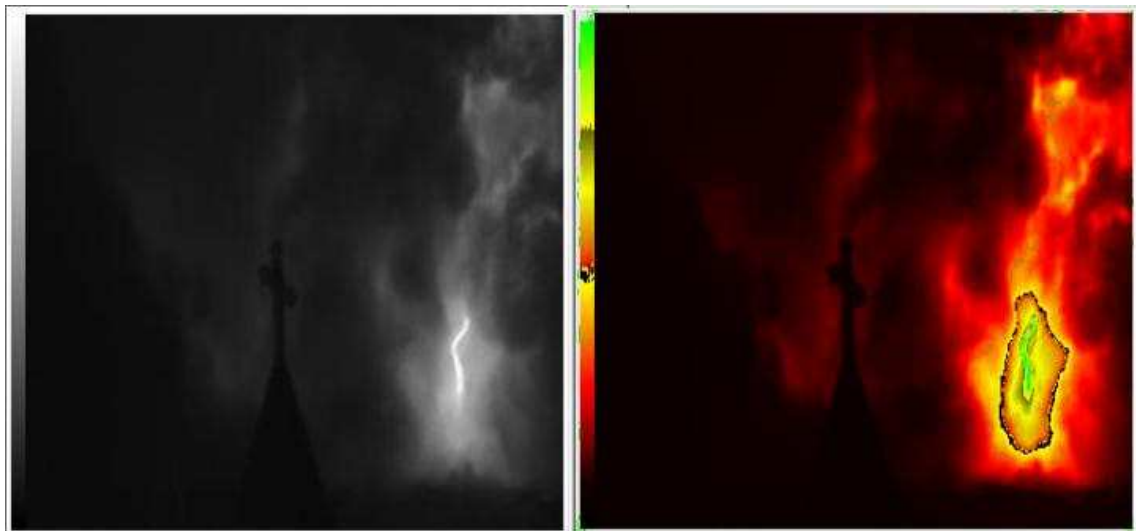
## 4.2 Barevná paleta

Metoda barevné palety pracuje na principu vyhledávací tabulky (LUT). Nepoužívá žádné parametry a jejím jediným vstupem je tak hodnota jasu obrazového bodu vstupního snímku. Základním kamenem metody je tabulka, která má tolik řádků, kolik diskretních jasových hodnot může nabývat pixel vstupního snímku. Tedy  $2^n$  řádků, pokud označíme počet bitů měřicího zařízení  $n$ . Obvykle počet bitů je 8, tedy 256 stupňů jasu a 256 řádků tabulky. Sloupce má tabulka 3, pro každou základní barevnou složku jeden (RGB). Hodnoty v tabulce se mohou pohybovat v rozmezí 0-255 (při 8bitech). Obsah tabulky je předem dán (v programu tabulku představuje dvojrozměrné pole). Principem metody je volba řádku tabulky a tím odpovídající barvy výstupního obrazového bodu v závislosti na hodnotě jasu vstupního obrazového bodu.

Obsah tabulky může být různý a tím může být použito nekonečné množství barevných palet (ukázka obr. 4.4). Při reálném použití ovšem není variabilita této metody nikterak velká. Patří spíše mezi techniky, při jejichž použití není změna barevné palety a tím ovlivnění výsledného obrazu jednoduché. Protože pro změnu palety by bylo nutné, mít uloženo větší počet vyhledávacích tabulek. Problém, tedy je, že paleta lze velmi přesně definovat, ale definici je časově náročná a většinou pro uživatele nepřístupná. Na druhé straně má metoda tu výhodu, že uživatel může velmi přesně ovlivnit výslednou barvu odpovídající konkrétním jasům. A to editací konkrétního řádku tabulky. Například pokud ho zajímá určitý jas, stačí mu nastavit barvu vysoce kontrastní k barvám blízkých jasů. Možnost editace tabulky program Peacock poskytuje. Použití této metody je zobrazeno na obr. 4.5. Obrázek, který je použit pro ukázkou aplikace programu byl získán na stránce [www.youvan.com](http://www.youvan.com)



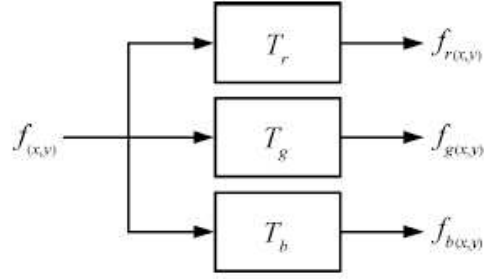
Obr. 4.4 Ukázka různých palet metody využívající pevnou tabulku ( převzato z[9])



Obr. 4.5 Ukázka funkce metody barevná paleta (původní snímek převzat [7])

### 4.3 Metoda založená na stupních jasu

Anglický název skupiny metod, do kterých spadá i metoda popsaná na následujících řádcích, je Gray-level – based (založené na stupních jasu). Tyto techniky potřebují konstrukci transformační funkce, která převádí stupně šedi, gradienty nebo jiné šedotónové funkce na různé barvy. Při jejich použití dojde k převedení všech stupňů šedi na sadu barev pomocí pevného pravidla, bez uvažování kontextových (doplňujících informací). A proto je jejich nedostatkem, že může dojít k přehlédnutí místní informace. Základní myšlenkou je vytvoření tří nezávislých transformací pro každou s barevných složek RGB (4.6).



Obr. 4.6 Princip zobrazení v nepravých barvách založený na stupních jasu (převzato z [8])

Konkrétní metoda použitá v programu Peacock (popsána LI J X a Visvanathanem v [8]) je založená na gradientech, tedy na hranové reprezentaci obrazu a první krok postupu představuje právě vytvoření hranové reprezentace šedotónového obrazu. Hranová reprezentace je vytvořena v prostorové oblasti aplikací lokálních operátorů (4.11a,b). Tyto operátory představují derivaci ve směru osy x a ve směru osy y [8].

$$\begin{matrix} \begin{vmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{vmatrix}, & \begin{vmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{vmatrix}. \end{matrix} \quad (4.11)$$

A v dalším postupu jsou hrany interpretovány ve falešných barvách. Vzhledem k tomu je metoda zvláště hodná k hledání malých rušivých objektů v obrazech s jinak převážně konstantním jasnem a malým množstvím hran. Při aplikaci na snímek s velkým počtem hran se stává obrázek nepřehledný a metoda nepoužitelnou. Obecné rovnice použité pro transformaci do jednotlivých barevných složek uvedl Li v [8] (viz (4.12), (4.13), (4.14)).

$$f_r(x, y) = \begin{cases} 0 & g_1 \leq f_G(x, y) \leq t_1 \\ \left( \sin \left( \frac{\pi \cdot (f_G(x, y) - t_1)}{2 \cdot (t_2 - t_1)} \right) \right)^{k_1} & t_1 \leq f_G(x, y) \leq t_2 \\ 1 & t_2 \leq f_G(x, y) \leq g_2 \end{cases} \quad (4.12)$$

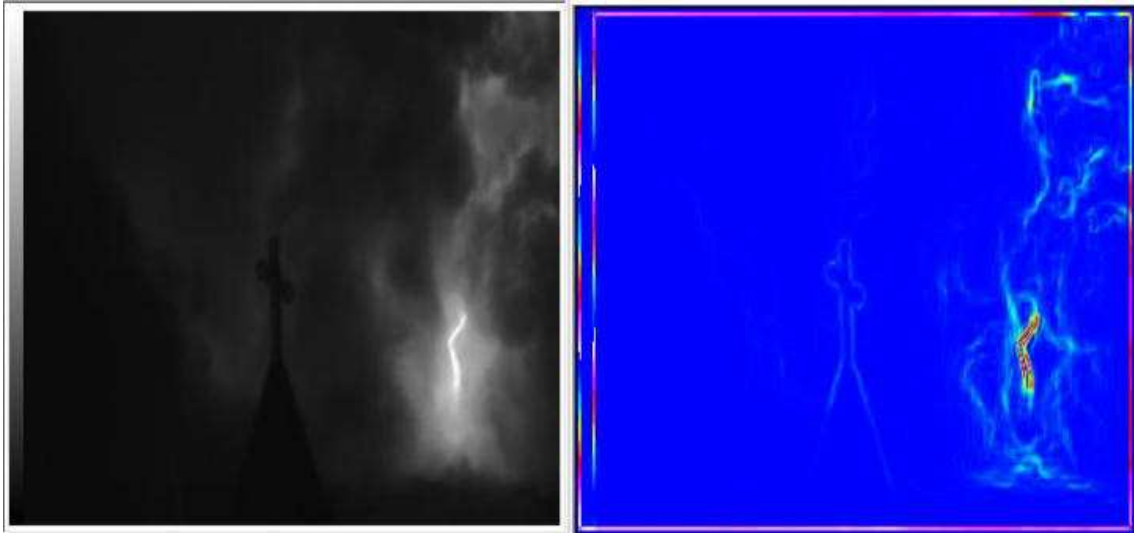
$$f_G(x, y) = \begin{cases} \left( \sin \left( \frac{\pi \cdot (f_G(x, y) - g_1)}{2 \cdot (t_1 - g_1)} \right) \right)^{k_2} & g_1 \leq f_G(x, y) \leq t_1 \\ \left( \sin \left( \frac{\pi \cdot (f_G(x, y) - t_2)}{2 \cdot (t_1 - t_2)} \right) \right)^{k_2} & t_1 \leq f_G(x, y) \leq t_2 \\ \left( \sin \left( \frac{\pi \cdot (f_G(x, y) - t_2)}{2 \cdot (g_2 - t_2)} \right) \right)^{k_2} & t_2 \leq f_G(x, y) \leq g_2 \end{cases} \quad (4.13)$$

$$f_B(x, y) = \begin{cases} \left( \sin \left( \frac{\pi \cdot (f_G(x, y) - t_1)}{2 \cdot (g_1 - t_1)} \right)^{k_3} \right) & g_1 \leq f_G(x, y) \leq t_1 \\ 0 & t_1 \leq f_G(x, y) \leq t_2 \\ \left( \sin \left( \frac{\pi \cdot (f_G(x, y) - t_2)}{2 \cdot (g_2 - t_2)} \right)^{k_3} \right) & t_2 \leq f_G(x, y) \leq g_2 \end{cases} \quad (4.14)$$

kde

$$\begin{cases} t_1 = g_1 + \frac{g_2 - g_1}{3} \\ t_2 = g_2 + \frac{g_2 - g_1}{3} \end{cases}, \quad (4.15)$$

$t_1$  a  $t_2$  jsou dolní respektive horní práh gradientního obrazu  $f_g(x, y)$ . Zatímco  $g_1$  je minimum a  $g_2$  maximum  $f_g(x, y)$ . Velikostí konstantních parametrů  $k_1$ ,  $k_2$ ,  $k_3$  lze ovlivnit omezení transformačních křivek. Výstupy rovnic jsou jednotlivé hodnoty složek RGB. Na obrázku 4.6 je znázorněna aplikace metody na reálném obrázku.



Obr. 4.7 Ukázka aplikace metody založené na stupních jasu (původní snímek převzat [7])

## 4.4 Metoda založená na barevných prostorech

Dalším přístupem k pseudobarvení jsou metody založené na barevných prostorech. Tyto metody nespolehnou jen na úrovně šedi, ale uvažují také sytost a další vlastnosti. Barevný prostor je vizuální interpretace našeho vizuálního vnímání. Téma barevných prostor je podrobněji probráno v kapitole 2. Další z použitých metod je založena na modelu HSI (H odstín, S sytost, I intenzita), který umožňuje separátní úpravu jednotlivých parametrů. Tento model je pro svoje výhodné vlastnosti používán pro pseudobarvení velmi často. Vztahy a podrobný popis konkrétní metody použité v programu jsou popsány v [8] Parametry H, S, I tohoto barevného prostoru získáme z jasu  $f(x, y)$  šedotónové snímku dle vztahů (4.16).

$$\left\{ \begin{array}{l} I = f(x, y) \\ H = \frac{2 \cdot \pi \cdot f(x, y)}{L} \\ S = \begin{cases} k \cdot f(x, y) & f(x, y) \leq L/2 \\ k \cdot (L - f(x, y)) & f(x, y) > L/2 \end{cases} \end{array} \right. \quad (4.16)$$

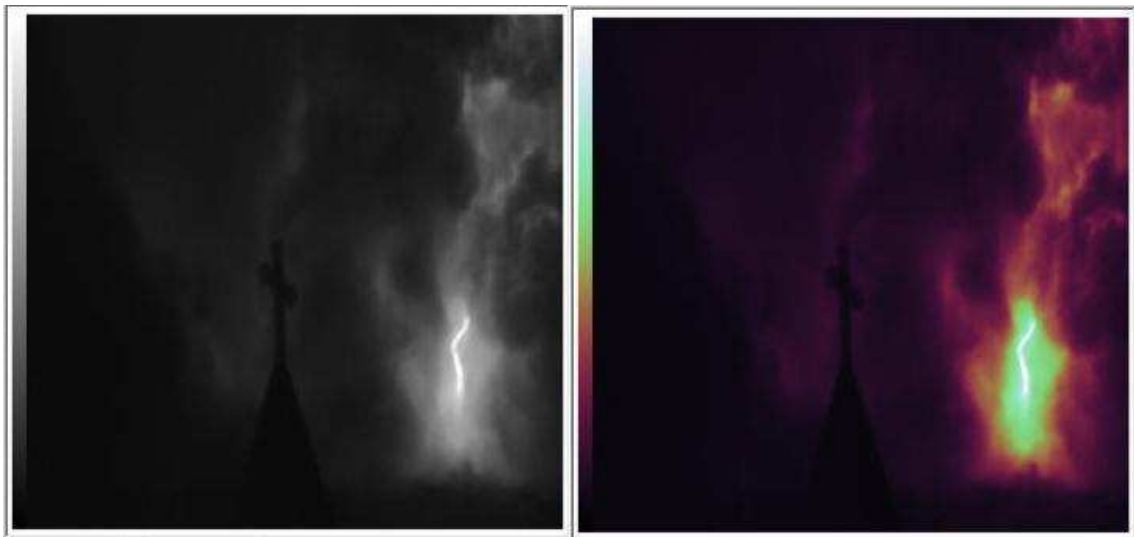
Kde L je maximální hodnota jasu ve vstupním obraze a "k" je proměnný parametr ovlivňující proces (často k cca 1,5). Následujícím krokem bude převedení z prostoru HSI do barevného modelu vhodného pro zobrazení. Ke konverzi jsou použity vztahy (4.17) a (4.18) [8]

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & -0,204124 & 0,612372 \\ 1 & -0,204124 & -0,612372 \\ 1 & 0,408248 & 0 \end{bmatrix} \cdot \begin{bmatrix} I \\ V_1 \\ V_2 \end{bmatrix} \quad (4.17)$$

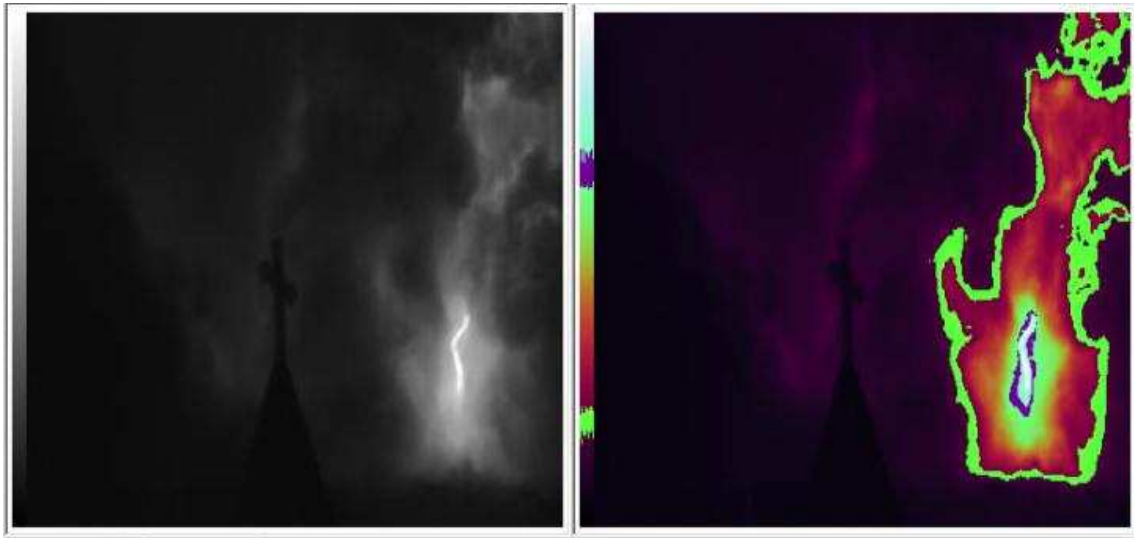
Kde

$$\begin{cases} V_1 = S \times \cos H \\ V_2 = S \times \sin H \end{cases} \quad (4.18)$$

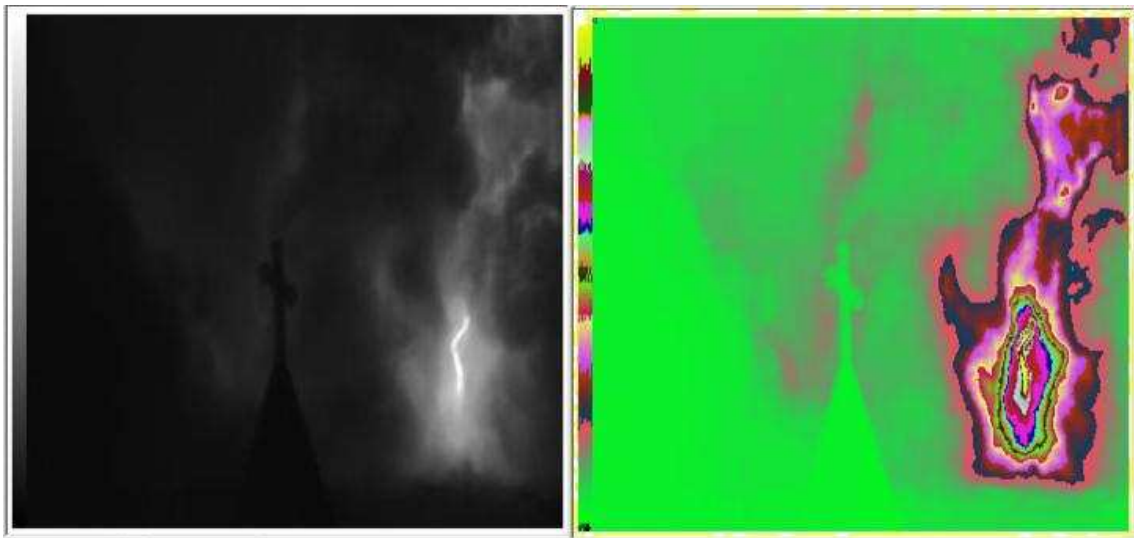
Přístupy založené na barevném prostoru HSI, než tradiční metody, pokud mají snímky relativně dobrou sytost. V takovém případě, zůstávají obvykle odstíny relativně konstantní v přítomnosti stínů a jiných světelných změn. V důsledku toho bude metoda založená na HSI fungovat u těchto snímků dobře. Nicméně když mají snímky nízkou sytost barev, může být obtížné přesně určit odstín. Například, když je nulová sytost, odstín není definován. V takových případech nebude použit HSI pro systémy, které musí být schopné rozlišit všechny barvy, vhodné. Použití této metody je zobrazeno na obrázcích 4.8, 4.9 a 4.10. Na obrázcích je zachycen účinek různého nastavení parametru k (vztah 4.16).



