VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

METODY PRO PŘEVOD BAREVNÝCH VIDEOSEKVENCÍ NA ČERNOBÍLÉ

COLOR-TO-GRAYSCALE VIDEO CONVERSIONS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **FILIP BŘEZNA**
AUTHOR

VEDOUCÍ PRÁCE **doc. Ing. MARTIN ČADÍK, Ph.D.**
SUPERVISOR

BRNO 2018

Zadání bakalářské práce/21014/2017/xbrezn00

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2017/2018

Zadání bakalářské práce

Řešitel: **Března Filip**
Obor: Informační technologie
Téma: **Metody pro převod barevných videosekvencí na černobílé**
Color-to-Grayscale Video Conversions
Kategorie: Počítačová grafika

Pokyny:

1. Seznamte se s problematikou konverze barevných obrázků na černobílé.
2. Vyberte a popište metody vhodné pro implementaci. Při výběru se soustřeďte na metody vhodné pro převod videa.
3. Do již existujícího systému implementujte alespoň tři zvolené metody.
4. S metodami experimentujte, posuďte jejich vlastnosti při převodu videosekvencí, implementované algoritmy porovnejte a diskutujte možnosti budoucího vývoje.
5. Dosažené výsledky prezentujte formou videa, plakátu, článku, apod.

Literatura:

1. http://cadik.posvete.cz/
2. http://cadik.posvete.cz/color_to_gray_overview/

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Čadík Martin, doc. Ing., Ph.D.**, UPGM FIT VUT
Datum zadání: 1. listopadu 2017
Datum odevzdání: 16. května 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Převod barevného obrazu na stupně šedi je stále aktuálním tématem a nachází uplatnění v několika oblastech, mezi které patří nejen umělecké fotografování, ale především nebarevný tisk a usnadnění některých operací při zpracování obrazu. Tato bakalářská práce se zabývá právě tímto převodem, který se nazývá dekolorizace, a to se zaměřením na videosekvence. Je zde vysvětlen základní princip reprezentace digitálního snímku a operace pro jeho zpracování využité v další části práce. Poté jsou zde také analyzovány typy metod, které se pro převod na stupně šedi používají a dále jsou rozebrány jejich vlastnosti i úspěšnost jejich použití. V druhé části práce jsou popsány tří vybrané implementované dekolorizační metody a sourhn jejich dosažených výsledků, jež je následně doprovázen zhodnocením praktické použitelnosti v oblasti převodu barevných videosekvencí na videosekvence ve stupních šedi.

Abstract

Color to grayscale conversion is still a relevant topic and finds use in multiple areas, to which belongs artistic photography, but primarily colorless print and for simplifying some processes in picture processing. This Bachelor thesis deals with this conversion - decolorization, with focus on video sequences. The basic principle of representation of digital image is explained here alongside with operations for its processing used in next part of thesis. Thereafter it includes the analysis of types of methods, which are used for the conversion to grayscale and further their properties and success rate are evaluated aswell. In the second part of this thesis, there are three chosen and implemented decolorization methods and summary of their achieved results, which is then accompanied by evaluation of practicability in conversion of colorful video sequences to grayscale ones.

Klíčová slova

fotografie, video, C++, dekolorizace, převod barevného obrazu na stupně šedi, umělá inteligence, strojové učení

Keywords

photograph, video, C++, decolorization, color to grayscale, artificial intelligence, machine learning

Citace

BŘEZNA, Filip. *Metody pro převod barevných videosekvencí na černobílé*. Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. Ing. Martin Čadík, Ph.D.

Metody pro převod barevných videosekvencí na černobílé

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana doc. Ing. Martina Čadíka, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Filip Března
17. května 2018

Poděkování

Tímto bych rád poděkoval panu doc. Ing. Martinu Čadíkovi, Ph.D. za odbornou pomoc, ochotu a trpělivost při vedení této bakalářské práce.

Obsah

1 Úvod **3**

2 Převod barevného obrazu na obraz ve stupních šedi **4**
2.1 Černobílý obraz a obraz ve stupních šedi 4
2.2 Reprezentace digitálního obrazu 4
2.2.1 RGB model 5
2.2.2 CIELab model 5
2.3 Pomocné transformace pro zpracování obrazu 7
2.3.1 Kvantizace barev v obraze 7
2.3.2 Detekce hran v obraze 9
2.3.3 Rozmazání obrazu 11
2.4 Strojové učení 12
2.4.1 Strojové učení s učitelem 12
2.4.2 Strojové učení bez učitele 12
2.5 Algoritmus sestupu gradientů (Gradient descent) 13
2.5.1 Adaptivní odhad momentu (Adaptive Moment Estimation) 14

3 Implementované metody pro dekolorizaci **15**
3.1 Použité technologie 15
3.1.1 Využité knihovny 15
3.2 Typy konverzních metod 16
3.2.1 Globální metody převodu 16
3.2.2 Lokální metody převodu 16
3.3 Efektivní dekolorizace zachovávající významné rozdíly 17
3.3.1 Dosažené výsledky 20
3.4 Parametrická poměrově založená metoda pro efektivní zachování kontrastu 21
3.4.1 Dosažené výsledky 24
3.5 Zachování vnímaného kontrastu v dekolorizaci pomocí optimalizovaných barevných řad 25
3.5.1 Optimalizace 27
3.5.2 Dosažené výsledky 28

4 Porovnání implementovaných metod **29**
4.1 Implementační náročnost metod 29
4.2 Časová náročnost metod 30
4.3 Paměťová náročnost metod 30
4.4 Hodnocení použitelnosti metod 31

5 Závěr **32**

Literatura **33**

A Porovnání dosažených výsledků jednotlivých metod **36**

Kapitola 1

Úvod

Dekolorizace, tedy proces převodu barevného obrazu na stupně šedi, je obor jejímž zdokonalováním a snahou získat co možná nejlepší výsledný kontrast se zabývají experti již skoro dvě dekády. Téměř všechny fotoaparáty dnes slouží k zachycování barevných snímků s výjimkou uměleckých fotografií ve stupních šedi, kterými proslul například americký fotograf Ansel Adams.

Zmíněná disciplína však nachází uplatnění především pro velmi oblíbený nebarevný tisk, což snižuje potřebné finanční náklady a v mnoha případech není nežádoucím aspektem. Tato skutečnost se pravděpodobně v blízké době nebude měnit. A to je společně s monochromními zařízeními, tedy zařízeními, které nepodporují ani základní barvy, například čtečky elektronických knih, hlavním důvodem, proč se dekolorizace stejně jako k ní inverzní operace kolorizace těší stálému studiu vedoucímu k vylepšením v mnoha směrech. Dalším významným důvodem je však dekolorizace pro obor počítačového vidění, kde je v mnoha případech zapotřebí nalézt extrémy na snímku. Tyto extrémy se lépe hledají na nebarevném obraze, kde není vyhodnocení operací tak časově náročné. S přesnější dekolorizací se pak logicky může dosáhnout přesnějších výsledků.

Běžné zastaralé metody převodu počítají odstíny šedi pouze na základě kanálu zaznamenávající jas. Tento způsob úplně ignoruje sytost barvy, což vede ke ztrátě některých informací a parazitnímu jevu; mapování lidským okem jednoznačně odlišitelných barev se stejným jasem na totožné odstíny šedi. Tento nemalý problém inicioval hledání dokonalejších způsobů a některými z nich se zabývá právě tato bakalářská práce.

V kapitole 2 je rozebrána teorie k jednotlivým technikám zpracování obrazu použitých při implementaci konverzních metod. Nejprve je zde popsáno, jakými způsoby lze reprezentovat digitální snímek, poté následuje motivace k procesu dekolorizace a výše zmíněné obrazové transformační techniky.

Další kapitola pak vysvětluje implementační detaily konverzních metod včetně funkcí, které byly použity. Na začátku ještě rozvíjí typy dekolorizačních metodik a specifikuje následné zaměření práce.

V předposlední kapitole je pak popsáno porovnání implementovaných metod a úvaha nad jejich případným využitím v praxi, a to hlavně v oblasti videosekvencí.

Na závěr je celá práce zhodnocena, komentovány jsou především slabé články, aby bylo zcela zřejmé, kde je prostor pro zlepšení, a také aby bylo jasné jaké výsledky lze od vytvořených metod očekávat.

Kapitola 2

Převod barevného obrazu na obraz ve stupních šedi

V této kapitole je rozebrána metodika pro získání obrazu ve stupních šedi ze zdrojového snímku v barevné formě. V první části je vysvětlen rozdíl mezi černobílým obrazem a obrazem ve stupních šedi. Dále je zmíněna teorie reprezentace digitálního snímku jako takového včetně modelů použitých v této práci, a poté v následující části je popsána samotná teorie převodu barevného obrazu na obraz ve stupních šedi.