Obr. 4.8 Ukázka aplikace metody založené na převodu do HSI modelu k =1 (původní snímek převzat [7])



Obr. 4.9 Ukázka aplikace metody založené na převodu do HSI modelu  $k = 1,6$  (původní snímek převzat [7])



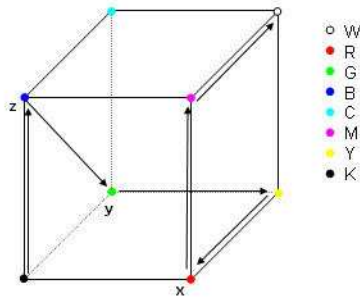
Obr. 4.10 Ukázka aplikace metody založené na převodu do HSI modelu  $k = 6$  (původní snímek převzat [7])

## 4.5 Pseudobarvicí krychle

Tato metoda pro zobrazování v nepravých barvách byla vyvinuta v roce 1990 v laboratořích Youvan jako schéma pro pseudobarvení obrazů bakteriální kolonie získaných digitálním spektrofotometrem. Metoda převzata z [7]. Nízké hodnoty jasu jsou mapovány na černé a modré odstíny. Světlejší obrazové body jsou mapovány postupně přes zelenou, žlutou, červenou, růžovou k bílé a jak je patrné z barevné krychle níže na obr. 4. 11. Zvyšování hodnoty R, G a B je zobrazeno pomocí šipek. V levém dolním rohu krychle je černá (0,0,0), a pravém horním rohu je bílá (255, 255, 255).

Názorný popis principu tvorby barev odpovídající jednotlivým jasům je popsán v tab. 4.1. V prvním sloupci jsou zapsány hodnoty jasu obrazových bodů ve vstupním obraze. V dalších třech barvách je zaznamenán vývoj jednotlivých barevných složek v průběhu rozsahu. Pokud na daném rozsahu složka roste, tak při jasů odpovídajícímu maximu rozsahu dosahuje

maximální sytosti, tedy hodnoty 255. Naopak pokud složka klesá, tak při jasů odpovídajícímu maximu rozsahu dosáhne nulové hodnoty. V pátém sloupci je zapsána výsledná barva na konci rozsahu, tedy v rozích krychle (je použito aditivní mísení barev).



Obr. 4.11 Barevný prostor s vyznačením vývoje barev v závislosti na jasů (převzato z[7])

Tab. 4.1 Tvorba barev v barevném prostoru RGB při použití metody Pseudobarvící krychle

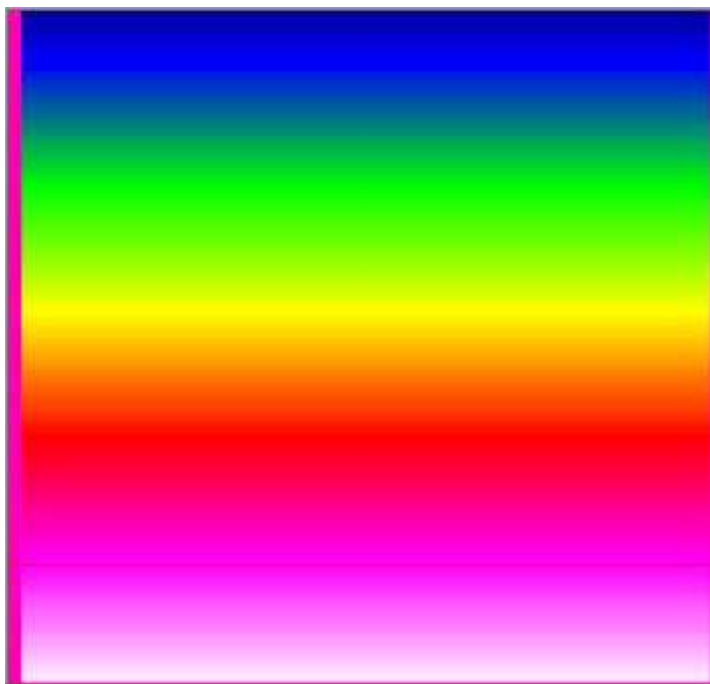
Rozsah jasů	červená	modrá	zelená	Výsledná barva
0	0	0	0	černá
1-42	0	roste	0	Modrá
43-84	0	klesá	roste	zelená
85-128	roste	0	255	žlutá
129-170	maximum	0	klesá	červená
171-212	maximum	roste	0	růžová
213-255	maximum	maximum	roste	bílá

V nejjednodušším případě se mění hodnota barevné složky při změně jasů o jeden stupeň o hodnotu 255/42. Pokud označíme hodnotu jasů v bodě daném souřadnicemi  $x, y$   $I(x, z)$  a výstupní složky barev  $R(x, y)$ ,  $G(x, y)$  a  $B(x, y)$ . Potom platí vztah (4.19), přičemž původní jas  $I(x, z)$  musí být před dosazením vždy upraven tak, aby nabýval v každém rozsahu maximálně hodnotu 42. Což je docíleno tím, že od vstupní hodnoty je odečtena dolní mez daného rozsahu. Pro názornost je ukázán i příklad převodu několika jasových hodnot.

$$R(x, y) = \frac{I(x, z) \cdot 255}{42}. \quad (4.19)$$

Při výchozím režimu této metody je nulový jas transformován na barevné složky  $R = 0$ ,  $G = 0$  a  $B = 0$ . Zatímco maximální jas na barevné složky  $R = 255$ ,  $G = 255$  a  $B = 255$ . Zároveň každá barevná kombinace RGB odpovídá právě jednomu jasů vstupního rozsahu, což je patrné z příslušné barevné palety zobrazené na obr. 4.12. Metodu lze však modifikovat dvěma způsoby a tím opět získat širokou nabídku barevných palet.

První modifikaci představuje změna frekvence. Frekvence je definována jako počet jasových hodnot, kterým odpovídá shodná výstupní barva. Při jejím zvětšení dojde k tomu, že původní paleta se během rozsahu několikrát zopakuje. Modifikace představuje umělé navýšení kontrastu v obraze a je ekvivalentní s úpravou jasové stupnice funkcí pila. Změna barevné palety v závislosti změny frekvence je zobrazena na obr. 4.13



Obr. 4.12 Ukázka barevné palety získané metodou RGB barvicí krychle

Druhou možností jak tuto metodu modifikovat je změna počáteční souřadnice v prostoru RGB odpovídající nulovému jas. Tedy nastavení nenulových složek RGB odpovídající nulovému jas.

## 4.6 Hustotní řezy

Princip metody hustotních řezů je založen na prahování a ve své podstatě se blíží metodě Barevná paleta. Jednoduché jednopruhé prahování je porovnávání okamžité hodnoty proměnné, kterou je v našem případě jas obrazového bodu, s předem určením prahem. Kromě prahu je potřeba také určit hodnoty, které bude nabývat výstupní proměnná v obou případech (pokud překročí práh nebo pokud nepřekročí práh). Tyto hodnoty bývají často rovny maximální a minimální hodnotě rozsahu proměnné. Hustotní řezy se liší od základního prahování tím, že má víc prahů. V limitu s 256 vzniká v podstatě metoda Barevné palety. Prahy v aplikaci nejsou zvoleny náhodně, ale vhodně pro medicínské obrazy na základě měření [10].

## 5 ÚPRAVY HODNOT JASU

Hlavním cílem aplikace je zvýšení diagnostické výtěžnosti snímků. Nejde tedy jen o to nějakým způsobem zobrazit původně šedotónový obraz v nepravých barvách, ale jde také o to, aby tento proces pomohl vyhodnotit informaci, kterou obraz poskytuje. Z daného důvodu program mimo několika různých metod samotného kalorizování snímků poskytuje uživateli také možnosti upravení hodnot jasu snímku, před samotnou transformací do barev. Tyto úpravy mohou také vést k zlepšení diagnostické výtěžnosti a k zvýraznění důležitých detailů.

Transformace hodnot jasu můžeme dělit do dvou základních skupin. První skupina se nazývá jasové korekce. Jas bodu ve výstupním snímku u nich závisí na hodnotě jasu na odpovídajícím místě ve vstupním snímku. V určitých případech může záviset i na jasu obrazových bodů v malém definovaném okolí daného bodu. Druhou skupinou je úprava jasové stupnice. Tyto metody transformují hodnotu jasu pixelu ve vstupním obraze na jinou hodnotu jasu ve výstupním obraze a to nezávisle na jeho poloze. Metody, které používá program Peacock, patří do druhé skupiny úprav. Tyto postupy můžeme současně zařadit mezi bodové operátory.

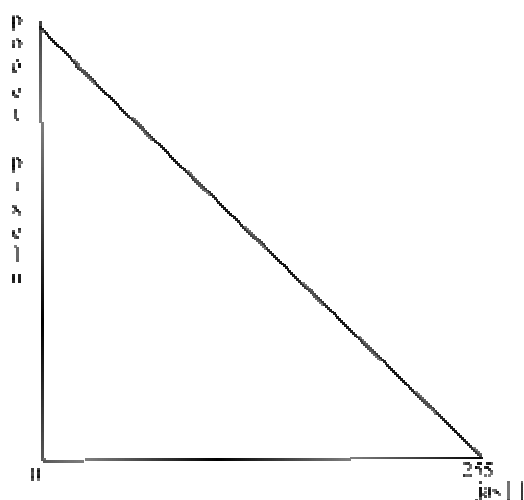
Důležitost transformací jasové stupnice je právě v okamžicích, kdy výsledný obraz vyhodnocuje lidský pozorovatel a snaží se o interpretaci vizualizovaného snímku. Transformace jasové stupnice vedou ke zvýšení kontrastu, které je potřebné z toho důvodu, že měřicí (snímací) zařízení mají relativně velký dynamický rozsah měřených hodnot (rozdíl mezi minimální a maximální možnou hodnotou pixelu), ale jasové hodnoty pixelů v konkrétních obrazech této dynamiky málokdy využívají a zaujímají pouze malou část stupnice. Snímaná data tak mají malý kontrast. Pomocí transformace jasu může být docíleno ke zvýšení kontrastu ve vybraných částech jasové stupnice (zvýraznění části histogramu či pilovitá funkce) nebo v celém obraze jak je tomu například u operace egalizace (vyrovnání histogramu). Kromě výhod má tato skupina úprav snímku, také své nebezpečí. V situacích, kdy jsou snímky podrobeny automatické analýze, nemají žádný smysl. Naopak jejich použití může dojít ke ztrátě informace, pokud není možné provést inverzní zpětnou transformaci. Transformace jasové stupnice jsou technicky snadno realizovatelné LUT tabulkou. Při popisu a implementaci metod do aplikace bylo čerpáno z [10], [12].

### 5.1 Negativ

Vytvoření negativu snímku patří mezi základní a velmi snadné transformace jasové stupnice. Jedná se o invertování hodnot jasu a je dána vztahem [12]

$$q(i) = 1 - f(x, y), \quad (5.1)$$

kde  $q(x, y)$  jsou jasy výstupního obrazu a  $p(x, y)$  jasy vstupního obrazu. Charakteristika transformace negativ je zobrazena na obr. 5.1 a vzhled reálného obrazu před a po transformaci jsou zachyceny na obr. 5.2.



Obr. 5.1 Charakteristika vytvoření negativu

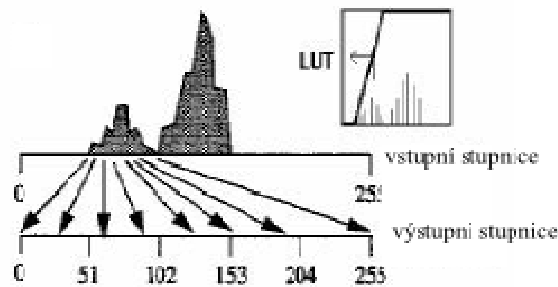


Obr. 5.2 Ukázka aplikace negativu, vlevo původní snímek (převzato z[7]) vpravo po provedení transformace jasové stupnice v programu

## 5.2 Zvýraznění části stupnice

Tato technika je velmi účinná v obrazech, kdy jas většiny obrazových bodů snímku dosahuje blízkých hodnot. Popřípadě pokud je takovýchto shluků často se vyskytujících se jasů více. Takovéto rozložení jasů v obraze vede k malému kontrastu obrazu a tím k obtížnému vyhodnocování snímku. Metoda docílí toho, že část hodnot ze snímku je přiřazena celému rozsahu výstupních hodnot. Základním principem metody je roztažení jednotlivých částí rozsahu stupnice, tak aby bylo využito celého dynamického rozsahu. Diagnostik s využitím histogramu vybere dolní a horní mez části stupnice, kterou chce zvýraznit. Poté je vytvořena nová stupnice, ve které je zvolenému minimum přidělen jas roven nule a maximum hodnota jasu 255. Všechny hodnoty menší než zvolené minimum a větší než zvolené maximum jsou v tento okamžik zanedbány (potlačeny). Hodnotám menším než zvolené minimum je přidělen nulový jas. Hodnotám větším než maximum maximální jas, tedy v 8bitovém rozsahu jasová hodnota 255. S využitím tohoto postupu jsou i původně malé rozdíly jasu ve výstupním

obrazu velmi zvýrazněni. Obrázek 5.3 názorně zobrazuje tento postup a na obrázku 5. 4 je zobrazeno praktické využití této metody [10].



Obr. 5.3 Princip zvýraznění části jasové stupnice (převzato z [10])



Obr. 5.4 Aplikace metody roztažení části jasové stupnice, vlevo původní přesvětlený snímek (převzato z [2]) vlevo upravený

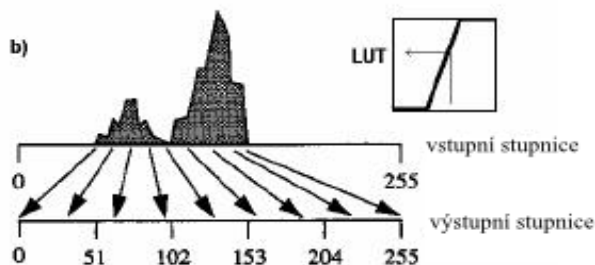
### 5.3 Lineární zvýraznění kontrastu

Použití této metody má význam v obrazech, které nevyužívají celou dynamiku stupnice. Tedy pro snímky, ve kterých se nevyskytuje maximální nebo maximální hodnota jasu. Využitím vztahu (5.2) dojde k lineárnímu rozdělení původních hodnot jasu ve vstupním rozsahu, tak aby využili celý rozsah výstupního zařízení (většinou 8bitů -> hodnoty jasu 0-255) viz obr. 5.5. Pixely, které měli ve vstupním obraze blízké hodnoty jasu, jsou tak ve výstupním obraze dostatečně kontrastní [10].

$$O(x) = \frac{255 \cdot [I(x) - I_{\min}]}{I_{\max} - I_{\min}} \quad (5.1)$$

Kde  $O(x)$  je výstupní hodnota jasu,  $I(x)$  představuje vstupní hodnotu jasu,  $I_{\min}$  je minimální hodnota jasu ve vstupním obraze a  $I_{\max}$  je maximální hodnota jasu ve vstupním obraze. Krátkým rozborem vztahu zjistíme, že technika nemá žádný dopad na snímky, ve kterých  $I_{\max}$  je rovno 255 a  $I_{\min}$  je rovno 0, v takovém případě  $O(x) = I(x)$ .

Účinek metody je tím výraznější čím menší dynamiku  $D$  ( $D = I_{\max} - I_{\min}$ ) má vstupní obraz. V okamžik kdy je dynamika například  $D = 50$ , dosahuje rozptyl původně sousedních jasů ve vstupním obraze 5 stupňů stupnice. Na obr. 5.6. je zobrazena aplikace metody na reálný snímek [10].



Obr. 5.5 Princip lineárního zvýraznění kontrastu (převzato z [10])



Obr. 5.6 Ukázka aplikace metody lineární zvýraznění kontrastu, vlevo původní přesvětlený (převzato z [2]) snímek vlevo upravený

## 5.4 Ekvalizace

Ekvalizace neboli vyrovnání histogramu je metoda vylepšující kontrast obrazů na základě změny rozložení jasových hodnot. Metoda je využívána velmi často, protože poskytuje optimální zvýšení kontrastu a tím diagnostické výtěžnosti snímku. Po provedení ekvalizace je teoreticky zajištěno stejné četné zastoupení jednotlivých jasových úrovní. Ideálním stavem je situace, kdy po ekvalizaci jsou všechny četnosti absolutně stejné. K tomuto stavu však nemůže v digitálním obraze (signálu) nikdy dojít. Ekvalizace zvýší kontrast v blízkosti maxim histogramu a sníží kontrast v oblasti minim histogramu. Což znamená, že zvýší kontrast okolí v těch jasových hodnot, které se v obraze často vyskytují. Využití této metody je stejně jako lineární transformace histogramu užitečné u snímků, které nevyužívají celou dynamiku měřicího potažmo zobrazovacího zařízení. Velké využití nalezneme také u všech snímků, které jsou příliš tmavé nebo naopak příliš světlé (přeexponované). Ekvalizace je také používána v LCD obrazovkách, které mají menší dynamiku obrazu. Výsledek použití ekvalizace je ukázán na obrázku 5.7.

Vzhledem k širokému využití bude užitečné uvedení odvození postupu (vztahu) ekvalizace histogramu [13]. Označíme-li histogram vstupního snímku  $H(p)$  a jeho jasovou stupnici  $p \langle p_o, p_k \rangle$ , potom cílem procesu je nalezení monotónní transformace jasové stupnice  $q = \Gamma(p)$ , která zajistí rovnoměrnost histogramu výstupního snímku pro celý výstupní interval jasů  $q \langle q_o, q_k \rangle$ . Z požadavku na monotónnost zobrazení  $\Gamma$  plyne

$$\sum_{i=0}^k G(q_i) = \sum_{i=0}^k H(p_i) \quad (5.2)$$

Sumy můžeme chápat jako diskrétní distribuční funkce. Ekvalizovaný histogram  $G(q)$  odpovídá rovnoměrnému rozložení  $f$ , jenž má konstantní hustotu pravděpodobnost

$$f = \frac{N \cdot M}{q_k - q_o}, \quad (5.4)$$

kde  $N$  a  $M$  jsou rozměry snímku (v diskrétním provedení počet pixelů). Následně dosadíme výsledek rovnice (5.4) za levou stranu vztahu (5.3) a při představě, že je histogram spojitý můžeme napsat

$$N \cdot M \cdot \int_{q_o}^q \frac{1}{q_k - q_o} ds = \frac{M \cdot N(q - q_o)}{q_k - q_o} = \int_{p_o}^p H(s) ds$$

$$q = \Gamma(p) = \frac{q_k - q_o}{M \cdot N} \int_{p_o}^p H(s) ds + q_o, \quad (5.5)$$

kde  $q$  je hledaná transformace. Integrál v rovnici se nazývá kumulativní histogram. V digitálních obrazech ho můžeme nahradit sumou, potom ekvalizace diskrétního obrazu je dána vztahem [13]

$$q = \Gamma(p) = \frac{q_k - q_o}{M \cdot N} \sum_{i=p_o}^p H(s) + q_o \quad (5.6)$$



Obr. 5.7 Ukázka účinku ekvalizace, vlevo přeexponovaný snímek (převzato z [2]) a vpravo tentýž snímek po aplikaci egalizace

## 5.5 Gama korekce

Gama korekce vznikla v minulosti jako reakce na potřebu zlinearizování nelineární závislosti televizních CRT obrazovek. V současné době se však v oblasti úpravy statických snímků používá zejména pro úpravu špatné expozice, kdy je část objektu v příliš tmavé nebo naopak příliš tmavé části obrazovky. Běžné je použití této funkce k úpravě osvětlení rentgenových snímků. Matematicky lze korekci vyjádřit vztahem [12]

$$g = c \cdot f^\gamma. \quad (5.7)$$

Kde vstupní hodnoty  $f$  jsou v rozsahu  $f \in \langle 0,1 \rangle$ . Rozhodující pro funkci je velikost  $\gamma$ , pokud je  $\gamma$  větší jak 1 bude korekce výsledný efekt ztmavení snímku a naopak pokud je větší jak 1 bude mít aplikace korekce za účinek zesvětlení snímku. Uživatel má v aplikaci možnost hodnotu  $\gamma$  nastavit. Na obrázcích 5.8 a 5.9 jsou zobrazeny obě možnosti využití gama korekce. Podobného účinku lze dosáhnout aplikací zvýraznění části histogramu[12].