2.1 Černobílý obraz a obraz ve stupních šedi

Přestože jsou tyto dva pojmy často zaměňovány, existuje mezi nimi značný rozdíl. U černobílého obrázku existují jen a pouze dvě barvy – černá a bílá. Není možné vytvořit jakýkoliv odstín šedi. Takový obrázek lze teoreticky reprezentovat jen dvěma hodnotami; kladnou hodnotou pro bílou a jakoukoliv jinou pro černou. Naproti tomu stupně šedi umožňují do jisté míry i vnímání barvy. Toto vnímání je spíše vlastností našeho mozku, který se snaží chápat obrazový vjem v podobě něčeho známého, a tak si může v některých situacích dosadit barvu i do scenérie, kde žádná k nalezení není. Protože je obrázek ve stupních šedi složen z bílé, černé a teoreticky všech možných odstínů mezi těmito dvěma barvami, dokáže zachytit větší detail než předchozí varianta.

2.2 Reprezentace digitálního obrazu

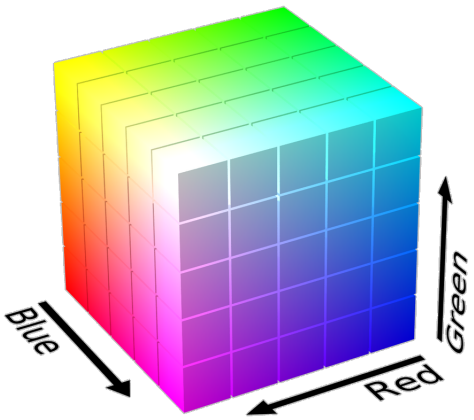
Obraz se na digitálních zařízení musí ukládat v souladu s limitacemi hardwaru. Tyto limitace daly za vznik dvěma způsobům, jak reprezentovat binární data jako obraz; vektorový obraz a rastrový obraz.

Vektorový obraz je reprezentován souřadnicemi a délkami jednotlivých vektorů, díky čemuž se ani při extrémním přiblížení obrázek nerozpadne na artefakty, a tedy nedojde k žádné degradaci kvality. Naproti tomu je rastrový obraz uložen jako matice intenzit jednotlivých barevných kanálů. V případě obrazu ve stupních šedi existuje jedna matice s intenzitami, u barevného snímku existuje těchto matic více, pro každý barevný kanál jedna. Složením těchto matic přes sebe dostaneme výsledný obraz, který díky míchání intenzit z barevného kanálu dosáhne věrné podoby barev své předlohy. Nejmenší takto zobrazený bod se nazývá pixel.

Protože by zachycení složité scenérie jako vektorový obraz bylo náročné na čas i prostředky, používá se v běžném životě spíše obraz rastrový. Vektorové obrazy nachází své využití spíše pro méně náročné ilustrace, jako jsou diagramy, loga a plakáty. V této práci jsou veškeré principy a metody prováděny na rastrových obrazech.

2.2.1 RGB model

Tento barevný model využívají barevné obrazovky a velmi často bude zmiňován v následujících kapitolách. Bývá většinou zobrazovaný jako krychle, kterou zachycuje obrázek 2.1. Skládá se ze třech základních kanálů; R (Red) – červená, G (Green) – zelená, B (Blue) – modrá. Sloučením těchto složek lze definovat celou barevnou paletu. Při nulové intenzitě všech tří kanálů získáváme barvu černou, v opačném případě pak bílou. Hodnoty jsou typicky ve většině souborových formátů reprezentovány osmibitovým číslem, a tedy v číselném rozsahu 0 až 255. Avšak existují i formáty s přesnějším 16bitovým rozmezím úrovně intenzity[21].



Obrázek 2.1: RGB model zachycený pomocí 3D RGB kostky; Převzato z [4].

2.2.2 CIELab model

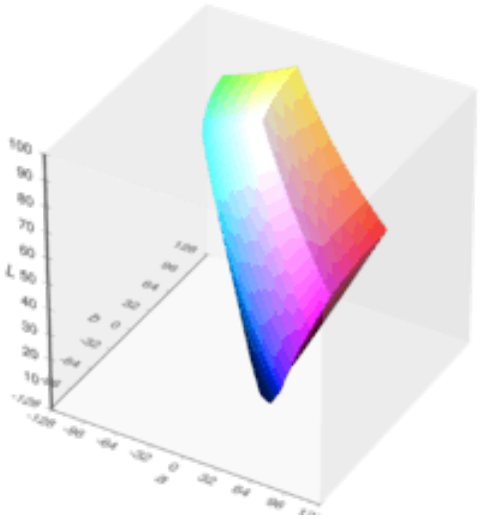
V roce 1892 přišel Ewald Hering s teorií, kterou nazval vzájemně exklusivním zpracováním barev[20]. Definoval dva páry barev; zelenou – červenou a žlutou – modrou. Domníval se, že jsou tyto dvě dvojice výjimečné právě tím, že z nich lze všechny ostatní barvy nakombinovat. Poukázal na fakt, že buď je zobrazena barva zelená anebo červená, nikoliv však jejich kombinace, a to samé platí rovněž pro druhý pár. Přestože žlutá vzniká v RGB modelu spojením červené a zelené, zdravé lidské oko to tak nevnímá.

Z těchto poznatků vychází barevný model Lab[6]. Stejně jako v RGB modelu zde máme tři složky. První z nich se značí L – jas (odvozeno z anglického slova Lightness), jehož hodnota přísluší v rozsahu nula až sto. Dále se zde nachází složka a, která nám definuje v plné intenzitě červenou a v opačném případě pak zelenou barvu. Pro poslední složku b platí, že nulová intenzita odpovídá modré a maximální intenzita odpovídá žluté barvě. Tyto dvě skutečnosti jsou inspirovány právě výše uvedenou teorií.

Lab model tedy umožňuje interpretovat všechny barvy, které lze vidět lidským okem, a právě v tomto ohledu předčí model RGB, kde není možné definovat tak velké množství

variant. Další nespornou výhodou tohoto modelu je schopnost identicky zobrazit stejný snímek na různých zařízeních.

V dnešní době se však více uchytila vylepšená varianta CIELab[6] (někdy značena rovněž jako L*a*b*), kde se provádí krychlová transformace, nikoliv transformace podle odmocniny jako v Lab. Tento model je velmi používaný při dekolorizaci obrazu a dobře jej popisuje také obrázek 2.2.



Obrázek 2.2: CIELab model; Převzato z [6].

2.3 Pomocné transformace pro zpracování obrazu

Počítačové zpracování obrazu může v určitých situacích vyžadovat úpravu zdrojového obrazu tak, aby metoda, která daný problém řeší, mohla následně provést všechny potřebné kroky. Některé postupy například vyžadují mírné rozostření obrazu, kvantizaci barev a podobně, aby vrátily data. Stručný popis těchto pomocných operací nad snímky je popsán v následujících podkapitolách.

2.3.1 Kvantizace barev v obraze

Kvantizace barev[23] je proces ztrátové redukce celkového počtu odlišných barev v obraze, jehož cílem je značné snížení počtu datových informací, aniž by došlo k významové degradaci snímku. V dnešní době není žádný problém s ukládáním obrázků ve vysoké kvalitě, kapacity úložných prostorů jsou pro tyto účely dostatečné. Potíž však nastává v oblasti zpracování obrazu, kde při velkém rozlišení vstupního objektu mohou modifikace trvat příliš dlouho. V takovém případě se nabízí jako možnost právě nejprve využití kvantizace barev, která probíhá běžně ve dvou krocích. Prvním z nich je určení barevné palety, která obvykle obsahuje 8 až 256 barev. Proces takové volby spadá mezi nejednoduché úkoly, jelikož hranice možné ztráty kvality není obecně daná a může být subjektivní záležitostí. Dalším krokem je nahrazení všech barev ve zdrojovém snímku těmi z palety. Zužitkování kvantizace barev přichází v situacích jako je například obrazová segmentace (automatické rozdělování obrazu na oblasti se společnými vlastnostmi), komprese, analýza barevných textur anebo v našem případě dekolorizace.