Obr. 5.8 Ukázka gama korekce při  $\gamma = 0,5$ . Vlevo původní snímek a jeho podoba v nepravých barvách. Vpravo upravený snímek a jeho podoba v nepravých barvách

# 6 ZÁKLADNÍ STRUKTURA PROGRAMU PEACOCK

K tvorbě programu bylo použito využitě volně šiřitelné prostředí Visual studio 2008 a programovací jazyk C#. Uživatel ovládá program prostřednictvím grafického rozhraní, tvořeného základním stavebním grafickým prvkem Form, který představuje pracovní okno odpovídající klasické oknové konvenci operačního systému Windows. Zdrojový text programu je rozdělen do několika základních unitů, které budou v této kapitole popsány. Unity mají příponu cs. Rozdělení zdrojového textu do více jednotek slouží k zvětšení přehlednosti kódu a jeho snadnější editaci v budoucnosti.

## 6.1 Základní struktura

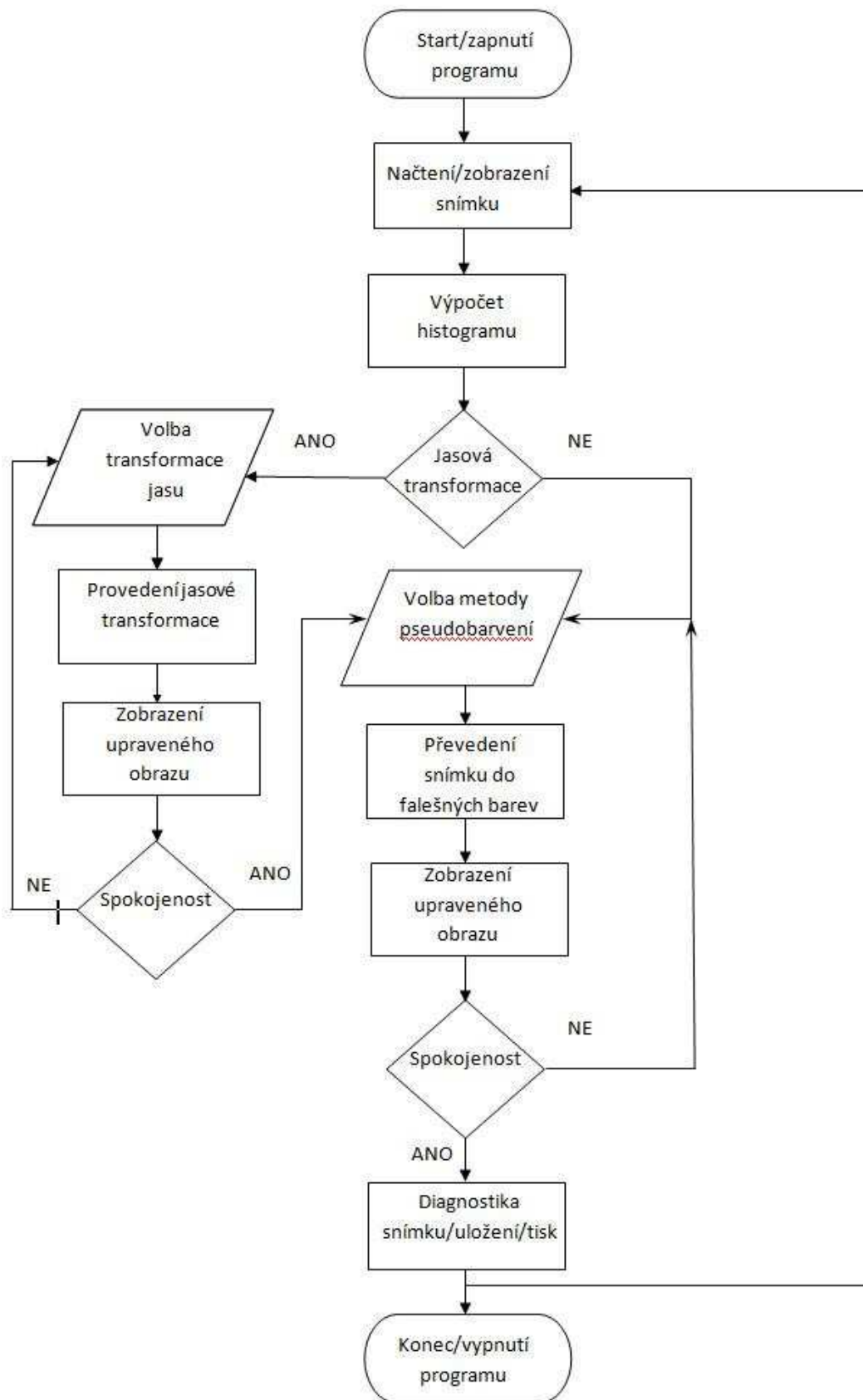
Na obrázku 6.1 je zobrazen vývojový diagram práce programu. Diagram znázorňuje jednotlivé kroky uživatele při práci s programem a nevěnuje se činnosti (algoritmu) samotné aplikace v konkrétních krocích postupy. Vývojové diagramy provádění základních operací jsou zobrazeny na obrázcích 6.2 a 6.3

Po spuštění aplikace Peacock je samozřejmě načtení snímku, který chce uživatel zobrazit v nepravých barvách. Snímek je po načtení automaticky zobrazen a jeho data původně ve formátu bitmap převedena na dvojrozměrné pole typu byte vhodné k další práci. Vzhledem k faktu, že rozhodování znalého uživatele o využití transformaci jasové stupnice je závislé na histogramu snímku, je dalším krokem v postupy výpočet a zobrazení histogramu (výpočet histogramu).

Na základě zobrazení histogramu se může uživatel rozhodnout, zda je nutné zvýšit kontrast snímku nebo jestli přistoupí přímo k zobrazení v nepravých barvách (jasová transformace ANO/NE). V tento okamžik dochází k větvení vývojového diagramu.

V případě, že se uživatel rozhodne použít transformaci jasové stupnice pro zvýšení jasu, je jeho dalším krokem volba, kterou z transformací k tomu využije (další větvení běhu programu, které však není v tomto celkovém diagramu znázorněno). Po zvolení konkrétní transformace je metoda aplikována na snímek a výsledný i jeho histogram zobrazen.

Po zobrazení snímku uživatel subjektivně vyhodnotí zlepšení kontrastu snímku a rozhodne se, zda je spokojen a přistoupí již k zobrazení v nepravých barvách nebo jestli vyzkouší upravit snímek pomocí jiné transformace jasové stupnice (spokojenost ANO/NE). Jestliže je spokojen, tak v dalším kroku rozhoduje, jakou z nabízených metod zvolí pro pseudobarvení. Metoda je aplikována na snímek a výstupní obraz ve falešných barvách je zobrazen. Uživatel má několik možností. Může snímek hned diagnostikovat, uložit jej na pevný disk nebo vytisknout. Následně může začít upravovat další obraz nebo ukončit práci s programem.



Obr. 6.1 Vývojový diagram práce s programem

## 6.2 Grafické rozhraní

Uživatel má k dispozici tři pracovní plochy a to Hlavní okno, Transformační tabulka a Video, mezi kterými se může pohybovat. Tyto pracovní plochy jsou realizovány pomocí komponenty *TabControl* a jednotlivé plochy jsou představovány komponentou *TabPage*.

Pro zobrazení veškerých grafických dat, tedy histogramu, vstupního šedotónového snímku, upraveného šedotónového snímku, obarveného snímku i promítání videosekvence slouží *PictureBox*. Konkrétně je použita základní vlastnost této komponenty a to *Image*. Další zajímavou a použitou vlastností *PictureBoxu* je *SizeMode*, která definuje přízpusobení zobrazovaného snímku velikosti okna komponenty.

Nejdůležitějším ovládacím prvkem je tlačítko tedy komponenta *Button* a její událost *OnClick*. Parametry transformací mění uživatel pomocí zadávání hodnot do komponenty *textBox*. Aplikace přistupuje k těmto datům pomocí vlastnosti komponenty *Text* (*textBox1.Text*). Pro tvorbu vlastní barevné palety uživatele je použita komponenta *GridView*

Dalšími viditelnými komponentami jsou *TrackBar* sloužící pro nastavení proměnných při práci s histogramem, k čemuž je využita jejich vlastnost *Value* (*tBmin.Value*). Rozbalovací menu sloužící k výběru metody pseudobarvení či šedotónové transformace byla použita komponenta *ComboBox*. Je pracováno s její vlastností *Itemindex*. A nakonec zatrhávací tlačítko *CheckBox* a k zobrazení popisných textů komponenta *Label*.

Velmi důležité jsou také neviditelné komponenty, které sice netvoří grafické prostředí, ale program by bez nich nemohl správně fungovat. *OpenFileDialog* slouží k načtení souboru z pevného disku v případě aplikace Peacock konkrétně k načtení vstupního snímku. Naopak *SaveFileDialog* slouží k uložení výstupního snímku na disk. K nastavení tiskové úlohy a vytisknutí snímku je použita komponenta *PrintDialog*. A nakonec k ovládání rovnoměrného snímkování při práci s videosekvencí je využita komponenta *Timer*.

## 6.3 Form1.Designer.cs

Prvním z cs souborů je *Form1.Designer*, ve kterém jsou definovány ve formě zdrojového kódu všechny vlastnosti komponent, jež projekt obsahuje. Tento kód je možné tvořit automaticky úpravou nastavení jednotlivých komponent v panelu *Properties* nebo přímo psát v textové formě. Pro ukázkou je zde ukázána definice tlačítka *Button* sloužícího pro uložení výstupního snímku.

```
// btnSave
//
this.btnSave.Location = new System.Drawing.Point(1079, 375);
this.btnSave.Name = "btnSave";
this.btnSave.Size = new System.Drawing.Size(84, 20);
this.btnSave.TabIndex = 37;
this.btnSave.Text = "Ulož";
this.btnSave.UseVisualStyleBackColor = true;
this.btnSave.Click += new
System.EventHandler(this.btnSave_Click_1);
```

Na prvním řádku je definované umístění komponenty. Následuje nastavení jména, velikosti, popisku tlačítka a viditelnosti. Na závěr je definována událost reagující na stisknutí tlačítka. Takto obdobně jsou definovány všechny komponenty včetně Formuláře *Form*.

## 6.4 Form1.cs

Form1.cs je základním unitem celé aplikace, z něhož jsou volány funkce a procedury napsané v jednotce Helper.

Prvním krokem, který je vždy nutné provést je načtení knihoven, které bude aplikace využívat.

```
#region Using directives
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Imaging;
using System.IO;
using System.Text;
using System.Windows.Forms;
using System.Runtime.InteropServices;
using System.Drawing.Printing;
#endregion
```

Před samotným psaním událostí reagujících na jednotlivé úkony uživatele je ještě nutné deklarovat globální proměnné. Následuje seznam globálních proměnných obsažených v aplikaci Peacock a stručný popis jejich využití.

```
double[,] tab
```

Dvojměrné pole typu double slouží jako původní paleta, kterou je naplněna komponenta DataGrid v záložce transformační tabulka, kterou může uživatel editovat.

```
double[,] user_tab
```

Dvojměrné pole typu double slouží k uložení uživatelské barevné palety.

```
int korekce=0
```

```
int korekce1 = 0
```

Dvojice celočíselných proměnných int použitých při zachycení snímku videosekvence

```
int x1=0, y1=0, n=0, x2=0, y2=0;
```

Globální proměnné pro označení požadovaného výřezu snímku, x-ové a y-ové pozice rohů

```
private System.Drawing.Printing.PrintDocument docPrint =new
System.Drawing.Printing.PrintDocument();
```

Deklarace tiskového objektu. Metoda bude nastavovat vlastnosti objektu Tiskového dokumentu při zobrazení dialogu.

```
private byte[,] Input_BitmapData = null;
```

Dvojměrné pole typu byte sloužící pro uložení otvíraného obrázku

```
private byte[, ,] Pseudo_BitmapData = null;
```

Trojrozměrné pole typu byte určené k uchování dat obarveného snímku

```
private byte[, ] Mono_modif_BitmapData = null;
```

Dvojměrné pole typu byte určené k uchování upraveného šedotonového snímku

```
private byte[,] n_BitmapData = new byte[1000, 1000];
```

Dvojměrné pole typu byte určené k uchování upraveného šedotonového snímku

```
private byte[,] To_pseudo_BitmapData = null;
```

Dvojměrné pole typu byte, které slouží jako vstup funkci pro transformace pseudobarvení.

```
private byte[,] Snap_BitmapData = null;
```

Dvojměrné pole typu byte určené k uchování snímku zachyceného z videosekvence

```
private Bitmap Input_Bitmap = null
```

Proměná typu Bitmapa jenž bude obsahovat načtený obrázek.

```
private Bitmap Video_Mono_Bitmap = null;
```

Proměná typu Bitmapa jenž bude obsahovat snímek zachycený s videosekvence a bude zobrazena v záložce video.

```
private Bitmap Video_Pseudo_Bitmap = null;
```

Proměná typu Bitmapa jenž bude obsahovat obarvený obrázek snapnutý z videosekvence a bude zobrazena a může být uložena.

```
private Bitmap Mono_modif_BitmapModif = null;
```

Proměná typu Bitmapa monochromatického snímku po transformaci jasové stupnice.

```
private Bitmap Usertab_BitmapModif = null;
```

Proměná typu Bitmapa jenž vytvořená z pole odpovídající uživatelem vytvořené paletě a bude sloužit pro její uložení na disk.

Vždy při spuštění aplikace dále proběhne vytvoření objektu nápovědy typu HelpProvider, počáteční nastavení komponent a vyplnění komponenty DaraGrid (tabulky), polem tab.

V další části kapitoly budou vyjmenovány a stručně probrány události aplikace Peacock.

```
private void btnLoad_Click(object sender, EventArgs e)
```

Událost je vyvolána stiskem tlačítka LOAD a zajišťuje načtení obrázku. V události je pouze volána funkce fOpen, která realizuje samotné načtení

```
private void btHistogram_Click(object sender, EventArgs e)
```

Událost reagující na stisk tlačítka Histogram. V rámci události proběhne volání funkcí fObnova a fShowHistograms, které zajistí vykreslení histogramu.

```
private void btHelp_Click(object sender, EventArgs e)
```

Událost je vyvolána stiskem tlačítka nápověda a pomocí příkazu ShowHelp spustí nápovědu.

```
private void Transform_video1_Click(object sender, EventArgs e)
```

Událost vyvolaná stiskem tlačítka Transformace videa. V události je pouze volána funkce fTransform, která realizuje transformaci přesněji řečeno volá externí funkce v unitu Helper, které transformaci počítají.

```
private void btnSave_Click_1(object sender, EventArgs e)
```

Událost tlačítka Save volá funkci fSave, která provede uložení výstupního snímku v nepravých barvách

```
private void Ekvalizace_Click(object sender, EventArgs e)
```

Událost reagující na stisk tlačítka Transformace v poli Výběr jasové transformace. Základem této události je funkce switch-case sloužící k rozvětvení běhu programu a k vybrání příslušné větve, dle zvolené šedotónové transformace. Ve větvích jsou volány funkce nacházející se v unitu Helper realizující transformace jasové stupnice, transformační funkce vracejí pole typu byte.

Po transformaci je volána funkce SaveArrayToBitmap1, který převede pole zpět na bitmapu a na je zobrazena. Zároveň jsou volány také funkce fObnova a ShowHistograms s parametrem 0, určující, že bude počítán histogram upraveného snímku nikoliv původního (viz popis funkce fShowHistogram).

```
private void btnSave_Click_1(object sender, EventArgs e)
```

Událost je vyvolána stiskem tlačítka Ulož na Hlavní záložce, pouze volá funkci fSave s parametrem 0. Funkce realizuje uložení bitmapy na pevný disk, přičemž parametr určuje jaká bitmapa má být uložena.

```
private void btSavepicture_usertab_Click(object sender, EventArgs e)
```

Událost je vyvolána stiskem tlačítka Ulož na záložce Transformační tabulka, pouze volá funkci fSave s parametrem 1. Funkce realizuje uložení bitmapy na pevný disk, přičemž parametr určuje jaká bitmapa má být uložena (viz. Popis funkce fSave).

```
private void btSavepicture_video_Click(object sender, EventArgs e)
```

Událost je vyvolána stiskem tlačítka Ulož na záložce Video, pouze volá funkci fSave s parametrem 2. Funkce realizuje uložení bitmapy na pevný disk, přičemž parametr určuje jaká bitmapa má být uložena (viz. Popis funkce fSave).

```
private void btTisk_Click(object sender, EventArgs e)
```

Obsluha tlačítka Tisk umístěného na hlavní záložce. V rámci funkce je vytvořen a nastaven tiskový dokument a je volána funkce private void printDoc\_PrintPage (viz níže). Pracuje se s komponentnou PrintDialog.

```
private void btOpenVideo_Click(object sender, EventArgs e)
```

Obsluha tlačítka pro otevření a spuštění videa. V programu je přehrávání videosekvence provedeno s využitím příkazů mciSendString s parametrem play MediaFile.

```
private void printDoc_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)
```

Událost nastavující tiskový dokument.

```
private void btPauze(object sender, EventArgs e)
```

Událost vyvolaná stiskem tlačítka Pauze. Zastaví přehrávané video opět použitím metody mciSendString, zasílající příkaz daný řetězcem, který je parametrem metody. V tomto případě je řetězec pauze MediaFile.

```
private void btResumeVideo_Click(object sender, EventArgs e)
```

Událost reagující na stisk tlačítka pokračovat na záložce Video. Použitím příkazu mciSendString("play MediaFile", null, 0, IntPtr.Zero) opětovně spouští videosekvenci.

```
private void btCapture1_Click(object sender, EventArgs e)
```

Událost reagující na stisk tlačítka Capture na záložce Video a volá funkci fPrinscreen s parametrem 0. Tato funkce zajistí snapnutí aktuální snímku videosekvence (popis níže).

```
private void btTransformace_UserTab_Click(object sender, EventArgs e)
```

Událost vyvolaná stiskem tlačítka Transformace na záložce Transformační tabulka. Pomocí cyklu for uloží obsah uživatelem upravené tabulky do pole user\_tab a vzápětí dané pole použije pro transformaci snímku do nepravých barev prostřednictvím volání funkce fTabUser umístěné v unitu Helper (viz. Popis funkce fTabUser). Transformační funkce vrací pole, takže je v zápatí volána funkce SaveArrayToBitmap, prostřednictvím, které je z pole vytvořena bitmapa a tak je pak zobrazena.

```
private void btTransformace_pseudo_static_Click(object sender, EventArgs e)
```