Obrázek 2.3: Operace kvantizace barev v snímku, použita metoda K-means s 30 clustery

Dělení metod kvantizace barev

a) Metody dělení barevných prostorů

Tyto metody rozdělují barevnou paletu podle různých kritérií na oddělené regiony. Tento proces se opakuje, dokud nenastane situace, kdy máme žádoucí počet regionů a v takovém případě se průměrná barva v regionu stává jednou z hledaných kvantizovaných barev. Hlavní výhodou těchto metod je nízká časová náročnost, avšak ne vždy nalezneme přímo optimální řešení. Za nejjednodušší příklad můžeme považovat uniformní kvantizaci, kde rozdělíme dvě barevné osy na osm dílů a třetí zbývající na čtyři. Volba os je na nás, avšak často se na čtyři části dělí modrá osa. Takovým výsledkem by pak bylo 256 regionů. Na první pohled tento

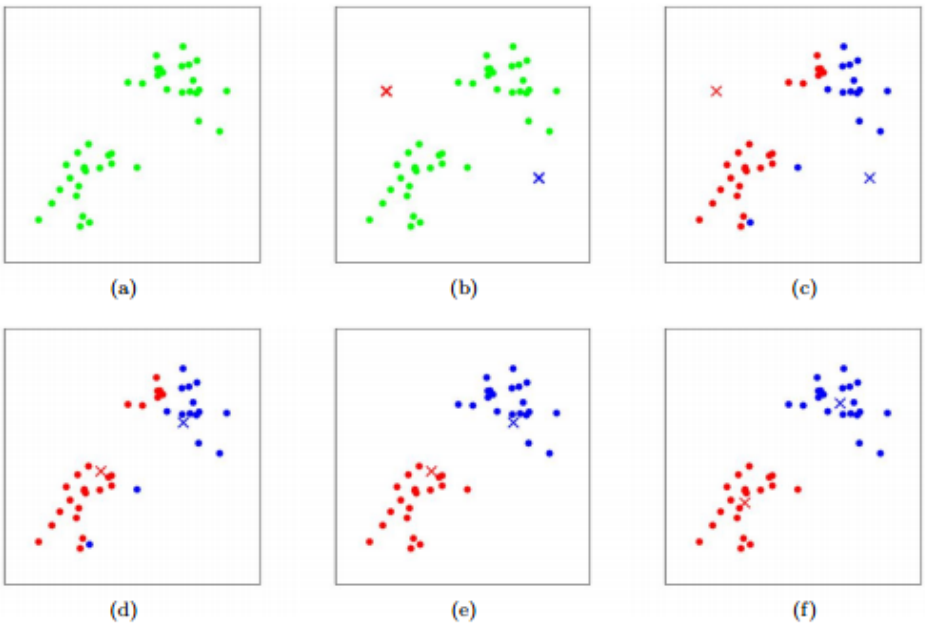
způsob nemá vůbec nic společného s barvami na vstupním obrazu, a proto neposkytuje dostatečně uspokojivé řešení.

b) Metody založené na clusterech

Jedná se o pokročilejší metody s vyšší časovou náročností, avšak poskytující optimálnější řešení než je tomu u metod dělení barevného prostoru. Důležitý parametr představují počáteční podmínky, jejich volba ovlivňuje kvalitu a rychlost řešení a neměla by tedy být podceňována. Cluster zde reprezentuje skupinu bodů s podobnými vlastnostmi, podle nichž jsou zmíněné body slučovány.

K-means

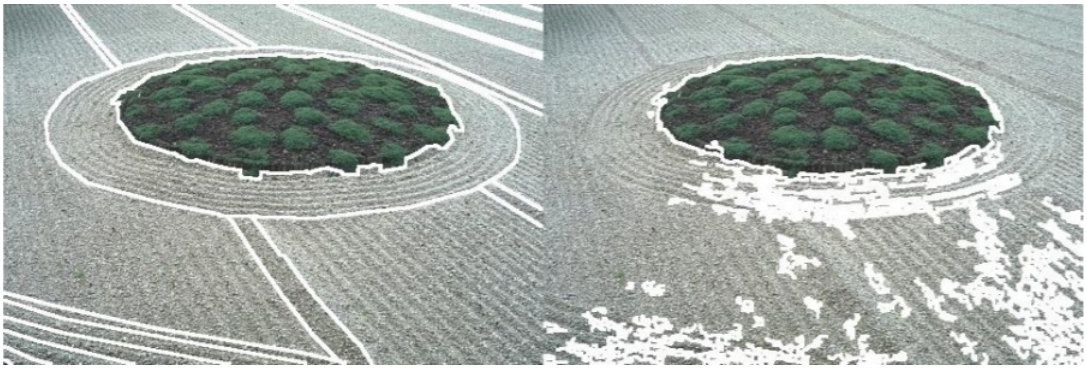
Jedná se o jeden z nejpopulárnějších algoritmů[24] v kategorii metod založených na clusterech využívající principů řízeného strojového učení 2.4. K-means slouží ke klasifikaci vstupních dat, v našem případě barevných pixelů, do zvoleného počtu clusterů. Tento počet odpovídá proměnné K a typicky se její hodnota pohybuje v řádech jednotek či desítek. K-means rovněž obsahuje stejný počet centroidů, což jsou středové body jednotlivých clusterů. Bod reprezentovaný vstupními daty připadne právě do takového clusteru, který odpovídá centroidu nejblíže položenému zkoumanému bodu. Nejprve se tedy, jak již bylo zmíněno, přidělí body do clusterů, a poté se začnou přemisťovat centroidy tak, aby lépe odpovídaly středu pro přidělené body. Tyto dvě operace se automaticky neustále opakují, dokud nedojde k nalezení optimálního řešení.



Obrázek 2.4: a) Originální datová sada; b) Náhodné rozdělení centroidů; c) – f) Iterace K-means hledající správnou polohu centroidů; Převzato z [24].

2.3.2 Detekce hran v obraze

Detekce hran[11] se stala oblíbeným fenoménem v oblasti zpracování obrazu, a i přestože se jejím studiem experti zabývají již téměř tři dekády, stále se jedná o jeden z nejtěžších úkolů v této oblasti. Slouží především k hledání extrémů, které mohou znázorňovat pro nás důležité objekty, a někdy také ke snížení počtu datových informací na vstupním snímku. Je dobré si uvědomit, že výsledek procesu hledání obyčejných hran se neshoduje s výsledkem při pátrání po hranicích oddělující regiony. Tuto skutečnost zachycuje obrázek 2.5.



Obrázek 2.5: Rozdíl mezi detekcí hranic regionů (vlevo) a detekcí hran (vpravo); Převzato z [22].

Existuje spousta matematických způsobů, jak tuto problematiku řešit, ale principiálně často sledují ostrý skok v hodnotě kanálu jasu mezi sousedními pixely. Tyto metody se rozdělují do dvou skupin podle principů, se kterými pracují[15]. První z nich je založena na zpracování gradientů popsaných v následující části kapitoly a druhá na bázi využití Laplaceova operátoru.

a) Detekce založená na výpočtu gradientů

Gradient[16] značíme ∇ (čteme nabla) a jedná se o techniku, která nám slouží k výpočtu rychlosti změny hodnot v obraze. Při výskytu hrany dochází k náhlému velkému rozdílu sousedních hodnot pixelů. V naší situaci, kde zpracováváme fotografii, tedy dvoudimenzionální objekt, pak gradient odpovídá vektoru nesoucímu informaci o změnách ve vertikálním a horizontálním směru, což reprezentuje směrovou orientaci, ve které dochází k největší odchylce.

Gradient je počítán pro každý obrazový pixel a tuto operaci matematicky popisuje vzorec:

$$\nabla I = (G_x, G_y) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right), \tag{2.1}$$

kde $\frac{\partial I}{\partial x}$ a $\frac{\partial I}{\partial y}$ značí změnu v obraze I v daném směru osy a je možné ji vypočítat pomocí:

$$\frac{\partial I}{\partial x} = I(x, y) - I(x + 1, y), \tag{2.2}$$

$$\frac{\partial I}{\partial y} = I(x, y) - I(x, y + 1). \tag{2.3}$$

Často nás také zajímá velikost gradientu, kterou lze snadno vypočítat ze získaného vektoru:

$$|\nabla I| = \sqrt{{G_x}^2 + {G_y}^2}. \tag{2.4}$$

Pro metody detekce hran využívající gradienty se používá vzorce založeného na předchozích principech:

$$|\nabla(\mathrm{f}(x,y))| = \sqrt{\left(\frac{\partial \mathrm{f}(x,y)}{\partial x}\right)^2 + \left(\frac{\partial \mathrm{f}(x,y)}{\partial y}\right)^2}, \tag{2.5}$$

kde funkce f zastupuje definici jádra (kernel), o které je tato metodika rozšířena; jádro představuje filtr různé, zpravidla liché velikosti. Kernel je přiložen na zkoumaný pixel ve zdrojovém obraze a součtem násobků jeho koeficientů s odpovídajícími pixely je získána jedna výsledná hodnota pro středový bod. A právě tato hodnota porovnáním se zvolenou prahovou hranicí určuje, zda došlo k vzniku hrany v obraze.