Událost reagující na stisk tlačítka Transformace na hlavní záložce. V události je pouze volána funkce fTransformPicture(), která realizuje transformaci přesněji řečeno volá externí funkce v unitu Helper, které transformaci počítají.

```
private void btCapture2_Click(object sender, EventArgs e)
```

Událost vyvolaná stiskem tlačítka ukládnání snímků na záložce Video. Zajišťuje pravidelné sejmutí obrazovky video sekvence. K čemuž je využita komponenta Timer (časovač), která je v této události nastavena.

```
timer1.Tick += obsluhaCasovace;  
timer1.Enabled = true;  
timer1.Interval = 5;
```

V prvním řádku je nastaveno, že časovač bude obsluhován událostí ObsluhaCasovace, což znamená, že tato událost bude vždy po uplynutí daného intervalu volána. Dále je časovač spuštěn a časový interval neboli perioda vyvolání obslužné události je nastavena na 5ms.

```
private void btStopSnap_Click(object sender, EventArgs e)
```

Událost reagující na stisk tlačítka Zastav ukládání. Slouží k zastavení snímání obrazovky, což je provedeno zastavením časovače (Timer1.Stop).

```
private void btTransformace_pseudo_sekvence_Click(object sender, EventArgs e)
```

Událost reaguje na stisk tlačítka Transformace videosekvence. Slouží k pseudobarvení videosekvence. Její kód je prakticky shodný jako u události pro sejmutí snímků ( událost btCapture2\_Click), jen s tím rozdílem, že je nastavena jiná obsluhující funkce tentokrát je to ObsluhaCasovace1.

```
private void cmbtOtevři_Click(object sender, EventArgs e)
```

Událost reagují na stisk tlačítka Otevři v kontextové nabídce. Obsahuje pouze volání funkce fOpen(), který zajistí otevření a zobrazení vstupního snímku.

```
private void cmbtPseudobarvení_Click(object sender, EventArgs e)
```

Událost reagují na stisk tlačítka Transformace v kontextové nabídce. Obsahuje pouze volání funkce fTransform(), který zajišťuje pseudobarvení snímku.

```
private void Form1_KeyDown(object sender, KeyEventArgs e)
```

Událost stisku klávesy. Nastavuje reakci na stisk určitých kláves při běhu aplikace.

```
private void btLoad_colorspectrum_Click(object sender, EventArgs e)
```

Událost tlačítka Otevři paletu na záložce Transformační tabulka, sloužící k otevření uživateln dříve ukožené palety. Otvírání probíhá pomocí openFileDialogu, objektu typu stream a objektu typu BinaryReader. Nejprve musí být oba dané objekty vytvořeny. Stream je vytvořen v módu open. Poté je nastavena počáteční pozice čtení readru na počátek. Zbytek algoritmu se již odehrává v cyklu for, který je opakován 768. Což plyne z toho, že hodnoty transformační tabulky (palety) jsou uloženy ve formě jednořádkového vektoru obsahujícího právě 768 znaků (viz. Událost btSave\_colorspectrum). Pomocí funkce reader.ReadByte() je přečtena první hodnota a je převedena na proměnou typu string, aby mohla být zobrazena v komponentě DataGridView (původně int). Poté je provedeno testování, zda se jedná o krok cyklu dělitelný 3 bez zbytku, pokud ano je číslo červenou složkou. Když podmínka neplatí, je testováno zda je zbytek po dělení třemi jedna pokud ano jedná se o zelenou složku. Pokud neplatí ani jedna z podmínek je přečtená hodnota modrou složkou. Toto testování vyplývá ze způsobu ukládání palety, kdy je tabulka procházena a postupně jsou složky v daném řádku ukládány za sebe ve formátu RGBRGBRGB atd. Podle toho o jakou složku jsme zjistili, že se jedná je zobrazena v příslušném sloupci tabulky (první sloupec R, druhá sloupec G, třetí sloupec B). Řádek je dán pořadovým číslem cyklu děleným třemi. Na konci algoritmu musí být příkazem close uzavřen objekt typu stream.

```
private void btSave_colorspectrum_Click(object sender, EventArgs e)
```

Událost reaguje na stisk tlačítka Ulož paletu na záložce Transformační tabulka a realizuje uložení uživatelské barevné palety. Ukládání probíhá s využitím saveFileDialogu, objektu typu stream a objektu typu BinaryWriter. Nejprve musí být oba dané objekty vytvořeny. Stream je vytvořen v módu create. K uložení slouží proměnná pvalue typu byte o velikosti 768. Velikost je dána ukládáním tabulky předstávající palety ve formátu jednořádkového vektoru, ve kterém se opakují složky RGBRGBRGB atd. Tabulka má 256 řádků, tedy třikrát 256 se rovná právě 768. Algoritmus probíhá v cyklu for, který má 256 kroků. V každém kroku je hodnota ze tří sloupců komponenty DataGridView (ve které je uživatelská paleta zobrazena) přečtena a uložena na příslušnou pozici v proměnné pvalue. Po ukončení cyklu je celá proměnná pvalue pomocí příkazu stream.Write(pvalue, 0, pvalue.Length) uložena. Na konci algoritmu musí být příkazem close uzavřen objekt typu stream, jinak by nebylo uložení trvalé.

```
private void cbTransform_video_SelectedIndexChanged(object sender, EventArgs e)
```

Událost reagující na změnu aktuálně vybrané transformace v komponentě combobox cbTransform\_video. Algoritmus je postaven na použití podmínek if, které testují, která z metod byla vybrána, což se vyhodnocuje podle vlastnosti komponenty SelectedIndex. A podle zvolené metody jsou povoleny či zakázány textBoxy pro zadání parametrů a mění se text ve status baru.

Důvod k použití této funkce je to, že některé z metod vyžadují vstup uživatele v podobě zadání parametru transformace zatímco jiné nikoliv.

```
private void cbColor_SelectedIndexChanged(object sender, EventArgs e)
```

Událost reagující na aktuálně vybranou transformaci v komponentně combobox cbColor. Různým transformací se nastavují různé parametry, proto na jejich změnu musí aplikace reagovat. Stejná funkce jako v případě předešlé události.

```
private void picBox_MouseClick(object sender, MouseEventArgs e)
```

Událost reagující na stisk tlačítka myši na komponentu PictureBox.

V další části této kapitoly budou popsány funkce unitu Form1, jsou to funkce, které byli zmíněny při stručné charakteristice události aplikace. Tyto funkce zpravidla dále volají funkce obsažené v Unitu Helper, který obsahuje vlastní algoritmy důležité pro činnost programu.

```
void fOpen()
```

Funkce fOpen slouží k otevření vstupního snímku, jeho zobrazení v pictureboxu a převedení bitmapy na pole. Převedení na pole je nutné, aby bylo možné provádět matematické operace, které jsou k úpravě snímků.

K otevření je potřeba použít komponentu OpenFileDialog. Před samotným otevřením dojde k nastavení open dialogu. Nejdůležitější funkcí v této fázi programu je GetBitmapArray() převádějící bitmapu na pole, která se nachází v unitu Helper a bude popsána později.

```
void fTransformPicture()
```

Jádrum této funkce je struktura switch-case větvičí běh aplikace v závislosti na zvolené transformaci (volba rozbalovacím seznamem). Před větvením je testováno pomocí podmínky if, jestli se transformuje původní snímek nebo snímek s upravenou jasovou stupnicí. V jednotlivých větvích jsou volány funkce, které provedou samotnou transformaci. Tyto funkce jsou umístěny v samostatném unitu Helper. Po provedení transformace je výsledek uložen do pole a k zobrazení v PictureBoxu je třeba ji převést zpět na bitmapu, k čemuž slouží funkce SaveArrayToBitmap. Nakonec je upravený snímek zobrazen v PictureBoxu.

```
void fTransformPicture()
```

V podstatě shodná funkce s fTransformPicture pracující se snímky získanými s videosekvencí, pouze chybí rozhodování jestli se pracuje s originálním nebo upraveným snímkem.

```
void fSave()
```

Funkce realizující uložení bitmapy snímku v nepravých barvách. Uložení je provedeno pomocí SaveFileDialogu. Nejprve je provedena pomocí podmínky if podmínka, že bitmapa výstupního snímku není nulová, tedy že je co ukládat. V praxi by došlo k chybové hlášce v okamžik, kdy by se uživatel pokusil uložit snímek aniž by nějaký snímek upravil.

```
private void fShowHistograms(int a)
```

Funkce zajišťuje výpočet i vykreslení histogramu. Algoritmus výpočtu jen velmi jednoduchý. Nejprve je vytvořeno pole typu int jpole o rozměru 1\*256, jenž reprezentuje histogram. A pomocná proměnná jas typu int. Pole reprezentující snímek je procházeno pomocí cyklu for hodnoty jasu jsou ukládány do proměnné jas a poté je inkrementována hodnota v sloupci pole histogram, příslušící danému jasu.

```
for (int y = 0; y < height; y++)
{
    for (int x = 0; x < width; x++)
    {
        jas = m_BitmapData[x, y];
        jpole[jas] = jpole[jas] + 1;
    }
}
```

```
private void fObnova()
```

Funkce slouží k vymazání starého histogramu, před vykreslením nového, aby se nepřekrývali. Vymazání je realizováno nakreslením obdelníku s výplní stejnou jako má form, tedy Gainsboro přes celý PictureBox, do kterého je histogram vykreslován.

```
private void obsluhaCasovace1(object sender, EventArgs e)
```

Událost je opět volána pravidelně v časových intervalech a to při stisku tlačítka Obarvení videosekvence. Funkce načítá z pevného disku jednotlivé snímky videosekvence (uloženy během ObsluhyČasovače1.), převádí je na pole (funkce [Helper.GetBitmapArray](#)) a provádí pseudobarvení pomocí volání funkcí unitu [Helper](#). Následně převede pole opět zpět na bitmapu (funkce [SaveArrayToBitmap](#)) a zobrazí ji v PictureBoxu. Díky volání funkce s krátkým intervalem vznikne efekt obarvené videosekvence. Načítání po sobě následujících snímků je zajištěno inkrementací proměnné `korekce1` a její využití při nastavení cesty.

```
private void obsluhaCasovace(object sender, EventArgs e)
```

Tato událost je volána časovačem opakovaně po uplynutí daného intervalu. Ve funkci je inkrementována proměnná `korekce` a poté je volána funkce `printscreen`, které je daná proměnná předána. Význam proměnné bude vysvětlen ve funkci `printscreen`.

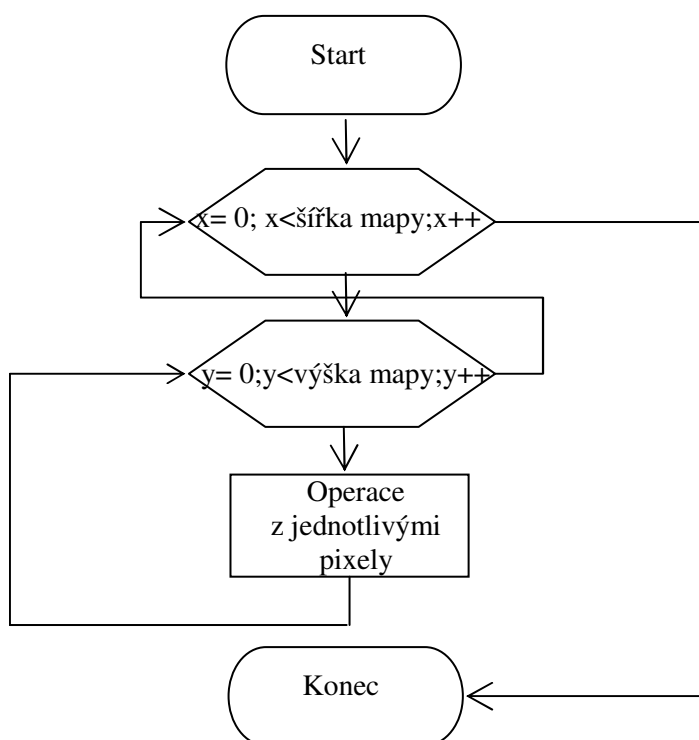
```
Bitmap printscreen(int x)
```

Tato funkce slouží k sejmutí aktuálního snímku videosekvence a k uložení dané bitmapy na pevný disk. Bitmapa může být později načtena v `obsluhaCasovace1` a snímky převeden do falešných barev. V první části funkce je vytvořen grafický objekt a je do něj uložen obraz pracovní plochy v rozsahu `picturebox`, ve kterém je přehráváno videosekvence. Tento grafický objekt je poté zobrazen v dalším `PictureBoxu` a uložen do bitmapy. Na konec je bitmapa uložena s využitím proměnné `korekce`, která díky inkrementaci zaručuje změnu názvu uloženého snímku. Což zabezpečuje, že se snímky navzájem nepřepisují. Proměnná typu `int` musí být převedena na typ `string`, aby mohla být použita při zadání cesty.

## 6.5 Helper.cs

Unit Helper obsahu know how celé aplikace. Obsahuje veškeré funkce zajišťující transformace jasové stupnice snímku i funkce realizující převod do pseudobarev. Důležitou součástí jsou také funkce sloužící pro převádění bitmapy na pole a naopak, což umožňuje úpravu snímků. V této kapitole budou popsány jednotlivé funkce a nastíněny použité algoritmy.

Veškeré prováděné transformace jsou bodové, což znamená, že na hodnotu obrazového bodu ve výstupní obrazce má vliv pouze hodnota odpovídajícího bodu ve vstupním obrazce. Funkce pro to musí procházet vstupní snímek bod po bodu a provádět operace s hodnotou daného bodu. Toto procházení je prováděno dvěma do sebe vnořenými cykly for. Vývojový diagram je zobrazen na Obr. 6.2. Z důvodu složitosti diagramu není již v následujících diagramech zakreslován, ale všechny operace automaticky probíhají vnořeny do tohoto cyklu.



Obr. 6.2 Vývojový diagram vnořených cyklů for

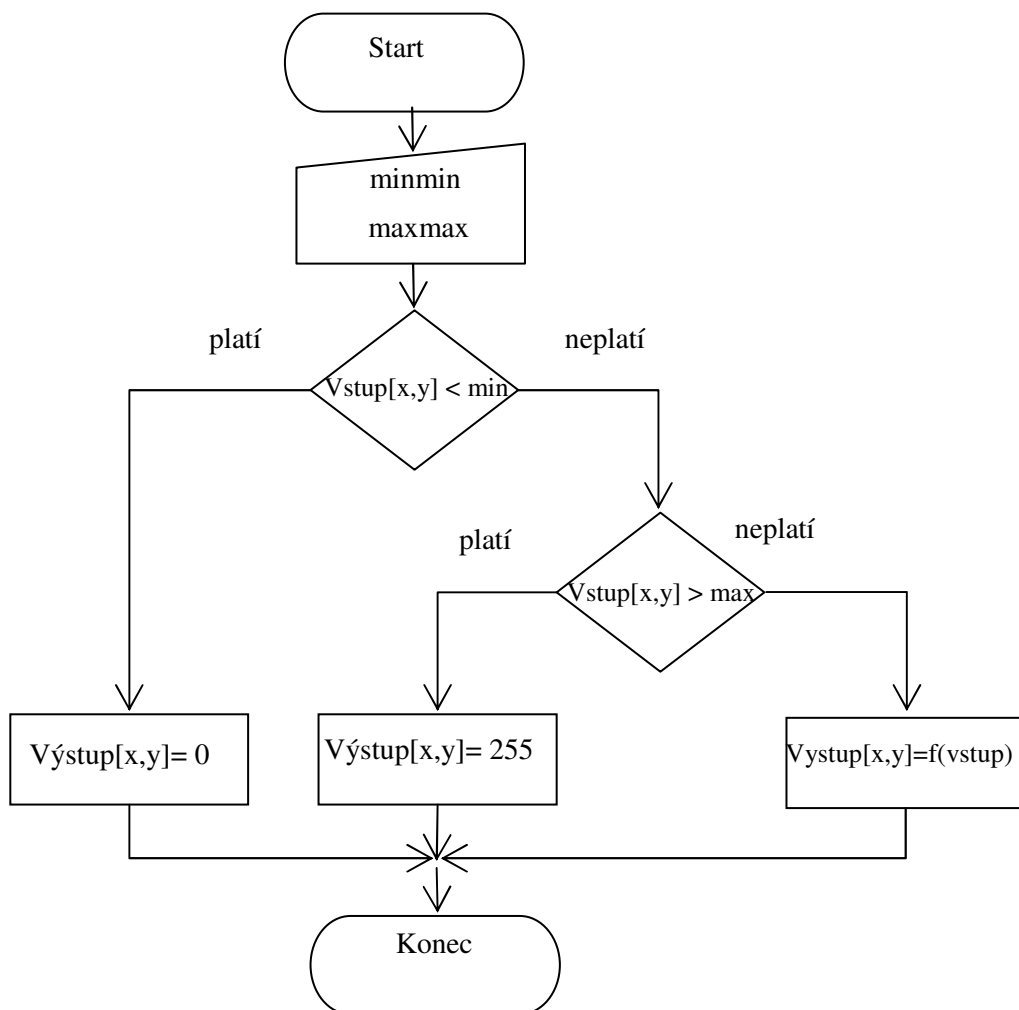
### 6.5.1 Funkce transformace jasové stupnice

```
public static byte[,] fscale(Form owner, byte[,] bitmapArray, int width, int height)
```

Vstupem funkce je pole typu byte vstupního snímku a jeho rozměry, které slouží pro definici cyklu for. Funkce realizuje šedotónovou úpravu lineární zvýraznění jasu (viz. Kapitola 5.3). Funkce má dva základní kroky. V obou krocích je za pomoci cyklu for procházen celý snímek bod po bodu. V prvním kroku je nalezena maximální a minimální hodnota jasu v původním snímku. A v druhém kroku je pomocí těchto hodnot (lokální proměnné maxmax a minmin) provedena samotná transformace dle vztahu 5.2.

```
public static byte[,] fZvyrazneni_casti(Form owner, byte[,]
bitmapArray, int widht, int height, double minmin, double maxmax)
```

Funkce realizuje zvýraznění části jasové stupnice (viz. Kapitola 5.2). Vstupem této funkce je kromě obvyklých (pole vstupního snímku a jeho rozměry), také proměnné minmin a maxmax. Hodnoty těchto proměnných nastavuje uživatel a určují rozmezí jasové stupnice, která bude zvýrazněna. Opět je procházen snímek pomocí cyklu for. Nejprve je kontrolováno jestli hodnota aktuálního bodu má hodnotu nižší než nastavené minimum, pokud ano je na dané pozici ve výstupním obraze nastavena nulová hodnota jasu. Jestliže podmínka neplatí je testována další. Tentokrát jestli není hodnota jasu větší než nastavené maximum, pokud podmínka platí je nastavena hodnota ve výstupním obraze na nejvyšší možný jas (255). Pokud ani druhá podmínka neplatí je provedena transformace dle vztahu 5.2. Vývojový diagram je znázorněn na Obr. 6.3.