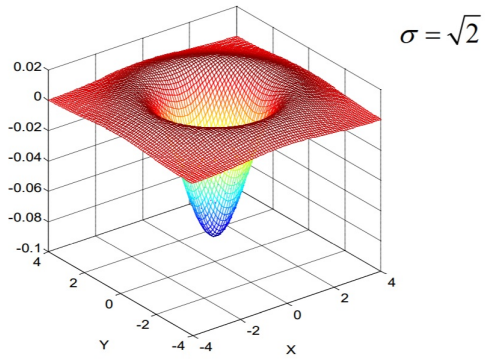
b) Detekce pomocí Laplace z Gaussianu (LoG)

Pokud dochází k pomalým přechodům intenzit luminanční složky, představuje vhodnější variantu využití Laplaceova operátoru, nikoliv výpočty gradientů. Tento způsob dokáže zachytit díky druhé derivaci i průchod funkce nulou a zpřesnit takový nález.

Jelikož jsou derivační metody náchylné na šum, způsobí metodika nejprve rozostření obrazu pomocí Gaussianova filtru 2.3.3, což vede k odstranění šumu a poté následuje filtrování pomocí Laplaceova operátoru Δ (skalární součin dvou gradientů). Tyto dvě operace mohou být spojeny konvolucí rovnou na LoG, který popisuje vzorec (2.6).

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left(1 - \frac{x^2+y^2}{2\sigma^2}\right)\mathrm{e}^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.6}$$

Tento vzorec neaplikujeme přímo na vstupní obrazové body, nýbrž jej využijeme k výpočtu jádra o liché velikosti. Střed jádra pak odpovídá hodnotám [0, 0] pro proměnné [x, y], z toho vyplývá, že mohou indexy charakterem příslušet i záporu. Ilustrační obrázek 2.6 zachycuje jádro o velikosti 9x9 s použitím $\sigma = 2$.



Obrázek 2.6: Laplace z Gaussianu; Převzato z [15].

2.3.3 Rozmazání obrazu

Důvod provádění operace rozmazání obrazu nemusí být intuitivně vůbec zřejmý a může působit velmi kontraproduktivně. Faktem však je, že při zachycení fotografie digitálním fotoaparátem dochází ke vzniku šumu v obraze. Tento nežádoucí jev narušuje kvalitu snímku, ale může být odstraněn pomocí rozmazání obrazu. Dva nejpoužívanější způsoby jak toho lze docílit:

a) Filtr dolní propust

Dolní propust je poměrně jednoduchá metodika pro rozostření obrazu[8], která odstraňuje rychlé změny intenzity průměrováním každého pixelu i s okolními body podle zvolené masky (jádra). Pro každou takovou masku platí, že součet obsažených koeficientů vždy odpovídá hodnotě jedna, jinak by docházelo k úplné změně vstupního obrazu. Nejjednodušší možné jádro představuje:

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \tag{2.7}$$

Obrázek 2.7: Rozostření obrazu pomocí jádra 2.7

b) Gaussianovo rozostření

Název Gaussianova filtru je na základě své podobnosti se zvonem odvozen od Gaussianovy křivky, která se často objevuje v matematické oblasti statistiky. Rozmazání pomocí Gaussianu umožňuje jednodušší manipulaci, jelikož má pouze jeden vstupní parametr volitelný uživatelem a to σ (sigma), který udává stupeň rozostření. Filtr opět pracuje i s okolními body, avšak na rozdíl od klasické dolní propusti přikládá počítanému pixelu největší váhu a těm okolním s rostoucí vzdáleností od takového středového bodu stále nižší váhu. Takto pokračuje až dojde do fáze, kdy jsou vzdálené body zanedbatelné a vůbec se neuvažují. Vzorec pro výpočet Gaussianova jádra, které poté opět aplikujeme nad obrazem je definován jako:

$$G(x,y) = \frac{1}{2\pi\sigma^2}\mathrm{e}^{-\frac{x^2+y^2}{2\sigma^2}}. \tag{2.8}$$

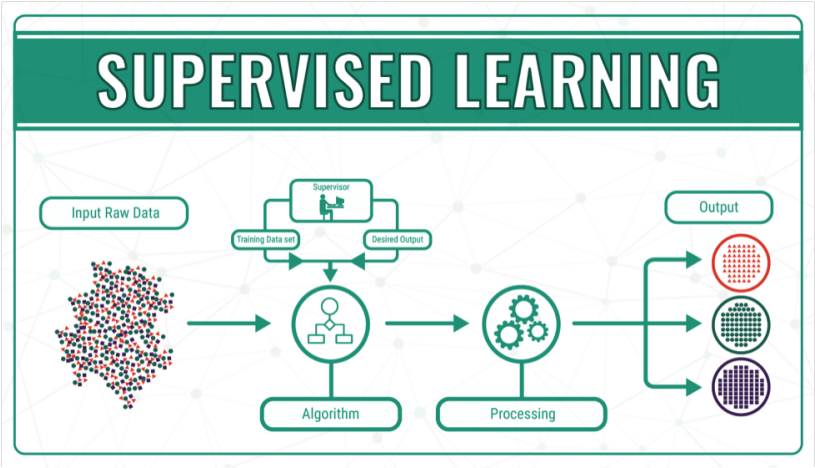
Obrázek 2.8: Gaussianovo rozostření; StDev představuje různé hodnoty σ; Převzato z [3].

2.4 Strojové učení

Arthur Samuel slibně definoval strojové učení jako vědní obor, který dává počítači schopnost učit se novým věcem, aniž by tyto způsoby získání vědomostí byly přímo programovány.

2.4.1 Strojové učení s učitelem

V souvislosti se strojovým učením je většinou spojováno strojové učení s učitelem[19]. U takové varianty je podstatné, že algoritmus nedostává pouze vstupní data, ale i očekávané řešení, jeho úkolem pak je se samostatně naučit ze vstupu získat právě takový výstup, který bude odpovídat referenčnímu výsledku. V případě neúspěchu se algoritmus vrací a zkouší opakovaně získat správné výstupy.



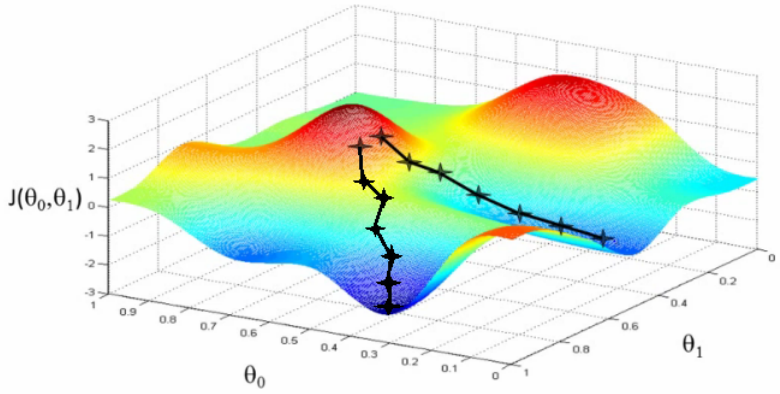
Obrázek 2.9: Diagram znázorňující princip řízeného strojového učení; Převzato z [19].

2.4.2 Strojové učení bez učitele

Strojové učení bez učitele[19] nemá ve většině případů datovou sadu a výstup zde nebývá známý. Přestože jsou algoritmy takového typu kvůli své náročnosti málo rozšířené, nabízí velký potenciál v oblasti strojového učení, jelikož umožňuje nalézt řešení na neomezené množství druhů nejen vstupních dat, ale rovněž situací.

2.5 Algoritmus sestupu gradientů (Gradient descent)

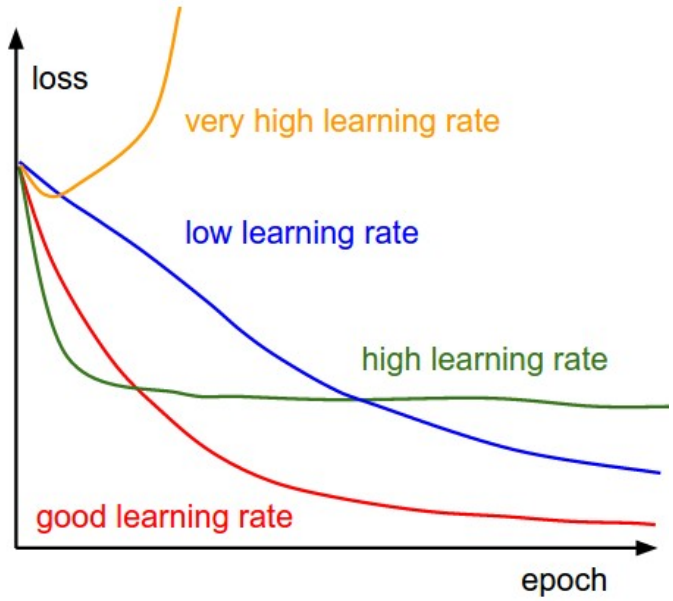
Na řešení některých operací je možné použití hrubé síly, tedy porovnávání výstupních hodnot pro úplně všechny možné vstupy. Tento proces je časově náročný a mnohdy si jej nemůžeme dovolit. V takových situacích se využívá optimalizačních algoritmů, které nehledají ideální řešení slepě, nýbrž posuzují vzniklé situace a snaží se rychleji přiblížit ke kýženému výstupu. Právě Gradient descent[26] patří mezi optimalizační algoritmy strojového učení 2.4, využívaný často při výskytu nekonvexních funkcí, což jsou funkce s více než jedním lokálním minimem, a tedy proces hledání globálního minima představuje značný problém. Pro vysvětlení této metodiky bude použita známá abstrakce. Nacházíme se na vrcholu nějaké hory a naším úkolem je najít nejníže výškově položený bod v celém pohoří. Problém způsobuje skutečnost, že máme zavázané oči a žádný přehled o okolí. Jedinou možností v takové situaci je tedy prozkoumání povrchu velmi blízkého okolí a vydání se směrem největšího poklesu, tento proces opakujeme v naději nalezení úplného minima. Algoritmus Gradient descent hledá stejným naivním způsobem, přesto však tato metodika není vůbec triviální a představuje veliký rozdíl oproti náhodnému hledání či testování všech variant. Obrázek 2.10 znázorňuje tento popsaný přístup.



Obrázek 2.10: Algoritmus sestupu gradientů; θ představuje parametr algoritmu, $J(\theta)$ udává cenu; Převzato z [26].

Existují dva hlavní druhy algoritmu poklesu gradientu. Prvním z nich jsou metody Plné dávky (Full batch), kde se v každém kroku pracuje se všemi vstupními testovacími daty. Druhým příkladem jsou metody stochastické, kde se v každém kroku bere pouze vzorek nějakého úseku.

Všechny metody poklesu gradientů berou jako vstupní parametr rychlost učení, kterou představuje konstanta udávající, jakou rychlostí se chceme blížit k minimu po kterém pátráme. Volba této hodnoty je velmi důležitá, jelikož v případě malé rychlosti trvá celý proces příliš dlouho a v případě rychlosti nadmíru velké, naopak hledané minimum nemusí být vůbec objeveno. Na tento problém poukazuje obrázek 2.11.



Obrázek 2.11: Volba parametru rychlosti učení; Převzato z [26].

2.5.1 Adaptivní odhad momentu (Adaptive Moment Estimation)

Adaptivní odhad momentu[12] (zkráceně se často setkáme s názvem Adam) je rozšířením pro stochastické metody poklesu gradientu. Vylepšení spočívá v přepočítávání parametru rychlosti učení. Adam je populární a doporučovaný algoritmus poskytující rychlé a uspokojivé výsledky.

Kapitola 3

Implementované metody pro dekolorizaci

Cílem této bakalářské práce je implementace tří dekolorizačních algoritmů podle vzorových studií představovaných na zahraničních konferencích. A právě touto tématikou se detailně zabývá tato kapitola. Jsou zde popsány nejen matematické vzorce, kterých bylo využito, ale také principy s kterými pracují.

3.1 Použité technologie

Veškerý implementovaný zdrojový kód je psán v programovacím jazyce *C++*, který patří mezi jazyky kompilované a provádění programu reprezentujícího dekolorizační metody pak bude rychlejší, než v případě kdy by byl program při každém spuštění opětovně interpretován, jako by tomu bylo například u oblíbeného programovacího jazyka *Python*.

Navržené metody byly implementovány do školního frameworku TMS (Tone Mapping Studio), který se zabývá velkou řadou odlišných operací pro zpracování obrazu a spadá pod kompetence vedoucího práce.

3.1.1 Využité knihovny

Při programování konverzních algoritmů bylo využito několika externích knihoven.

Knihovna OpenCV

OpenCV[1] (Open Source Computer Vision Library) je knihovna pro počítačové zpracování obrazu a strojové učení, která byla vydána pod BSD licencí, a tedy ji lze používat zcela zdarma pro akademické i komerční účely.

Knihovna obsahuje velké množství optimalizovaných algoritmů a zaměřuje se na provádění operací v reálném čase, a i proto se stala po celém světě velmi oblíbenou variantou.

Knihovna Eigen

Eigen[9] představuje alternativu k provádění matematických operací pro programovací jazyk *C++*, především k řešení matematickým problémů v oblasti lineární algebry, což zahrnuje práci se soustavou rovnic, vektory, maticemi a také umožňuje operace nad těmito prvky. Umožňuje také práci s geometrickými objekty, ale na tyto účely nebyla knihovna použita.

Za největší výhodu knihovny Eigen lze považovat využívání přístupu nazývaného jako líné vyhodnocování (lazy evaluation)[7], kde dochází k zjišťování hodnoty výrazu až v případě její akutní nutnosti jinou operací a tento proces vyhodnocování pro proměnnou probíhá pouze jednou, což může v mnoha případech urychlit práci s takovými objekty.

Knihovna MLPack

MLPack[2] je knihovna pro jazyk *C++*, používaná pro účely strojového učení, která se snaží zajistit maximální výkon a flexibilitu operací pro pokročilé uživatele. Také nabízí programy spustitelné v příkazové řádce, které mohou být využívány jako černá skříňka (black box)[1] anebo upraveny zkušenějšími uživateli pro své účely.

3.2 Typy konverzních metod

Všechny metody převodu barevných snímků na stupně šedi se dají rozdělit do dvou kategorií, podle způsobů jejich mapování. Liší se nejen v rychlosti, ale i v kvalitě výstupního řešení. Obě dvě kategorie jsou popsány v následujících podkapitolách.

3.2.1 Globální metody převodu

Jedná se o takové metody[27], kde se celý výsledný obraz odvíjí od úrovně hodnot váhových koeficientů, sloužících pro přepočty jednotlivých barevných kanálů. Tedy stejná barevná hodnota bude vždy namapována na stejný stupeň šedi, nehledě na okolí a umístění pixelu ve snímku. Nejjednodušším a velmi používaným příkladem takového převodu je *rgb2gray*[2], kde se hodnoty váhových koeficientů neodvíjí od podoby vstupního obrazu, ale jsou pevně dány a výpočet výsledné hodnoty jasu pak probíhá ze všech tří barevných kanálů v modelu RGB. Klasický známý vzorec pro tuto operaci:

$$Gray = 0.229 \cdot Red + 0.587 \cdot Green + 0.114 \cdot Blue. \tag{3.1}$$

Za běžného denního světla je lidské oko nejcitlivější na vlnové délky světla odpovídající zelené barvě. Taková barva se pak jeví opticky jako světlejší, a to je důvod proč tyto metody uvažují nejvyšší koeficient právě pro zelený kanál v modelu RGB.

Metoda *rgb2gray* je jednoduchá, rychlá a může být v některých situacích dostatečná, ale má již zmíněnou zásadní slabinu. Pixely s malým rozdílem hodnoty jasu, avšak velkým rozdílem v sytosti barvy přestávají být ve výsledném obrazu ve stupních šedi odlišitelné. Příkladem takových barev je bílá a žlutá anebo tmavě modrá a černá.

Tato práce se věnuje především globálnímu mapování a hledání ideální funkce pro zjištění vhodných výhových koeficinetů. Je velká řada způsobů, jak k takovému získání potřebných informací ze vstupního snímku přistupovat, metody popsané v následujících kapitolách vysvětlí některé z nich.

3.2.2 Lokální metody převodu

Tyto postupy[27] lze považovat za univerzálnější a poskytující přesnější výsledek na úkor vyšší paměťové a časové náročnosti. Lokální metody berou v potaz konkrétní hodnoty vstupního snímku a zpracovávají pixel po pixelu, přičemž takové operace pak vedou na jev, kdy jsou pixely se stejnou barevnou hodnotou mapovány na odlišné stupně šedi. Nevýhodou těchto metod je riziko výskytu nežádoucích artefaktů, jako je šum v obraze.