Obr. 6.3 Vývojový diagram funkce fZvyrazneni\_casti

```
public static byte[,] fNegativ(Form owner, byte[,] bitmapArray, int
widht, int height)
```

Funkce realizuje vytvoření negativu snímku. Negativ představuje převrácené hodnoty jasu původního snímku. Opět je použit cyklus for po bodu a je bod po bodu provedena následující matematická operace.

```
Vystup[i, j] = (byte)(255 - bitmapArray[i, j]);
```

Tedy vstupní hodnota jasu je odečtena od možného maxima. To znamená, že tam kde byl ve vstupním obraze velký jas bude malý.

```
public static byte[,] fGama(Form owner, byte[,] bitmapArray, int
widht, int height)
```

Funkce realizuje funkci Gama (viz. Kapitola 5.5), tedy ztmavení či zesvětlení snímku v závislosti na hodnotě parametru gama. Jas vstupního snímku musí být před transformací vydělen 255, protože funkce předpokládá rozsah jasu 0-1, zatímco systém pracuje s jasnem v rozsahu 0-255.

```
public static byte[,] fVyrez(Form owner, byte[,] bitmapArray, int
widht, int height, int x1, int x2, int y1, int y2)
```

Tato metoda provádí výřez uživatelem vybrané části obrazu. Vstupní proměnné x1, y1 představují souřadnice levého horního rohu vybrané oblasti a x2, y2 pravého dolního rohu. Tyto hodnoty jsou porovnávány se souřadnicemi jednotlivých bodů obrazu, které jsou dané cyklem. Pokud zkoumaný bod nespadá do zvolené oblasti je jeho jas nastaven na maximální hodnotu (zobrazen jako bílý). Naopak pokud bod patří do oblasti je jeho jas ve výstupním obraze shodný s jasnem ve vstupním obraze.

## 6.5.2 Funkce zobrazení v nepravých barvách

```
public static byte[, ,] fTab(Form owner, byte[,] bitmapArray, int
widht, int height)
```

Funkce fTab realizuje metodu pseudobarvení Barevná paleta (viz. Kapitola 4.2). Vstupem funkce je kromě vstupního snímku a jeho rozměrů proměnná n, která určuje uživatelem zvolenou paletu. Základem této funkce je několik polí typu byte, které představují vyhledávací tabulky. Jejich řádky představují jas vstupního snímku a hodnoty v jednotlivých sloupcích základní barvené složky (0. Sloupec = červená složka, 1. sloupec = zelená složka, 2. Sloupec = modrá složka). Samotné obarvení je realizováno jednoduchým algoritmem. Pomocí cyklu for je procházen snímek bod po bodu a trojrozměrné pole, představující výstupní barevný snímek je plněno v závislosti na hodnotě jasu hodnotami v příslušném řádku. V algoritmu i a j, zastupují pozici ve snímku (tabulce) a třetí rozměr, určuje zda se jedná o modrou, zelenou či červenou složku.

```
for (int i = 0; i < widht; i++)
{
    for (int j = 0; j < height; j++)
    {
        Vystup[i, j, 0] = (byte) (tabulka[bitmapArray[i, j], 0]);
        Vystup[i, j, 1] = (byte) (tabulka[bitmapArray[i, j], 1]);
        Vystup[i, j, 2] = (byte) (tabulka[bitmapArray[i, j], 2]);
    }
}
```

```
public static byte[, ,] fTabUser(Form owner, byte[,] bitmapArray,
int widht, int height, double[,] tabulka)
```

Algoritmus funkce je stejný jako v předešlém případě (fTab). Změna spočívá v tom, že v této funkci nejsou vytvořeny přednastavené palety, ale vstupem je kromě snímku i tabulka, kterou si uživatel sám nastavil v záložce Transformační tabulka. A tato tabulka určuje transformaci.

```
public static byte[, ,] fPrah(Form owner, byte[,] bitmapArray, int
widht, int height)
```

Funkce aplikuje metodu hustotních řezů (viz. Kapitola 4.6). V každém kroku dvou vnořených cyklu for je hodnota jasu příslušného obrazového bodu, porovnávána s postupně roztoucími prahy. V okamžik, kdy je hodnota menší než prah, porovnávání končí a výstupnímu pixelu jsou přiřazeny hodnoty barevných složek (příslušných k danému prahu, pro každý prah jsou různé RGB složky). Testování prahu je realizováno podmínkou if. Prahy jsou zvoleny vhodně na základě odborných lékařských průzkumů tak, aby největší kontrast nastával v jasových oblastech, kde je to u lékařských snímků důležité.

```
public static byte[, ,] fCube2(Form owner, byte[,] bitmapArray, int
widht, int height, int n, int x1, int x2, int y1, int y2)
```

Algoritmus funkce je založen na metodě Schrodingerovi barvicí kostky popsané v kapitole 4.5 a převzaté ze zdroje [7]. Vstupem je kromě snímku a jeho velikosti parametr *n*, který určuje kolika jasům ve vstupním obraze bude přiřazena shodná barva ve výstupním snímku. Pokud jde tedy vlastně o určení počtu repetitivních barev duhy. Dalšími vstupy jsou proměnné *x1*, *x2*, *y1* a *y2*, které vymezují oblast snímku, která má být obarvena. Funkce může být volána ve dvou různých situacích. Uživatel může zvolit metodu Parametrické křivky a pak jsou tyto hodnoty automaticky nastaveny, takže jsou obarveny všechny body snímku. Pokud uživatel zvolí metodu Obarvení výřezu jsou hodnoty dány uživatelem.

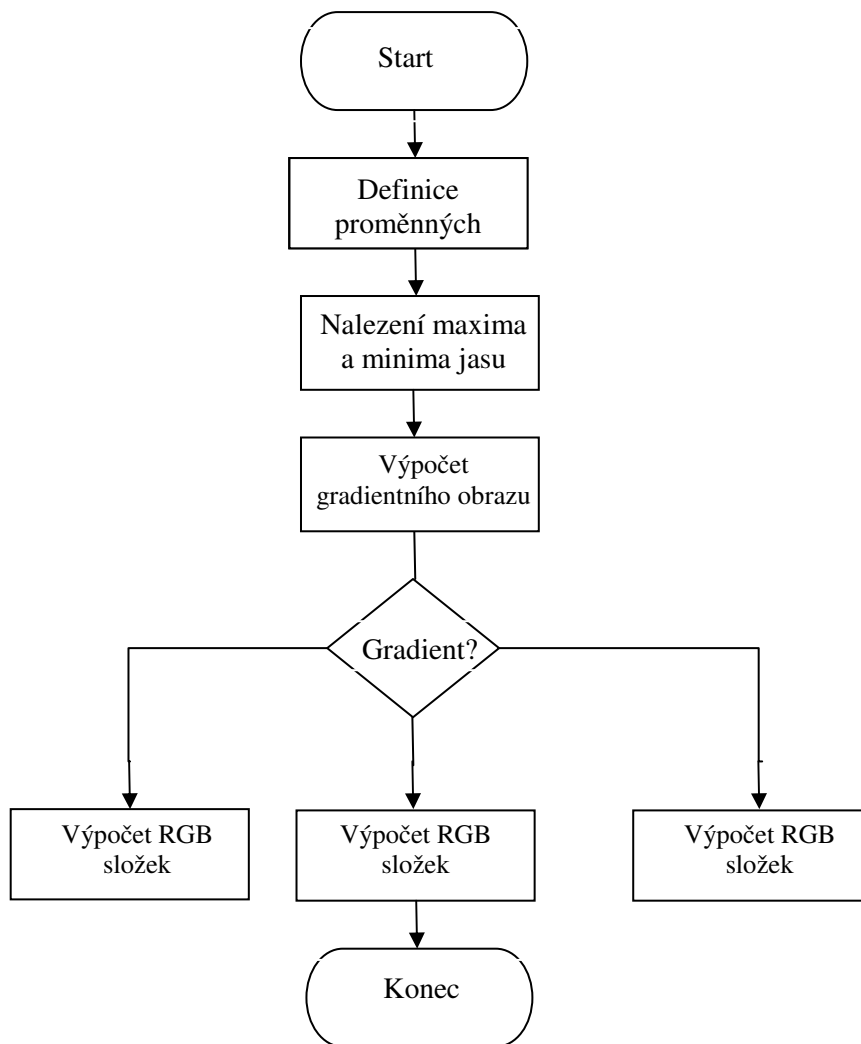
Na začátku je vytvořeno pole vstup typu `byte`, o stejné velikosti jako má pole vstupního snímku. Celý algoritmus probíhá ve dvou vnořených cyklech `for`. V první části je běh programu větven dle parametru *n* a v závislosti na tom je různě upravená hodnota jasu vstupního snímku. V další fázi jsou testovány proměnné *x1*, *x2*, *y1*, *y2* a přichází další větvení programu. Pokud platí  $((i > x1 \ \&\& \ i < x2) \ \&\& \ (j > y1 \ \&\& \ j < y2))$  znamená to, že aktuální obrazový bod patří oblasti, která má být obarvena a přichází na řadu série podmínek, testujících velikost jasu a podle jeho velikosti jsou přiřazeny pixelu barevné složky. Pokud podmínka neplatí, je všem třem složkám výstupního obrazového bodu přiřazen jas vstupního snímku, což znamená že zůstane šedivý.

```
public static byte[, ,] fKrivka(Form owner, byte[,] bitmapArray, int
widht, int height, double w, double fi)
```

Funkce realizuje převod na nepravé barvy metodou Parametrické křivky, která byla popsána v kapitole 4.1. Vstupem je pole šedotónového snímku, jeho rozměry a parametry transformace *w* a *fi*. Tyto parametry zadává uživatel. Algoritmus je založen na výpočtu dle vztahu 4.9. Před výpočtem je nutné normalizovat jas z rozsahu 0-255 na rozsah 0-1 a po výpočtu naopak zpět na rozsah 0-255 vhodný k zobrazení. Z tohoto důvodu jsou vytvořeny dvě pole typu `double`.

```
public static byte[, ,] fGraylevels(Form owner, byte[,] bitmapArray,
int widht, int height)
```

Funkce převádí monochromatický snímek na nepravé barvy metodou založenou na stupních šedi (viz. 4.3). K převodu jsou použity vztahy 4.11 až 4.15. Algoritmus je zachycen na Obr. 6.4 vývojovým diagramem. Kromě proměnných sloužících k uložení mezivýsledků a konečného výsledku jsou v první fázi definovány i dvě pole 3x3, které reprezentují masky vyhledávající hrany.



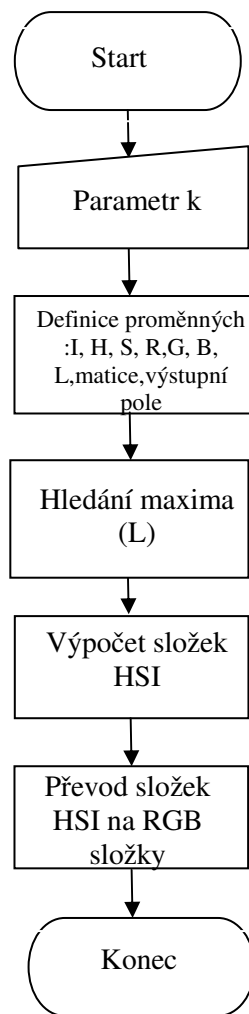
Obr. 6.4 Vývojový diagram funkce fGraylevels

```

public static byte[, ,] fHSI(Form owner, byte[, ] bitmapArray, int
widht, int height, double k)

```

Funkce realizuje převod do pseudobarev metodou založenou na barevných prostorech (viz. 4.4), která je založena na úpravě v barevném prostoru IHS a převodu zpět do RGB prostoru. Vstupem funkce je pole monochromatického snímku, jeho rozměry a parametr transformace  $k$ , který zadává uživatel. Algoritmus funkce přesně odpovídám vztahům 4.16 a 4.17. Kdy jsou nejprve spočítány parametry HSI modelu (4.16) a ty jsou poté převedeny na RGB složky (4.17). Vše opět probíhá bod po bodu pro každý bod snímku k čemuž jsou použité dva vnořené cykly for. Princip je zobrazen ve vývojovém diagramu na obrázku Obr. 6.5 (kvůli přehlednosti bez cyklu for)



Obr. 6.5 Vývojový diagram funkce fHSI

### 6.5.3 Funkce převodu bitmapy-pole

```
public static byte[,] GetBitmapArray(Form owner, string fileName, out
Bitmap bmp)
```

Metoda volá funkci `GetBitmapArraySafe`, která vytvoří 2 rozměrné pole obsahující data z otevřené bitmapy (snímku). Před voláním funkce je provedeno vytvoření objektu bitmapy s obrázkem načteného z pevného disku. K tomu je použit vstup `fileName`, který obsahuje plnou cestu a název souboru obrázku. Funkce vrací dvourozměrné pole typu `byte`, kdy souřadnice pole `[x,y]` jsou souřadnice obrazového bodu snímku a obsah je intenzita jasu.

```
public static byte[,] GetBitmapArray1(Form owner, Bitmap bitMapa)
```

Metoda volá funkci `GetBitmapArraySafe`, která vytvoří 2 rozměrné pole obsahující data z otevřené bitmapy (snímku). Funkce shodná s `GetBitmapArray` jen s tím rozdílem, že v ní není vytvořena bitmapy ze souboru, ale bitmapy je již přímo vstupem funkce. Výstup funkce je shodný.

```
private static byte[,] GetBitmapArraySafe(Bitmap bitMapa,
IncBarDelegate incMethod)
```

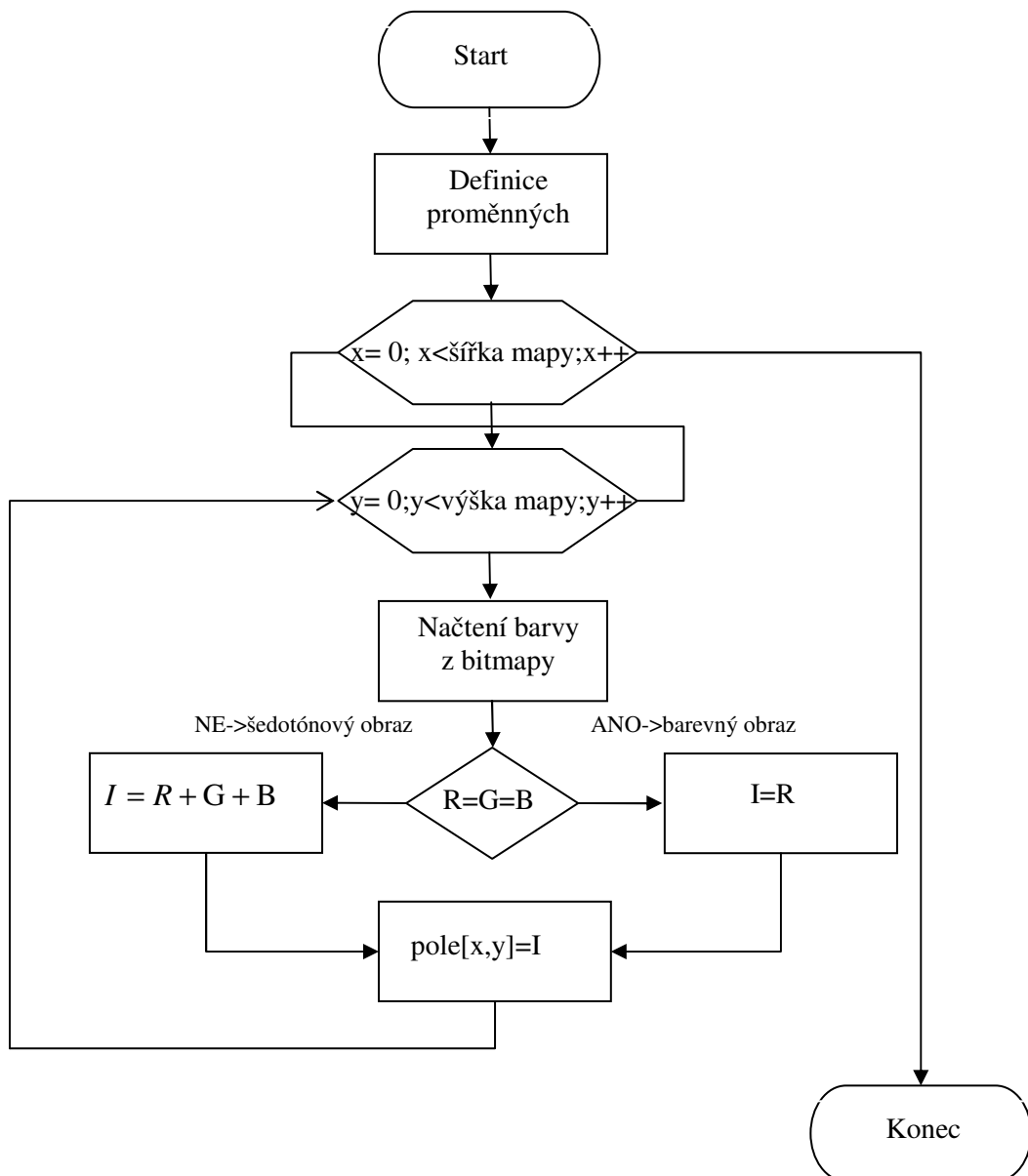
Funkce převede bitmapu na dvojrozměrné pole typu `byte`. Bitmapa je procházena bod po bodu a základním principem je použití funkce `GetPixel(x, y)`, který načte barvu obrazového bodu bitmapy. Princip je jasný s vývojového diagramu na Obr. 6.6.

```
public static Bitmap SaveArrayToBitmap(Form owner, byte[, ,] bitmapArray)
```

Metoda uloží obsah pole dat do bitmapy. Vstupní proměná bitmapArray je pole obsahující intenzitu jasu. Vstupním polem představuje snímek v nepravých barvách, proto je trojrozměrné. Přičemž dva rozměry jsou dány výškou a šířkou snímku. Třetí rozměr je 3 a určuje, že se jedná o tři matice a každá reprezentuje jednu barvenou složku RGB. Funkce prochází bod po bodu pole a na základě obsahu pole nastavuje hodnotu obrazového pole bitmapy pomocí funkce SetPixel. Výstupem je bitmapa, která je zobrazena a může být uložena jako klasický obrázek v jednom z používaných formátů.

```
public static Bitmap SaveArrayToBitmap1(Form owner, byte[, ] bitmapArray)
```

Metoda uloží obsah pole dat do bitmapy. Vstupní proměná bitmapArray je pole obsahující intenzitu jasu. Vstupním polem je tentokrát monochromatický snímek šedotónového obrazu po jasové transformaci, proto je pole pouze dvojrozměrné (šedotónový obraz je reprezentován jen jednou maticí v případě tří matic by se jednalo o tři stejné matice). Funkce je až na rozměr vstupního pole shodná s funkcí SaveArrayToBitmap.



Obr. 6.6 Vývojový diagram GetBitmapArraySafe

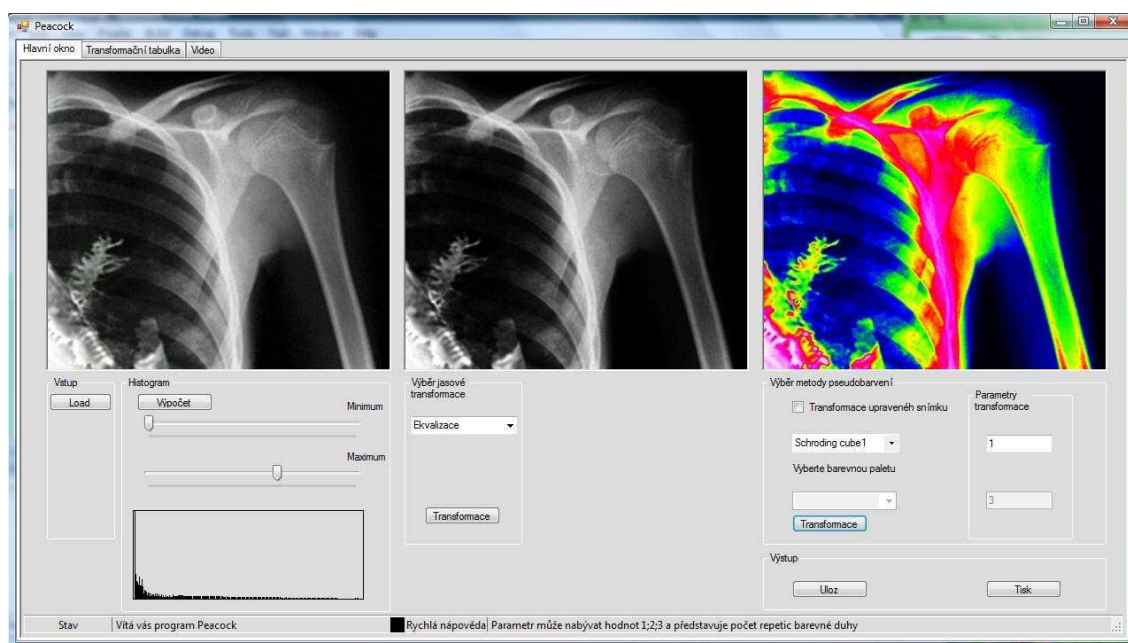
# 7 UŽIVATELSKÁ PŘÍRUČKA

Program Peacock je uživatelsky přívětivý a neklade na běžného uživatele nároky na hlubší teoretickou znalost problematiky. Na druhou stranu však poučenému uživateli poskytuje několik pokročilých funkcí, které mu mohou pomoci k efektivnímu používání a co největšímu zlepšení diagnostické výtěžnosti snímků.

## 7.1 Hlavní obrazovka

Po spuštění programu uživatel uvidí úvodní obrazovku programu. Na této obrazovce jsou uživateli dostupné téměř všechny funkce programu a v průběhu používání na ní názorně uvidí výsledky své práce

Mimo hlavní záložku jsou uživateli k dispozici další dvě pracovní plochy, které se mu zpřístupní po kliknutí na záložky s názvy Transformační tabulka a Video v levém horním rohu základní obrazovky (Obr 7. 1). O možnostech, které mu tyto nabídky nabízejí, bude pojednáno v dalších částí příručky.



Obr. 7.1 Hlavní obrazovka aplikace Peacock

Hlavní obrazovka je rozdělena na několik bloků. V horní části okna jsou umístěny tři boxy pro zobrazení snímků. V levém z nich je po načtení zobrazen vstupní monochromatický snímek. Prostřední slouží k zobrazení obrazu po aplikaci jasové transformace. V posledním boxu je umístěn výsledný snímek v nepravých barvách.

Pod blokem zobrazujícím snímky v různých fázích úpravy je umístěno několik bloků sloužících k ovládní programu. Ovládací prvky jsou dále rozděleny podle funkce. Úplně vpravo uživatel vidí blok Vstup, ve kterém je pouze jediné tlačítko Načti, které slouží k načtení vstupního snímku z pevného disku a k jeho zobrazení na příslušném místě. Kromě stisku tohoto tlačítka má uživatel další dvě možnosti jak otevřít obraz. Druhou možností je otevření s použitím rychlé nabídky častých operací přístupné stiskem pravého tlačítka myši. Poslední možností je využití klávesové zkratky CTRL-O. Výběr obrázku probíhá použitím standardního Windows dialogu. Aplikace může pracovat se snímky různých nejčastěji používaných formátů (jpeg, bmp, png, tiff, gif).

Dalším uceleným blokem zleva je Histogram(3), zde je umístěno tlačítko Vypočti po jehož

stisku je zobrazen histogram. Uživatel zde také najde dvě posuvné lišty, které slouží k nastavení minima a maxima jasu pro jasovou transformaci zvýraznění části jasové stupnice.

Následuje blok Volba jasové transformace. Jeho součástí je rozbalovací menu, které slouží k výběru transformace a tlačítko Transformace, po jehož stisku je transformace provedena a snímek zobrazen. Program nabízí tyto jasové transformace:

- ekvalizace,
- roztažení osy,
- zvýraznění části histogramu,
- negativ,
- logaritmizace,
- gama korekce,
- výřez.

Největším a pro funkci programu zásadním je skupina prvků nazvaná Nastavení metody transformace. Tento blok obsahuje rozbalovací menu, které je použito k volbě použité metody. Pod ním je další rozbalovací pole, které je přístupné pouze v případě, že uživatel zvolí metodu barevné palety a slouží k výběru jedné z připravených palet. Dále zde uživatel nalezne dva boxy sloužící k nastavení parametrů některých metod (Schrodingerova kostka, Schrodingerova křivka a IHS). Význam jednotlivých parametrů je vysvětlen po najetí na box formou plovoucí nápovědy a v status panelu v závislosti na vybrané metodě. Aplikace umožňuje volbu obarvovat původní snímek nebo ten po šedotónové úpravě. Tato volba je provedena zatrhnutím respektive nezatrhnutím zatrhávacího tlačítka Transformace upraveného obrazu. Po nastavení uživatel provede transformaci stiskem tlačítka transformace. Transformace je provedena a snímek je zobrazen. V programu jsou k dispozici tyto metody:

- Schrodingerova krychle,
- Křivka,
- Tabulka,
- Hustotní řezy,
- Metoda založená na IHS modelu,
- Metoda založená na šedotónových úrovních,
- Výřez.

Posledním blokem je Výstup. Tento blok slouží uživateli k závěrečnému naložení s upraveným snímkem. Tlačítko Tisk slouží k vytisknutí obrazu a tlačítko Ulož k jeho uložení na pevný disk. Obrazy lze ukládat stejně jako načítat do několika nejobvyklejších formátů. Uložení může být opět provedeno několika způsoby. A to mimo stiskem tlačítka Ulož, stiskem klávesové zkratky nebo přes kontextovou nabídku pravého tlačítka myši. Uložení opět probíhá pomocí standardního dialogu Windows a uživatel může sám zvolit název souboru, jeho umístění a formát snímku.

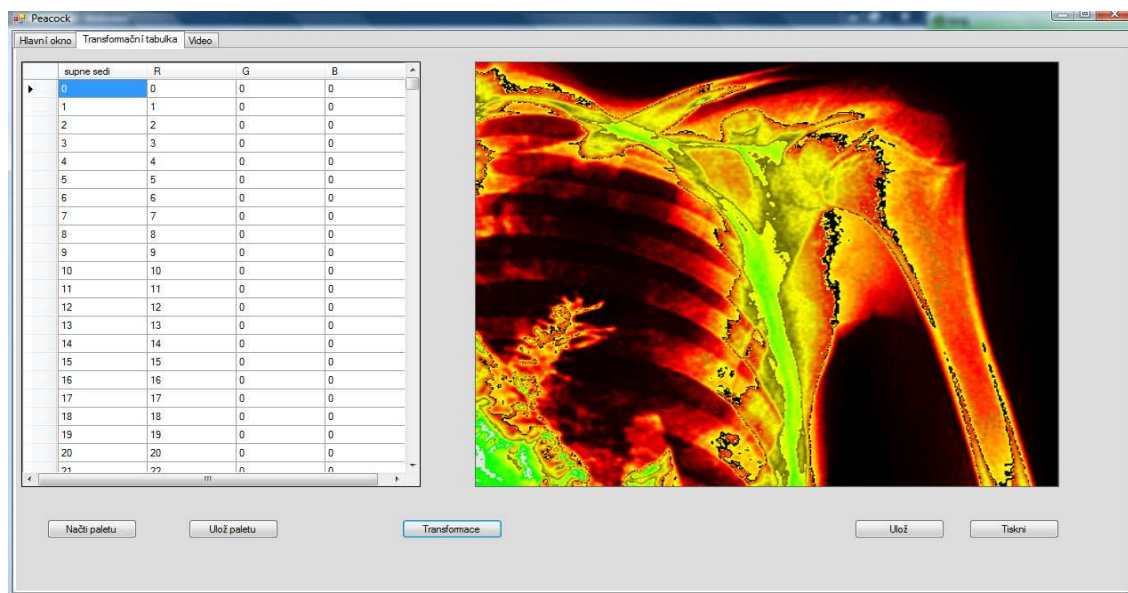
Na spodním okraji obrazovky je umístěn status panel, ve kterém uživatel nalezne informaci o stavu programu (například o označení hranic výřezu snímku) a v jeho druhé části rychlou nápovědu.

Kromě možností, které jsou uživateli patrné při pohledu na obrazovku, umožňuje program Peacock výběr části snímku, kterou chce obarvit, zatímco zbytek snímku zůstane chromatický. Výběr je proveden dvěma kliky na původní snímek. První kliknutí označuje levý horní roh výřezu a druhé kliknutí jeho pravý dolní roh. Pokud chce uživatel obarvit jen výřez, musí jej tímto způsobem zvolit a v rozbalovacím menu pro výběr metody vybrat možnost výřez.

## 7.2 Záložka tabulka

Po stisku záložky s názvem tabulka se uživateli zobrazí nová pracovní plocha (obr. 7. 2). I při použití této záložky je snímek načten tlačítkem Načti na hlavní záložce. Na ploše je k dispozici převodní tabulka, snímku obsahující 256 řádků, každý pro jeden stupeň šedé a panel, ve kterém bude zobrazován obarvený snímek. Ovládacím prvky je tlačítko pro provedení pseudobarvení, pro uložení či načtení palety a tlačítka pro uložení a vytisknutí snímku.

Tato část programu slouží k přímému ovlivnění LUT tabulky. Umožňuje tak uživateli nastavit barvu na jakou bude obarvena konkrétní jasová hodnota ve vstupním obraze. Což může být výhodné, pokud jej zajímá výskyt konkrétních jasů ve snímku. K nastavení slouží tabulka (1). Tabulka obsahuje 4 sloupce a 256 řádků. Jednotlivé řádky představují jednotlivé jasové úrovně. V prvním sloupci je zapsána číselná hodnota jasu ve vstupním obraze. Do druhého, třetího a čtvrtého sloupce bude obsluha zapisovat barevných složek RGB výstupního obrazu odpovídající příslušným vstupním jasům. Hodnoty nabývají hodnot 0-255. Přičemž při špatně zadané hodnotě bude uživatel upozorněn chybovou hláškou. Barvu, kterou si uživatel tímto způsobem namíchal, může pozorovat na přiloženém vzorníku jasů.



Obr. 7.2 Záložka transformační tabulka

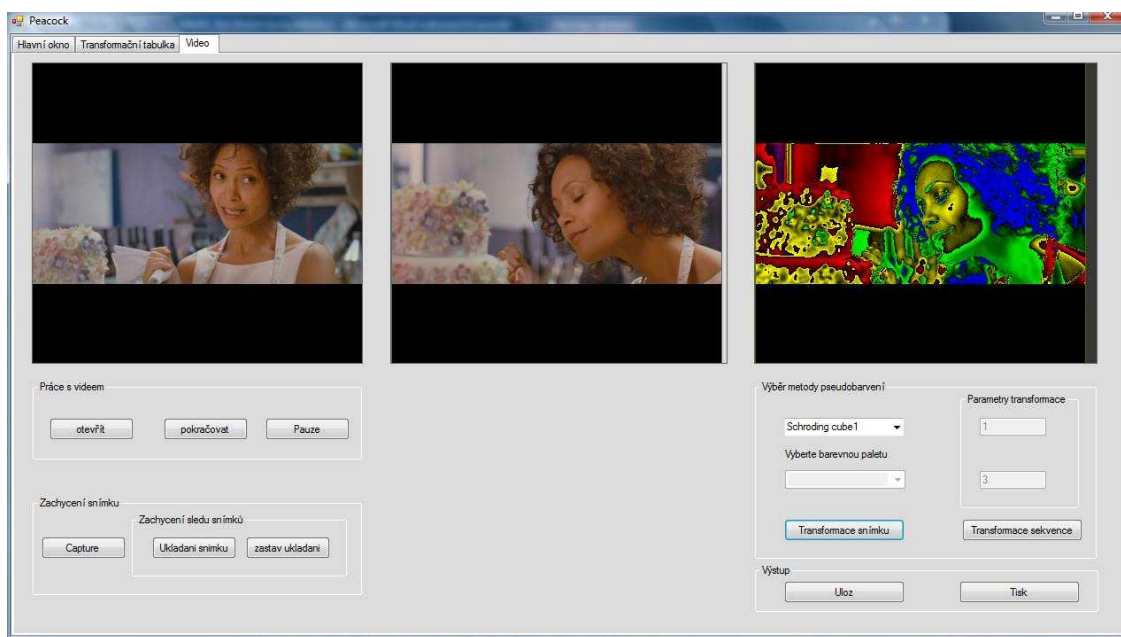
## 7.3 Záložka video

Poslední záložka Video nabízí, jak již napovídá její název, práci s video sekvencemi. Největší část okna aplikace zabírají 3 zobrazovací panely. V prvním z nich (1) je zobrazováno aktuálně přehrávané video. Prostřední okno slouží k dvěma různým úkolům v závislosti na zvoleném režimu práce s video sekvencí. Uživatel má dvě možnosti. Jeho první volbou může být zachycení jediného snímku videa ten je pak zobrazen v tomto panelu a v dalším kroku obarvena zobrazen v pravém panelu (3). Další možností je zachycení několika snímků současně, které jsou poté zobrazovány postupně v prostředním panelu a později jako obarvená sekvence v pravém panelu.

Na obrazovce jsou kromě zobrazovacích prvků, také prvky ovládací. K otevření, zastavení či přetočení videosekvence slouží tlačítka v bloku Práce s videm (2). Tlačítko Otevři slouží k načtení (otevření) videosekvence z pevného disku. Po výběru videa ve standardním dialogovém okně systému Windows dojde k automatickému spuštění přehrávání. Přehrávání sekvence lze zastavit tlačítkem Pause. K opětovnému spuštění zastaveného videa slouží tlačítko

pokračovat. Stejně jako většina ostatních akcí aplikace lze i v tomto případě použít místo stisku tlačítka klávesové zkratky. Kompletní souhrn klávesových zkratk je k dispozici v kapitole 7.4. Dalším jednotným celem tvoří tlačítka pro sejmутí snímku či sekvence snímků. Prvním tlačítkem v tomto bloku je Capture po jeho stisku dojde k sejmутí (printcreenu) aktuálního snímku přehrávané audio sekvence a její zobrazení v prostředním panelu. Tlačítko Ukládání snímků slouží k uložení více po sobě následujících snímků a na pevný disk. K zastavení této akce slouží tlačítko Zastavení ukládání.

Následují dva bloky, které jsou prakticky totožné s bloky v hlavním okně. Prvním z nich je Nastavení metody pseudobarvení. Tato skupina opět obsahuje rozbalovací menu pro výběr metody a v případě volby metody Barevná paleta se zpřístupní i výběr konkrétní palety. Dále zde najdeme boxy určené pro zadání číselných parametrů metod a tlačítko Transformace. Nově přibýlo tlačítko Transformace sekvence, po jehož stisku dojde k obarvení a postupnému zobrazení snímků, které byly uloženy na pevný disk stisknutím tlačítka Ukládání snímků (viz. Výše). Posledním blokem je blok výstup obsahující tlačítko Ulož a Tisk.



Obr. 7.3 Záložka Video

## 8 DOSAŽENÉ VÝSLEDKY

Závěrečná kapitola si klade za cíl seznámit čtenáře s dosaženými výsledky. Byl vybrán ukázkový rentgenový snímek a na něm jsou ukázány všechny možnosti programu. Současně jsou výsledky porovnány s výsledky obarvení snímků v jiných dostupných programech.

### 8.1 MIPS 2.0 - Microscopy Image Processing Software

MIPS 2.0 je výukový software vytvořený na pražské ČVUT, který umožňuje různé metody předzpracování obrazu a je možné nad obrazem provádět různé operace. Hlavní důraz je kladen na jednoduchost a přehlednost všech operací. K realizaci bylo použito prostředí C++ Builder 5.0. Tato aplikace tedy není specializovaná na zobrazování v nepravých barvách. Pseudobarvení je jen jednou ze součástí tohoto softwaru, proto nenabízí širokou škálu metod, ale pouze metodu odpovídající metodě barevné palety. V programu je stejně jako v programu Peacock k dispozici několik předdefinovaných palet:

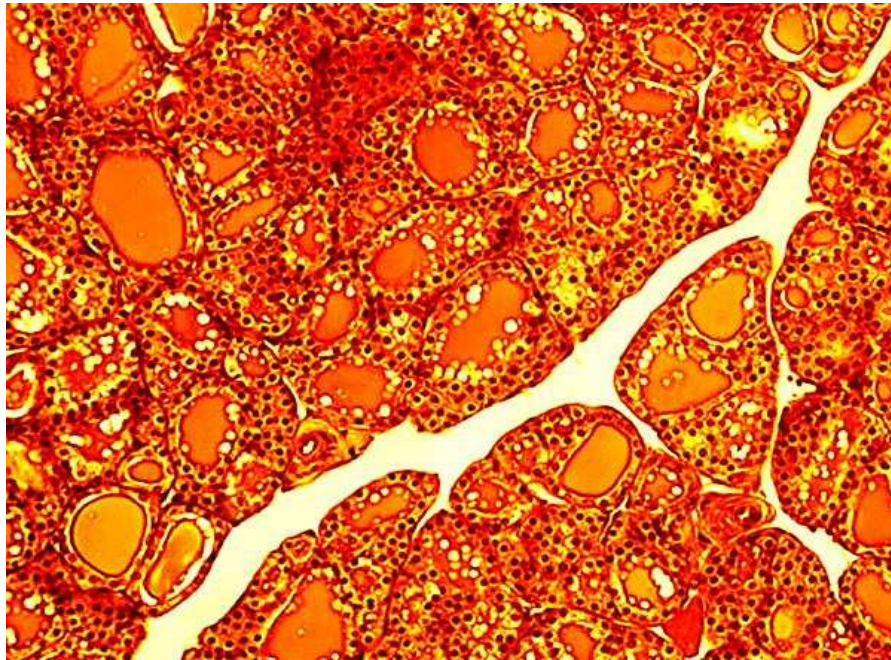
- bone,
- copper,
- hot,
- jet,
- HSV,
- flag.

Při výběru jedné z nich se v obrázku změní informace v LUT tabulce, kde je informace o tom jaká barva odpovídá dané jasové úrovni. Změní se tedy pouze informace o tom, jak má být daný obrázek zobrazen. Např. šedotónový obrázek je vždy možné po aplikaci barevné palety zobrazit opět jako šedotónový.