[1] zařízení, kterému poskytneme vstup a vrátí nám výstup; nezajímá nás jakým způsobem funguje
[2] konverzní funkce používaná v MATLABu

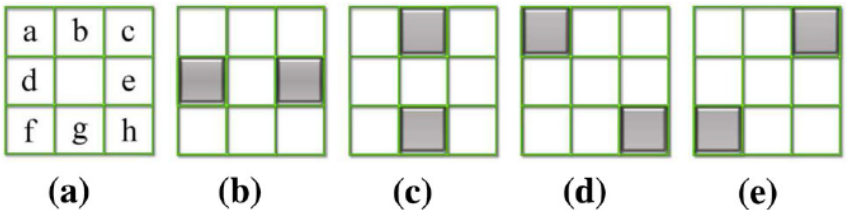
3.3 Efektivní dekolorizace zachovávající významné rozdíly (Efficient decolorization preserving dominant distinctions)

Tato metoda[27] (zkráceně také jako *Zhongping15*) cílí při mapování barevného obrazu na černobílý na zachování převládajícího kontrastu a konzistence jasu. Také pracuje na bázi globálního mapování 3.2.2, a tedy všechny barvy co se shodují budou namapovány přesně na společně totožné stupně šedi. Algoritmus se vypořádává s prvkem šumu pomocí uživatelem volitelného rozostření obrazu.

První krok přísluší převodu snímku do barevného modelu CIELab 2.2.2, který více odpovídá barevnému rozsahu vnímání lidského oka. Poté se z tohoto modelu vypočítá chromatická orientace, která zachycuje informaci o globálním kontrastu. Aby mohla být chromatická orientace zjištěna pro všechny dílčí pixely, tak musí být nejprve v každém obrazovém bodě spočítán gradient pro všechny tři kanály, a tedy pro ∇L, ∇a, ∇b. Chromatickou orientaci řízenou luminancí a nesoucí informaci o gradientech, pak představuje pro každý pixel dvourozměrný vektor o hodnotách zachycujících vertikální a horizontální složku. Výpočet těchto vektorů znázorňuje vzorec:

$$Gp = \sum_{<i,j>\in N_p} s_{ij}\mathfrak{g}(\Delta L_{ij})(\Delta a_{ij}, \Delta b_{ij}), \tag{3.2}$$

kde N_p značí množinu dvojic sousedních pixelů, která je ilustrována na obrázku 3.1. Pořadí bodů v těchto dvojicích není zaměnitelné a první prvek dvojice vždy připadá bodu reprezentovanému podle abecedního řazení.

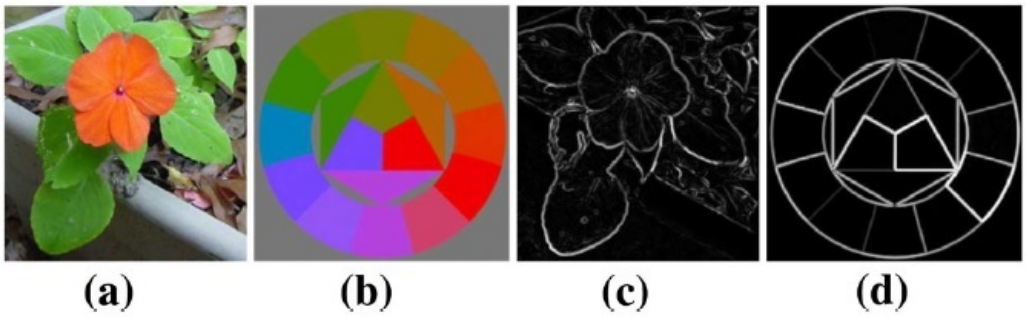


Obrázek 3.1: a) Pojmenovaná sada pixelů značená N_p; b) dvojice pixelů <d, e>; c) dvojice pixelů <b, g>; d) dvojice pixelů <a, h>; e) dvojice pixelů <c, f>; Převzato z [27].

Proměnná $\mathfrak{g}(x)$ reprezentuje váhovou funkci $1 + x^2$ pro luminanční kanál, který svou významovou váhou řídí celou tuto operaci a váhovou funkci poté rozšiřuje znaménková konstanta s_{ij}:

$$s_{ij} = \begin{cases} 1 & \text{pokud } \Delta l \geq 0 \\ -1 & \text{jinak} \end{cases}$$

Chromatická orientace vlastně udržuje informace o skokových změnách v intenzitách jasu a tedy zastupuje detekci hranic pro regiony, jak je ukázáno na následujícím obrázku 3.2.

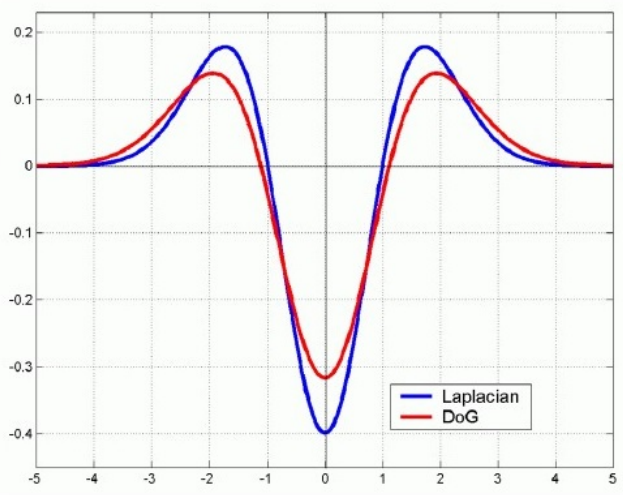


Obrázek 3.2: a), b) originální obrázky; c) d) detekované hrany reprezentované vypočtenou chromatickou orientací; Převzato z [27].

Globální chromatická orientace je zjištěna ze všech dílčích chromatických orientací pro jednotlivé pixely pomocí vzorce (3.3) a je uváděna jako jediný vektor udávající dominantní směr vertikální osy i horizontální osy pro celý obraz.

$$(O_a, O_b) = \sum_p Gp / |\sum_p Gp| \tag{3.3}$$

Pro získání snímku ve stupních šedi je definován luminanční filtr za pomoci rozdílu Gaussianů 2.8. Tento rozdíl tradičně slouží k zvýšení viditelnosti hran a detailů v obraze, avšak zde simuluje podobný filtr a to Laplace z Gaussianů 2.6, jejichž vzájemnou podobnost znázorňuje obrázek 3.3.



Obrázek 3.3: Rozdíl mezi LoG a Dog filtrem; Převzato z [14].

Luminanční filtr zachytí změny v obraze pro kanál jasu, a to v dané šířce jeho plochy, která souvisí s parametry σ_1 a σ_2 udávající velikosti Gaussianských jader. V rovnici (3.4) se skrývá nejen odstranění barev ze vstupního obrazu, ale rovněž možnost vyhlazení šumu.

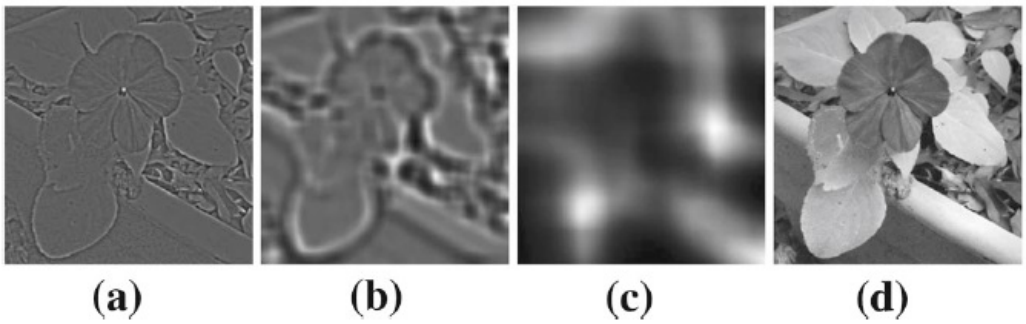
$$\mathscr{L}_{\sigma_1,\sigma_2}(x,y) \otimes I(x,y) = |(\mathscr{G}_{\sigma_1}(x,y) - \mathscr{G}_{\sigma_2}(x,y)) \otimes I(x,y)|_O, \tag{3.4}$$

kde $\otimes$ značí matematickou operaci konvoluce, která slučuje dvě odlišné funkce a vytváří z nich funkci třetí na základě jejich výstupních rysů.