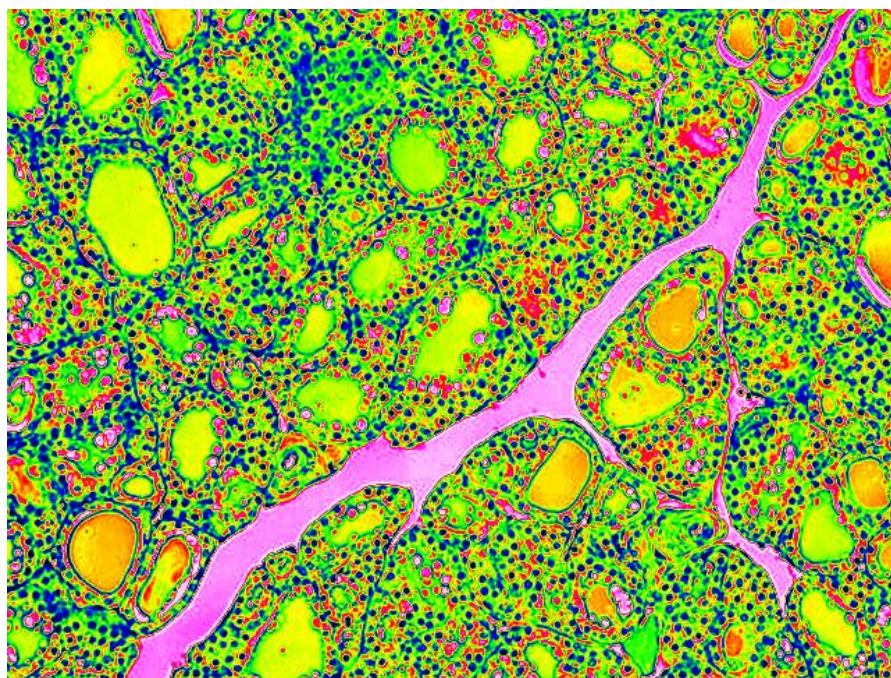
K dispozici je i možnost vytvořit si vlastní barevnou paletu. Je možné si zvolit jasový rozsah, který se zobrazí vybranou barvou, nebo použít gradientní přechod od jedné barvy k druhé, takže není možné přesné přiřazení barvy danému jasu, jako v programu Peacock. Tento způsob je na druhou stranu příjemnější pro laika.

Program nabízí široké spektrum funkcí počínaje FFT transformací, ale k účelu pseudobarvení poslouží lépe program Peacock. Mírnou nevýhodou aplikace může být to, že umí pracovat pouze se snímky ve formátu BMP.

Pro ukázkou byl vybrán snímek přiložen jako demonstrační k programu MIPS a byl obarven v obou programech. Na obrázku 8.1 je zobrazen snímek po aplikaci programu MIPS, konkrétně palety HOT. Zatímco na obrázku 8.2 vidíme snímek po aplikaci programu Peacock, konkrétně metody Pseudobarvící krychle (viz. 4.5). Na první pohled je patrný rozdíl v účinnosti a využitelnosti těchto snímků. V místech, která se na snímku upraveném programem MIPS zdají mít stejný jas (po obarvení prakticky bílá barva), vidíme po obarvení programem Peacock barevné rozdíly.



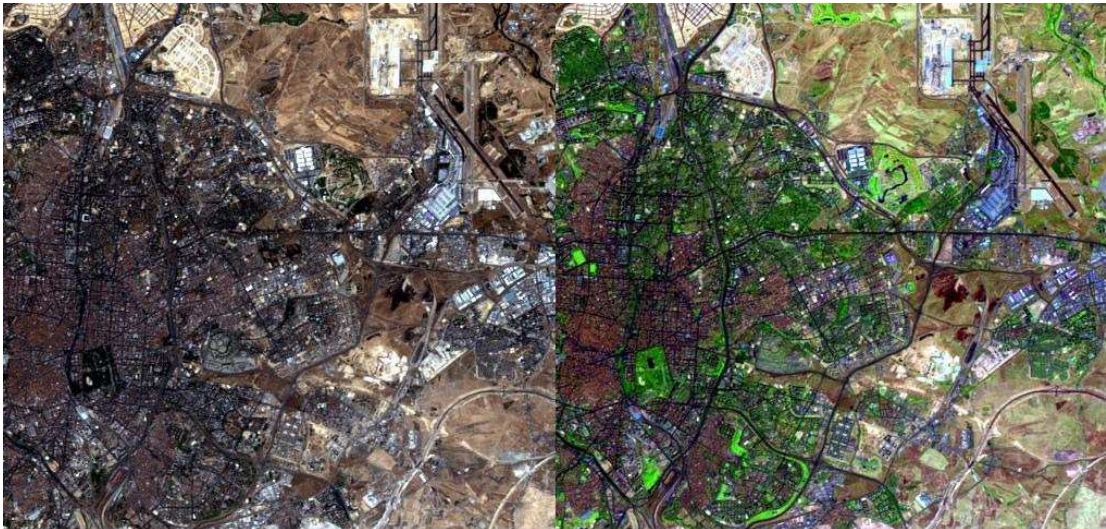
Obr. 8.1 Snímek převedený do falešných barev programem MIPS.



Obr. 8.2 Snímek převedený do falešných barev programem Peacock.

## 8.2 MultiSpec

Dalším programem nabízející možnost pseudobarvení je program Multispect. Tento program však neupravuje obrázky v používaných grafických formátech (JPEG, BMP). Program umí pracovat se snímky ve zvláštním formátu popsané třemi textovými soubory. Aplikace tedy slouží výlučně k obarvení snímku LANDSAT. Software pracuje na principu vhodné kombinace tří základních barevných kanálů. Program poskytuje velmi dobré výsledky, ale je úzce specializovaný a neumožňuje pseudobarvení lékařských snímků.



Obr. 8.3 Ukázka práce programu Multispec

### 8.3 Peacock

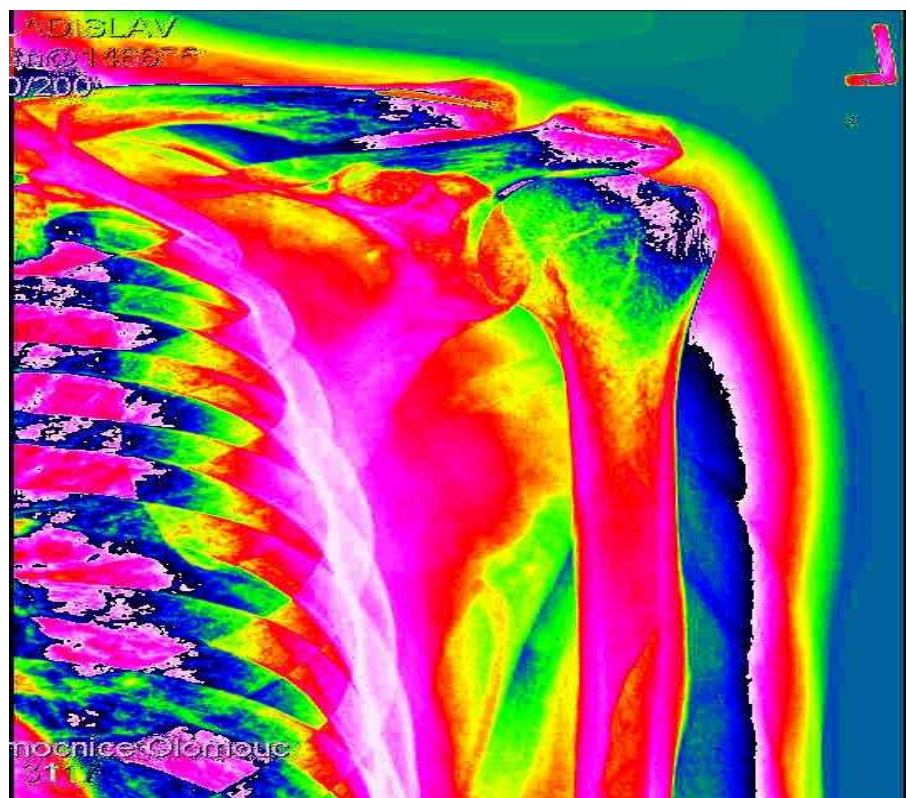
Program Peacock nabízí několik různých metod navíc v kombinaci s několika metodami transformace jasové stupnice. Vzhledem k možnosti u některých metod měnit nastavení parametrů ovlivňující výsledek pseudobarvení poskytuje program prakticky nekonečné množství výsledných vzhledů snímku. Na následujících snímcích je ukázáno alespoň několik ukázek. K demonstraci možností programu byl použit reálný rentgenový snímek ramene. Snímek pro tento účel poskytl pacient, který souhlasil se zpracováním snímku, ale nechtěl být v práci jmenován.



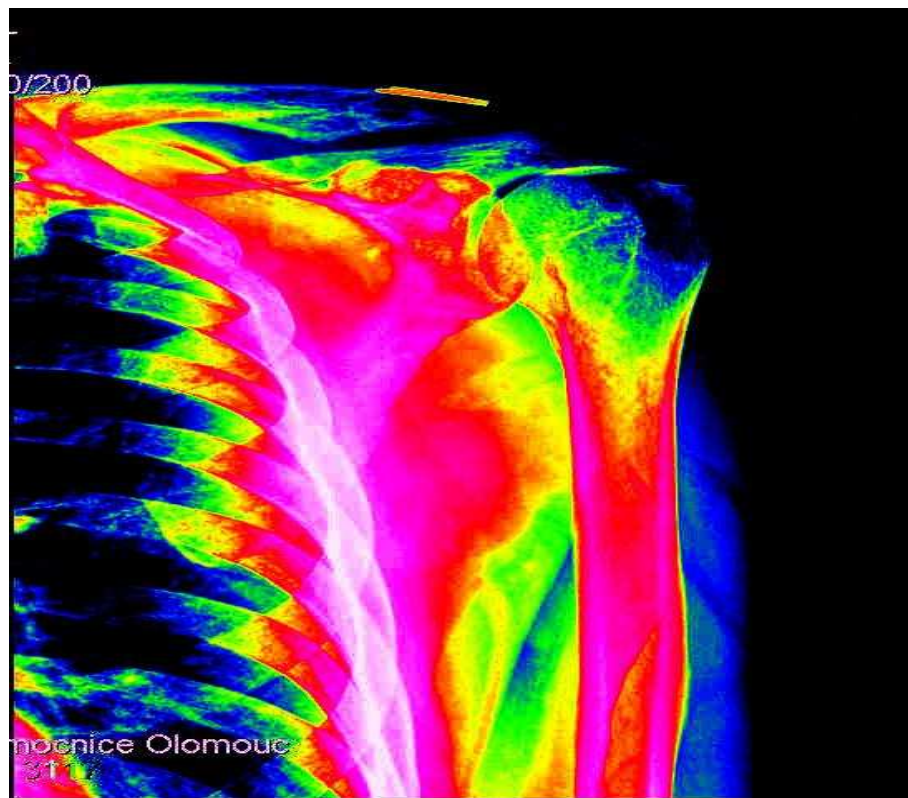
Obr. 8.4 Původní rentgenový snímek



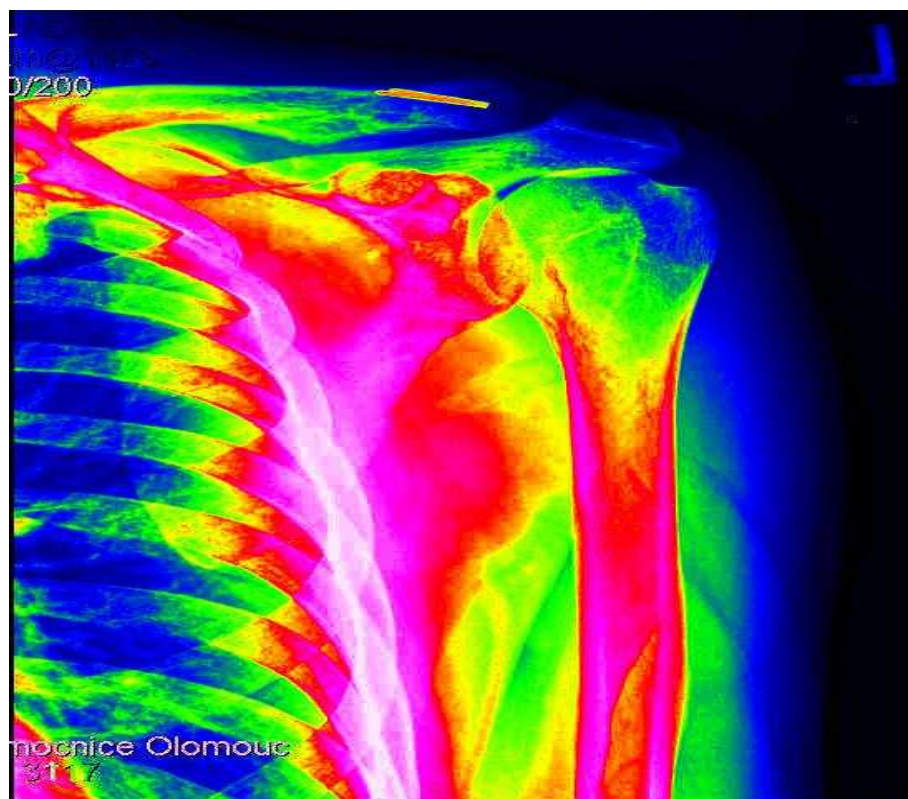
Obr. 8.5 Snímek obarvený metodou Barvící krychle s parametrem 1.



Obr. 8.6 Snímek obarvený metodou Barvící krychle s parametrem 2.



Obr. 8.7 Snímek upravený šedotónovu úpravou zvýraznění části stupnice a obarvený metodou Barvící krychle s parametrem 1



Obr. 8.8 Snímek upravený šedotónovu ekvalizace a obarvený metodou Barvící krychle



Obr. 8.9 Rentgenový snímek obarvený metodou křivky s parametry ( $\omega = 4$  a  $\phi = 6$ )



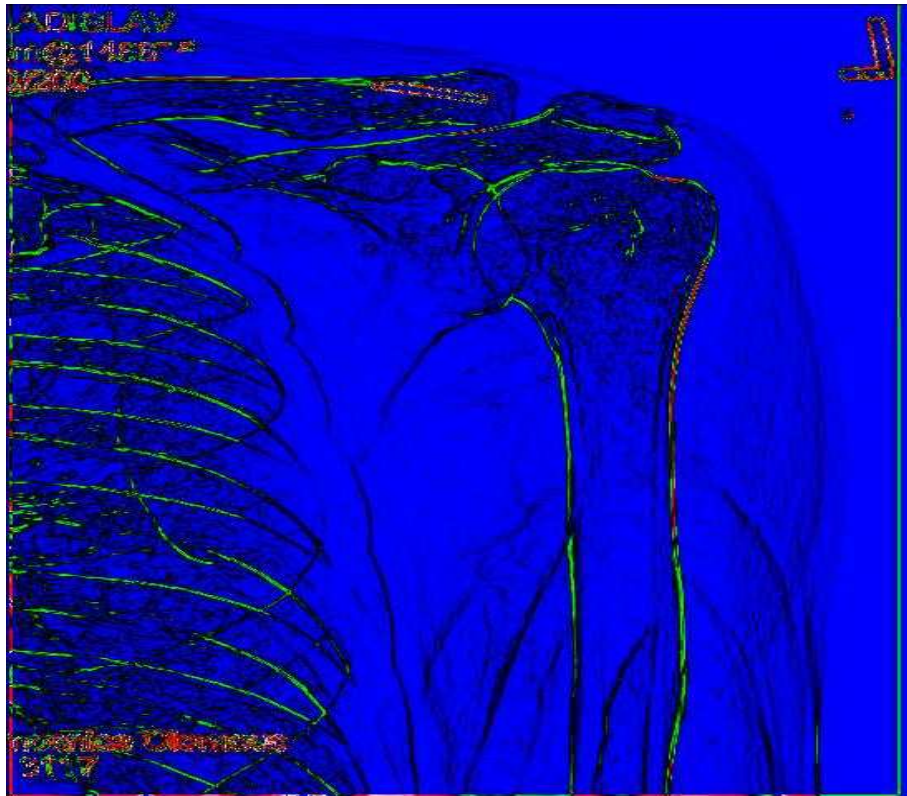
Obr. 8.10 Rentgenový snímek obarvený metodou křivky s parametry ( $\omega = 10$  a  $\phi = 8$ )



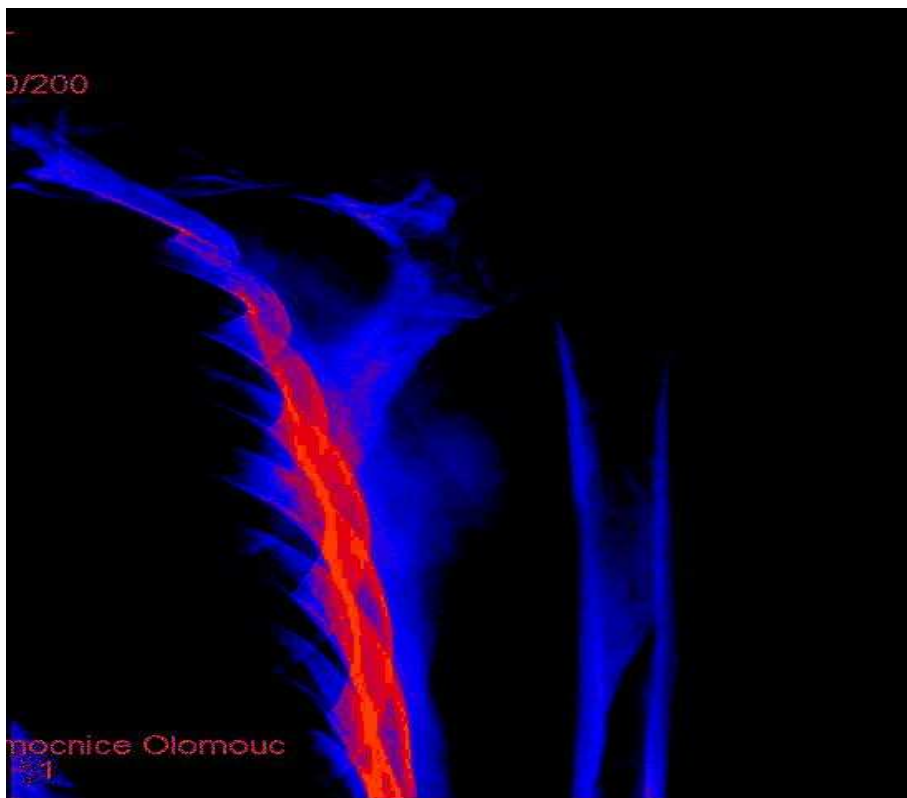
Obr. 8.11 Rentgenový snímek obarvený metodou křivky s parametry ( $\omega = 20$  a  $\varphi = 8$ )



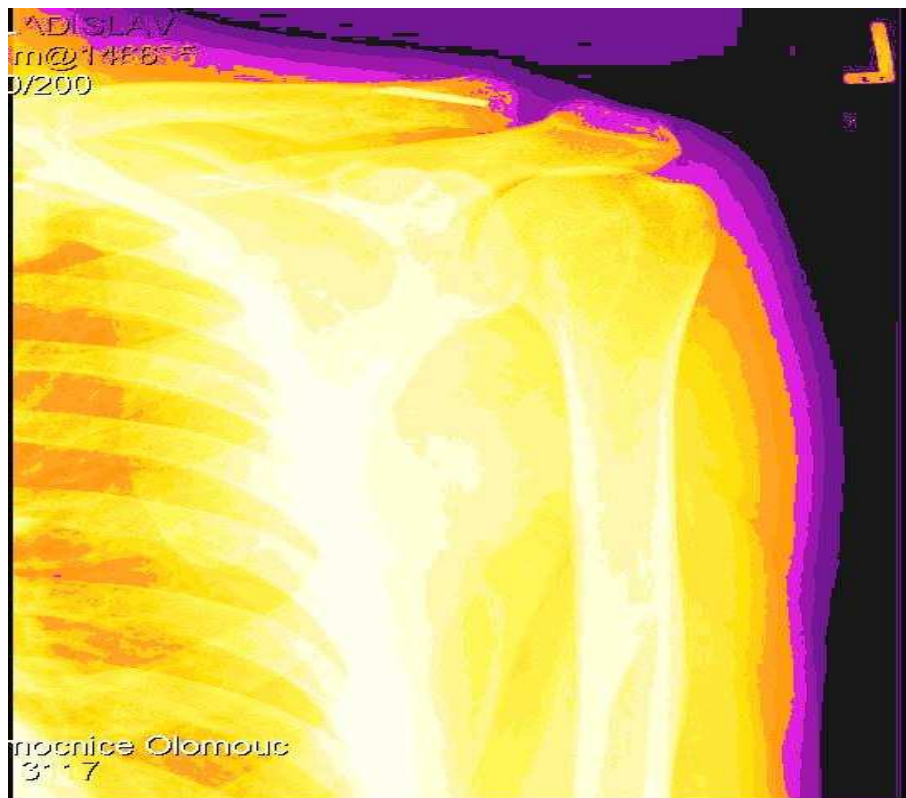
Obr. 8.12 Rentgenový snímek obarvený metodou IHS s parametrem 1,6