Symboly $|\cdot|_O$ znázorňují směrovou vzdálenost a lze ji určit pomocí vzorce:

$$|c_1 - c_2|_O = \Delta l + O_a \Delta a + O_b \Delta b. \tag{3.5}$$

Jakých výsledků při různých nastaveních rozostření Gaussianských jáder dosahuje výše zmíněný luminanční filtr lze pro lepší představu vidět na obrázcích 3.4.



Obrázek 3.4: Použití dekolorizačního vzorce (3.4) s parametry: a) $\sigma_1 = 0, \sigma_2 = 5$; b) $\sigma_1 = 10$, $\sigma_2 = 20$; c) $\sigma_1 = 50$, $\sigma_2 = 100$; d) $\sigma_1 = 0$, $\sigma_2 = 100$; Převzato z [27].

Finální převod zajišťuje vzorec (3.6), ve kterém pro zachycení celého gamutu[3] normalizujeme dílčí kroky výpočtu do rozsahu nula až jedna, čímž dojde k odstranění záporných hodnot.

$$\hat{l}(x,y) = l_m + |\mathscr{G}_{\sigma_1}(x,y) \otimes I(x,y) - I_m|_O, \tag{3.6}$$

kde do vzorce vstupují proměnné $I_m = \mathscr{G}_\infty(x,y) \otimes I(x,y)$, která značí průměrnou barevnou hodnotu pro celý vstupní snímek a také $l_m = \mathscr{G}_\infty(x,y) \otimes l(x,y)$ udávající průměrnou úroveň pro luminanční kanál.

Nastavení parametru sigma

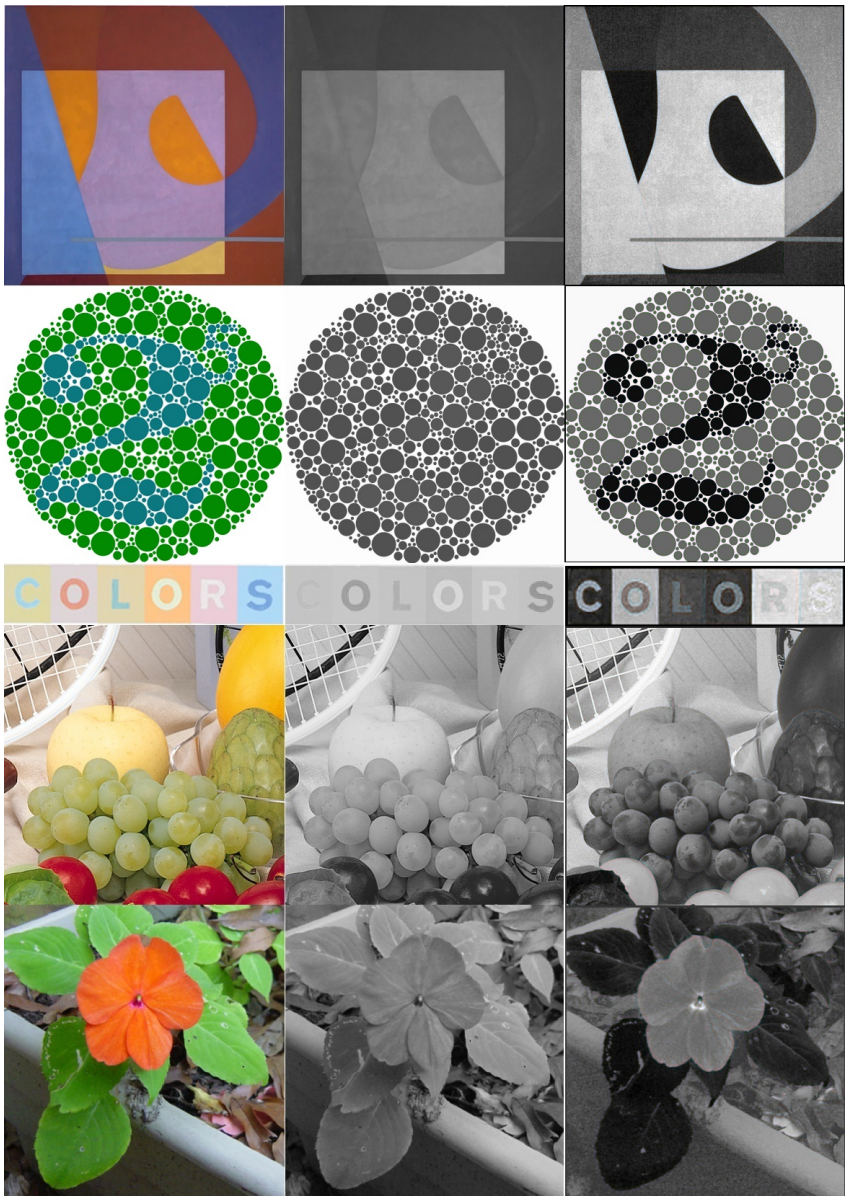
Autoři práce[27], z které je čerpáno, udávají jako výchozí hodnotu pro σ_2 – nekonečno, jelikož algoritmus pak dosahuje nejlepších výsledků. To neplatí pro σ_1, u té je pro jednoduchost výpočtu volena malá hodnota, často rovna nule, kde pak $\mathscr{G}_{\sigma_1}(x,y) \otimes I(x,y)$ odpovídá pouze vstupnímu obrazu $I(x,y)$. Takovou volbou zajišťujeme globální mapování dekolorizace a zároveň využití celého pásma jasu. V případě výskytu šumu v obraze je doporučen výběr uživatelsky volitené vyšší číselné konstanty pro velikost jádra σ_1, to poté vede k rozostření obrazu a eliminaci tohoto parazitního jevu.

[3] Barevná oblast, které je možné teoreticky dosáhnout

3.3.1 Dosažené výsledky

Algoritmus[27] je značně rychlý 4.2 a měl by umožňovat převod na stupně šedi téměř v reálném čase, a proto může být vhodným kandidátem pro využití při dekolorizaci videosekvencí. Nicméně stejně jako u mnoha jiných algoritmů se ukázalo, že tato metoda není naprosto univerzální a pro některé případy vstupních snímků není výsledek dostatečně vyhovující viz příloha A, přesto však poskytuje dobrý kompromis mezi kvalitou a rychlostí.

Obrázek 3.5 znázorňuje v pravém sloupci, kterému by měla být věnována hlavní pozornost, dosažené výsledky dokončené implementované metody.



Obrázek 3.5: Ukázka výsledných převodů na obrázkovém datasetu doc. Čadíka[28]; vlevo původní snímek, uprostřed *rgb2gray* viz vzorec 3.1, vpravo algoritmus *Zhongping15* 3.3

3.4 Parametrická poměrově založená metoda pro efektivní zachování kontrastu (Parametric ratio-based method for efficient contrast-preserving decolorization)

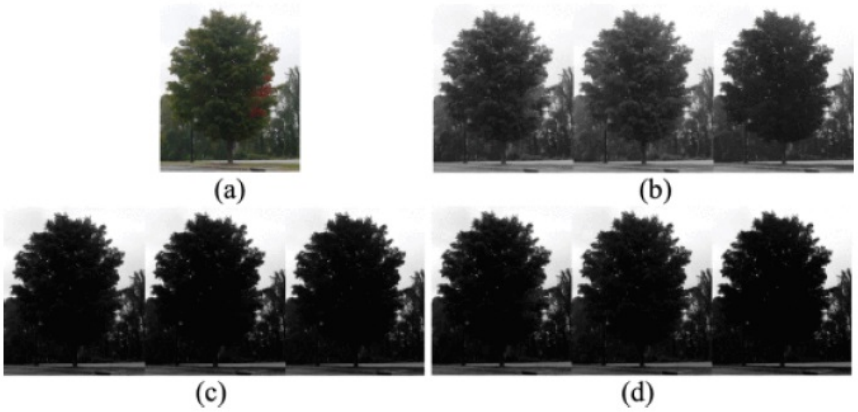
Jedná se o iterativní globální metodu[17] (zkráceně nazáváno také *Xiong17*) 3.2.2, obsahující váhové koeficienty pro RGB kanály, podobné jako je tomu u lokálních metod 3.2.1, které jsou zde však iterativně přepočítávány ze vstupního snímku, přičemž postupně konvergují k ideálnímu řešení.

$$g = f(I_r, I_g, I_b; W) = \sum_{I_l \in Z_2} W_l I_l, \tag{3.7}$$

kde RGB vektor $c = (I_r, I_g, I_b)$; prvky vektoru c tedy značí jednotlivé RGB kanály.

$$Z_2 = \{I_r, I_g, I_b, I_{rg}, I_{rb}, I_{gb}, I_{r^2}, I_{g^2}, I_{b^2}\} \tag{3.8}$$

Vzorec (3.7) popisuje finální převod barevných kanálů na stupně šedi. Je zde možné vidět, že váhové koeficienty W_l se mapují izomorfním vztahem na prvky z množiny Z_N. Právě množinou Z_N je tato metoda odlišná od většiny ostatních algoritmů. Jedná se o multivariantní polynomiální model, který obsahuje základní RGB kanály, zde jako I_r, I_g, I_b a podle úrovně modelu i jejich vzájemnou lineární kombinaci. Autoři vzorové práce[17] experimentálně zjistili, že nejvhodnější řád pro N se rovná hodnotě dva. Množina Z_2 (3.8) pak velmi dobře popisuje matematický pojem lineární obal. V takové variantě máme devět prvků, které se, jak již bylo řečeno, izomorfně mapují na devět váhových koeficientů. Každý vyšší řád rozšiřuje počet nutných obsažených prvků v množině, a tudíž také časovou a paměťovou náročnost. Méně zřejmé pak ale může být to, že s vyšším řádem dochází ke ztmavování výstupního obrazu. Tato skutečnost vyplývá z faktu, že s barvami je manipulováno v normalizované podobě v intervalu nula až jedna. Různé úrovně barevných řádů lze vidět na obrázku 3.6.