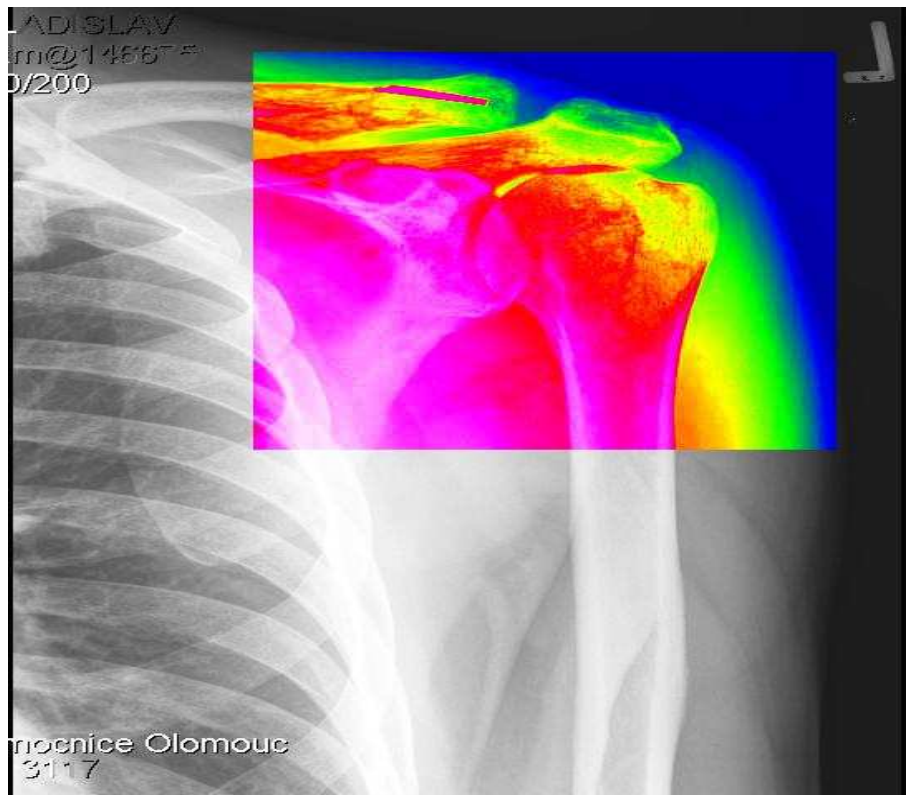
Obr. 8.13 Snímek obarvený metodou založenou na stupních jasu



Obr. 8.14 Snímek obarvený metodou barevné palety



Obr. 8.15 Snímek obarvený metodou hustotních řezů



Obr. 8.16 Snímek z obarveným výřezem

## 9 ZÁVĚR

Cílem diplomové práce bylo vytvořit funkční aplikaci pro zobrazování černobílých snímků v nepravých barvách. V první fázi diplomové práce byly podrobně prostudovány metody zobrazování v nepravých barvách a prozkoumána možnost jejich implementace do softwaru. V další fázi byl vytvořen samotný program Peacock, který byl postupně optimalizován, zejména možnosti uživatele ovlivnit přiřazení barev.

Při výběru metod byl kladen důraz na variabilitu metod. Tedy na možnost změny přiřazení barev uživatelem a tím ovlivnění výsledného vzhledu snímku tak, aby operace přinesla maximální zvýšení diagnostické výtěžnosti. Do nabídky programu Peacock bylo nakonec zařazeno šest různých metod. Těmito metodami jsou metoda založená na jednoduché parametrické rovnici (pracovní název Pseudo Křivka), metoda přímo aplikující vyhledávací tabulku (pracovní název Barevná paleta), metoda založená na stupních jasu (pracovní nazvaná Graylevels), metoda založená na barevných prostorech (pracovní nazvaná podle používaného modelu IHS), metoda hustotních řezů a mírně upravená metoda vyvinutá v laboratoři Yuvan v MIT (Pseudobarvicí krychle).

Nejprecizněji lze ovlivnit přiřazení barev v metodě založené na vyhledávací tabulce, ve které uživatel může vytvořit vlastní přiřazovací tabulku. Tedy přesně určit, jaké hodnoty barevných složek budou přiřazeny vybranému jasů. Vytvořenou paletu může uživatel uložit a použít i v budoucnu. U dalších metod může uživatel ovlivnit přiřazení barev volbou určitého parametru metody.

Efektivita využití zobrazování v nepravých barvách je do velké míry ovlivněna jasovou dynamikou vstupního snímku. V situaci, kdy je kontrast snímku malý (například u snímku přесvícených nebo naopak nedostatečně osvícených) a jas obrazových bodů obrazu se pohybuje jen v malém rozsahu jasové stupnice (0-255), nemusí pseudobarvení znamenat výrazné zlepšení diagnostické výtěžnosti snímku. Proto je užitečné před samotným pseudobarvením upravit snímek transformací jeho jasové stupnice vedoucí k zvětšení kontrastu snímku. Z toho důvodu jsou transformace jasové stupnice monochromatického snímku součástí programu Peacock. V aplikaci je na výběr z těchto metod ekvalizace, lineární transformace jasu, zvýraznění části histogramu, gama korekce a negativ.

Program umožňuje načtení obrázku v několika běžně používaných formátech (JPEG, BMP, GIF, PNG, TIFF). Po převodu do nepravých barev má uživatel několik možností. Může snímek bezprostředně vyhodnotit, uložit nebo vytisknout. Uložit lze snímek opět do několika běžně používaných formátů.

Součástí aplikace je relativně rozsáhlá nápověda, která kromě manuálu použití programu obsahuje také stručný popis použitých metod.

Nad rámec zadání práce byla do aplikace zařazena práce s videosekvencí. Vstupem aplikace tedy nemusí být jen statické snímky, ale také videosekvence. Program umí přehrát a pracovat s videem v běžných formátech (avi, mpeg). Uživatel může zachytit jednotlivý snímek videosekvence a ten zobrazit v nepravých barvách. Další možností je zachycení několika snímků následujících po sobě a jejich obarvení. Program tedy nabízí i jednoduché obarvení videa, i když ne v reálném čase.

Aplikace byla vytvořena v prostředí Visual Studio 2008 programovacím jazykem C#. Tato skutečnost má ten význam, že program bez problémů běží na všech PC a notebookech, kde je nainstalován operační systém Windows (XP, Vista, Windows 7). K jeho správné funkci není nutné mít nainstalované dané vývojové prostředí či jiný program. Jedinou podmínkou je, že systém musí mít nainstalován balíček Microsoft .NET Framework, který je však zpravidla nainstalován přímo při instalaci operačního systému. Instalace balíčku je přílohou elektronické verze diplomové práce.

Na úvod textu práce je čtenáři stručně předložena teorie barev v takovém rozsahu, který je potřebný pro pochopení dalšího textu. V další kapitole je zavedeno názvosloví používané v práci a jsou popsány různé způsoby zvýrazňování obrazu. Jedním ze způsobů je i zobrazení v nepravých barvách, kterému je věnována celá následující kapitola, v níž jsou teoreticky popsány metody implementované do aplikace Peacock. V další kapitole jsou objasněny metody transformace jasové stupnice použité v programu jako preprocessing před zobrazením v nepravých barvách. Text práce obsahuje také kapitulu, ve které jsou vysvětleny a často i pomocí vývojového diagramu popsány algoritmy použité při tvorbě programu. Následující kapitola má sloužit jako stručný manuál programu, je v ní popsáno ovládání programu a možnosti, které uživateli aplikace poskytuje. V poslední kapitole je čtenář seznámen s výsledky programu.

# POUŽITÁ LITERATURA

- [1] *Barvy* [online]. [2005] [cit. 2009-11-20]. Dostupný z WWW: <<http://www.volny.cz/lk77/barvy/>>.
- [2] WALES, Jimmy, et al. *Wikipedie : otevřená encyklopedie* [online]. 2001- [cit. 2009-11-22]. Dostupný z WWW: <<http://cs.wikipedia.org>>.
- [3] KADĚRKA, P. Systém pro zobrazování černobílých snímků v nepravých barvách (Pseudocolor). Brno: FEKT VUT v Brně, 2008. 53 s., 2 příl., vedoucí diplomové práce prof. Ing. Václav Říčný.
- [4] JAN, Jiří. Číslicová filtrace, analýza a restaurace signálů. Brno: Vysoké učení technické v Brně, 2002. 426s., ISBN 80-214-2911-9.
- [5] HOZMAN, J. *Základní metody předzpracování obrazu* [online]. 2003 [cit. 2009-04-13]. Dostupný z WWW: <[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)>.
- [6] LEHMAN, Thomas, KASER, Andreas, REPGES, Rudolf. *A simple parametric equation for pseudocoloring grey scale images keeping their original brightness progression* [online]. 1993-1996 [cit. 2009-11-23]. Dostupný z WWW: <[http://ganymed.imib.rwth-aachen.de/deserno/ps-pdf/IVC\\_1997-15%283%29251-257.pdf](http://ganymed.imib.rwth-aachen.de/deserno/ps-pdf/IVC_1997-15%283%29251-257.pdf)>.
- [7] DOUGLAS, Youvan. *YOUVAN FOUNDATION: Pseudocolor in Pure and Applied Mathematics* [online]. 2008 [cit. 2009-11-25]. Dostupný z WWW: <<http://www.youvan.com/>> .
- [8] LU, Xiang- Ju, DING , Ming-Xiao, WANG , Yun-Kua. *A New Pseudo-color Transform for Fibre Masses Inspection of Industrial Images* [online]. 2009 [cit. 2009-11-26]. Dostupný z WWW: <http://www.aas.net.cn/qikan/manage/wenzhang/2007-1284.pdf>.
- [9] SCOTT, Ed . *Masking Spectrum 5 - photo.net* [online]. 1997 [cit. 2009-11-27]. Dostupný z WWW: <<http://photo.net/learn/optics/edscott/ms000050.htm>>.
- [10] DOBROVOLNÝ, Petr. *Radiometrická zvýraznění* [online]. 2004 [cit. 2009-11-27]. Dostupný z WWW: <[http://www.geogr.muni.cz/archiv/vyuka/DPZ\\_CVICENI/Texty/DZO\\_04\\_zvyrazneni\\_1.pdf](http://www.geogr.muni.cz/archiv/vyuka/DPZ_CVICENI/Texty/DZO_04_zvyrazneni_1.pdf)>.
- [11] ŘÍČNÝ, V. *Videotechnika*. Skriptum FEKT VUT v Brně. Brno: FEKT VUT, 2003
- [12] DOBEŠ, Michal. *Zpracování obrazu a algoritmy v C#*. Praha : BEN, 2008. 144 s. ISBN 978-80-7300-23-6
- [13] HLAVÁČ, Václav; SEDLÁČEK, Miloš. *Zpracování signálů a obrazů*. Praha : ČVUT, 2007. 255 s. ISBN 978-80-01-03110-0.

# SEZNAM OBRÁZKŮ

OBR. 2.1 CITLIVOST ČÍPKŮ NA SVĚTLO V ZÁVISLOSTI NA VLNOVÉ DÉLCE [1].....	3
OBR. 2.2 MÍŠENÍ BAREV [1].....	4
OBR. 2.3 KOLORIMETRICKÝ OBRAZEC MKO [2] .....	5
OBR. 2.4 RGB MODEL [2] .....	5
OBR. 2.5 MODEL HSV [2] .....	6
OBR. 3.1 HISTOGRAM [5].....	7
OBR. 4.1 RGB PROSTOR SE ZOBRAZENOU PARAMETRICKOU KŘIVKOU [6].....	10
OBR. 4.2 KŘIVKA VYHOVUJÍCÍ ROVNICI ( PŘEVZATO Z [6]) .....	11
OBR. 4.3 URČENÍ FUNKCÍ R(T) A Z(T) [6].....	13
OBR. 4.4 UKÁZKA RŮZNÝCH PALET METODY VYUŽÍVAJÍCÍ PEVNOU TABULKU [9].....	14
OBR. 4.5 UKÁZKA FUNKCE METODY BAREVNÁ PALETA (PŮVODNÍ SNÍMEK PŘEVZAT [7]) .....	14
OBR. 4.6 PRINCIP ZOBRAZENÍ V NEPRÁVÝCH BARVÁCH ZALOŽENÝ NA STUPNÍCH JASU [8].....	15
OBR. 4.7 UKÁZKA APLIKACE METODY ZALOŽENÉ NA STUPNÍCH JASU (PŮVODNÍ SNÍMEK PŘEVZAT [7]) .	16
OBR. 4.8 UKÁZKA APLIKACE METODY ZALOŽENÉ NA PŘEVODU DO HSI MODELU K =1 (PŮVODNÍ SNÍMEK PŘEVZAT [7]) .....	17
OBR. 4.9 UKÁZKA APLIKACE METODY ZALOŽENÉ NA PŘEVODU DO HSI MODELU K =1,6 (PŮVODNÍ SNÍMEK PŘEVZAT [7]).....	18
OBR. 4.10 UKÁZKA APLIKACE METODY ZALOŽENÉ NA PŘEVODU DO HSI MODELU K =6 (PŮVODNÍ SNÍMEK PŘEVZAT [7]) .....	18
OBR. 4.11 BAREVNÝ PROSTOR S VYZNAČENÍM VÝVOJE BAREV V ZÁVISLOSTI NA JASU [7] .....	19
OBR. 4.12 UKÁZKA BAREVNÉ PALETY ZÍSKANÉ METODOU RGB BARVÍCÍ KRYCHLE.....	20
OBR. 5.1 CHARAKTERISTIKA VYTVOŘENÍ NEGATIVU .....	22
OBR. 5.2 UKÁZKA APLIKACE NEGATIVU, VLEVO PŮVODNÍ SNÍMEK [7] VPRAVO PO PROVEDENÍ TRANSFORMACE JASOVÉ STUPNICE V PROGRAMU.....	22
OBR. 5.3 PRINCIP ZVÝRAZNĚNÍ ČÁSTI JASOVÉ STUPNICE [10].....	23
OBR. 5.4 APLIKACE METODY ROZTAŽENÍ ČÁSTI JASOVÉ STUPNICE, VLEVO PŮVODNÍ PŘESVĚTLENÝ SNÍMEK [2] VLEVO UPRAVENÝ .....	23
OBR. 5.5 PRINCIP LINEÁRNÍHO ZVÝRAZNĚNÍ KONTRASTU [10].....	24
OBR. 5.6 UKÁZKA APLIKACE METODY LINEÁRNÍ ZVÝRAZNĚNÍ KONTRASTU, VLEVO PŮVODNÍ PŘESVĚTLENÝ [2] SNÍMEK VLEVO UPRAVENÝ .....	24
OBR. 5.7 UKÁZKA ÚČINKU EKVALIZACE, VLEVO PŘEEXPOZOVANÝ SNÍMEK [2] A VPRAVO TENTÝŽ SNÍMEK PO APLIKACE EGALIZACE .....	25
OBR. 5.8 UKÁZKA GAMA KOREKCE PŘI $\gamma = 0,5$ . VLEVO PŮVODNÍ SNÍMEK A JEHO PODOBA V NEPRÁVÝCH BARVÁCH. VPRAVO UPRAVENÝ SNÍMEK A JEHO PODOBA V NEPRÁVÝCH BARVÁCH .....	26
OBR. 6.1 VÝVOJOVÝ DIAGRAM PRÁCE S PROGRAMEM.....	28
OBR. 6.2 VÝVOJOVÝ DIAGRAM VNOŘENÝCH CYKLŮ FOR .....	36
OBR. 6.3 VÝVOJOVÝ DIAGRAM FUNKCE FZVYRAZNENI_CASTI .....	37
OBR. 6.4 VÝVOJOVÝ DIAGRAM FUNKCE FGRAYLEVELS .....	40
OBR. 6.5 VÝVOJOVÝ DIAGRAM FUNKCE FHSI .....	41
OBR. 6.6 VÝVOJOVÝ DIAGRAM GETBITMAPARRAYSAFE .....	42
OBR. 7.1 HLAVNÍ OBRAZOVKA APLIKACE PEACOCK .....	43
OBR. 7.2 ZÁLOŽKA TRANSFORMAČNÍ TABULKA .....	45
OBR. 7.3 ZÁLOŽKA VIDEO.....	46
OBR. 8.1 SNÍMEK PŘEVEDENÝ DO FALEŠNÝCH BAREV PROGRAMEM MIPS.....	48
OBR. 8.2 SNÍMEK PŘEVEDENÝ DO FALEŠNÝCH BAREV PROGRAMEM PEACOCK.....	48
OBR. 8.3 UKÁZKA PRÁCE PROGRAMU MULTISPEC.....	49
OBR. 8.4 PŮVODNÍ RENTGENOVÝ SNÍMEK .....	49
OBR. 8.5 SNÍMEK OBARVENÝ METODOU BARVÍCÍ KRYCHLE S PARAMETREM 1. ....	50
OBR. 8.6 SNÍMEK OBARVENÝ METODOU BARVÍCÍ KRYCHLE S PARAMETREM 2. ....	50
OBR. 8.7 SNÍMEK UPRAVENÝ ŠEDOTÓNOVU ÚPRAVOU ZVÝRAZNĚNÍ ČÁSTI STUPNICE A OBARVENÝ METODOU BARVÍCÍ KRYCHLE S PARAMETREM 1.....	51
OBR. 8.8 SNÍMEK UPRAVENÝ ŠEDOTÓNOVU EKVALIZACE A OBARVENÝ METODOU BARVÍCÍ KRYCHLE ...	51

OBR. 8.9 RENTGENOVÝ SNÍMEK OBARVENÝ METODOU KŘIVKY S PARAMETRY ( $\Omega = 4$ A $\Phi = 6$ ) .....	52
OBR. 8.10 RENTGENOVÝ SNÍMEK OBARVENÝ METODOU KŘIVKY S PARAMETRY ( $\Omega = 10$ A $\Phi = 8$ ) .....	52
OBR. 8.11 RENTGENOVÝ SNÍMEK OBARVENÝ METODOU KŘIVKY S PARAMETRY ( $\Omega = 20$ A $\Phi = 8$ ) .....	53
OBR. 8.12 RENTGENOVÝ SNÍMEK OBARVENÝ METODOU IHS S PARAMETREM 1,6.....	53
OBR. 8.13 SNÍMEK OBARVENÝ METODOU ZALOŽENOU NA STUPNÍCH JASU .....	54
OBR. 8.14 SNÍMEK OBARVENÝ METODOU BAREVNÉ PALETY .....	54
OBR. 8.15 SNÍMEK OBARVENÝ METODOU HUSTOTNÍCH ŘEZŮ.....	55
OBR. 8.16 SNÍMEK Z OBARVENÝM VÝŘEZEM .....	55