Obrázek 3.6: Vliv Z_N úrovně na výsledný snímek; a) původní obraz, b) I_r, I_g, I_b, c) I_{rg}, I_{rb}, I_{gb}, d) $I_{r^2}, I_{g^2}, I_{b^2}$; Převzato z [25].

Jádro problematiky spočívá ve výpočtu zmíněných váhových koeficientů a toho lze dosáhnout pomocí zde nejdůležitějšího vzorce:

$$W^{k+1} = \underset{W}{argmin} \frac{\gamma}{2} \left\| \sum_{I_l \in Z_2} W_l^k \nabla I_l - \left(S_{x,y}^k - \frac{\lambda^k}{\gamma} \right) \right\|_2^2 . \tag{3.9}$$

Gradient ∇ 2.3.2 počítaný pro obraz znázorňuje změnu intenzity hodnoty pixelu vůči sousednímu pixelu v daném směru. Mezi tyto směry patří horizontální, pak počítáme gradient pro osu X a vertikální směr pak reprezentuje gradient pro osu Y.

Speciálním případem jsou pixely na posledním řádku či sloupci obrázku, které lze řešit mnoha způsoby. Přístup k těmto okrajovým pixelům v metodě *Xiong17* se shoduje se s přístupem autorů algoritmu; okrajové body obrazu pro X složku gradientu se rovnají nule, v složce Y je výjimečným případem pouze poslední řádek, který kopíruje hodnoty vstupního obrazu.

Nejpodstatnější částí vzorce je $\underset{W}{argmin}$, který znázorňuje snahu zjistit minimum, kterého může při vstupních parametrech váhový koeficient nabývat. Pokud se jedná o klasický druhý polynomiální řád množiny Z_N, pak je toto minimum hledáno v devítirozměrném prostoru. Na to bylo využito minimalizační funkce *ColPivHouseholderQr* z matematické knihovny *Eigen* 3.1.1.

Součástí předchozího vzorce je i S_{xy} (3.10), jedná se pouze o pomocnou proměnnou rozkládající předchozí výpočet na podproblém.

$$s_{x,y}^{k+1} = \frac{\gamma \sum_{I_l \in Z_2} W_l^{k+1} \nabla I_l + \lambda^k}{2 + \gamma - 2Kt \sum_{(x,y) \in P} |\delta_{x,y}|} \tag{3.10}$$

kde je $\delta_{x,y}$ odpovídající vzdálenosti mezi dvěmi body x a y v CIELab modelu 2.2.2 počítána jako:

$$|\delta_{x,y}| = \sqrt{(L_x - L_y)^2 + (a_x - a_y)^2 + (b_x - b_y)^2}. \tag{3.11}$$

Posledním z trojice hlavních vzorců je (3.12), kde je počítán Lagrangeův násobitel[5]:

$$\lambda^{k+1} = \lambda^k + \gamma \left(\sum_{I_l \in Z_2} W_l^{k+1} \nabla I_l - s_{x,y}^{k+1} \right) . \tag{3.12}$$

Výpis (3.4) znázorňuje pseudokód, který názorně popisuje celý výpočet váhových koeficientů, tyto kroky probíhají v několika iteracích a získané hodnoty jsou poté aplikovány na vstupní barevný obrázek.

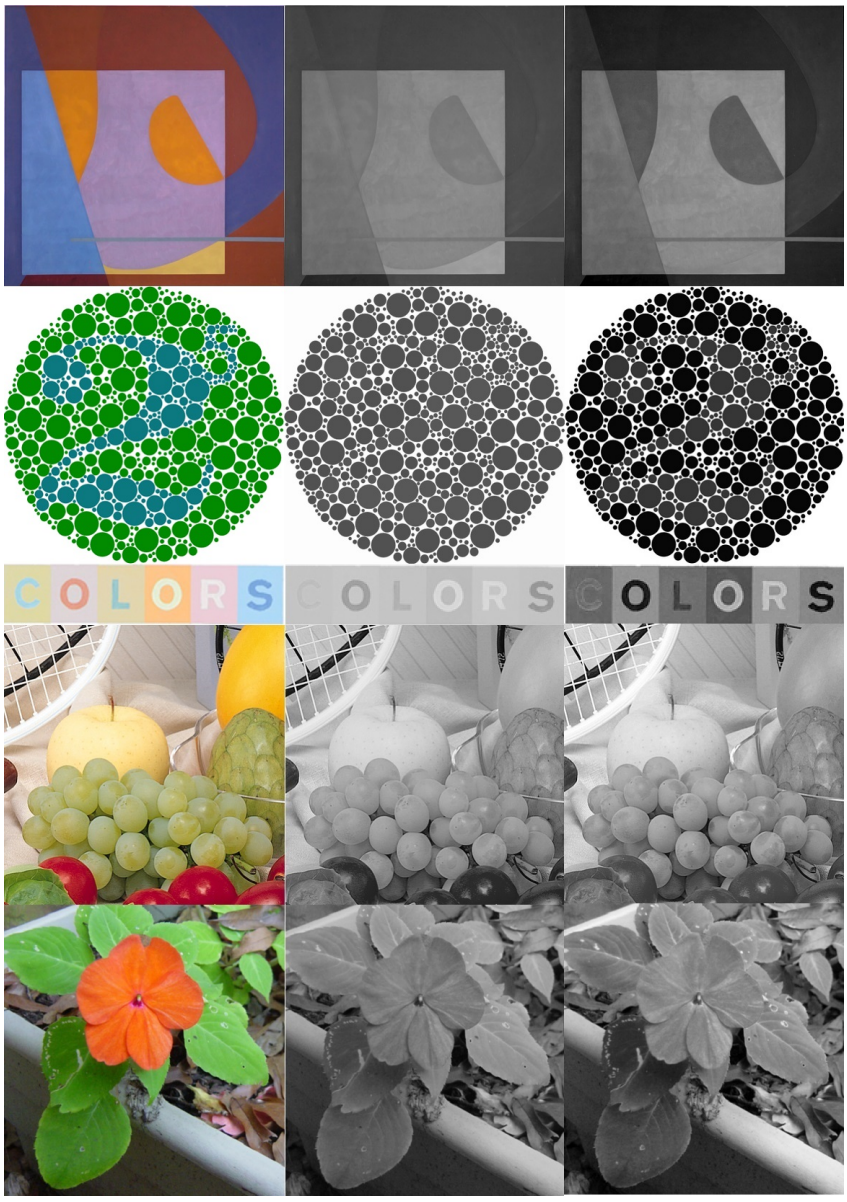
```
1: vstupni promenne: barevny obrazek, I, μ
2: inicializace: W^0; s^0 = λ^0 = 0
3: for (unsigned int k = 0; k < maxIteraci; ++k)
   {
4:  vypocet W^{k+1};

5:  vypocet s_{x,y}^{k+1};

6:  vypocet λ^{k+1};
   }
```

Ukázka kódu (3.4.) – Pseudokód metody *Xiong17*[17]

3.4.1 Dosažené výsledky

Metoda *Xiong17* [17] poskytuje velmi kvalitní výstupní obraz, včetně dobré úrovně zachyceného kontrastu a to především pro snímky s velkým množstvím odlišných barev a obrazových vzorů. Tento algoritmus byl vyhodnocen podle kvality převodu za nejuspokojivější z trojice implementovaných metod a také předčil v porovnání kvality převodu mnoho dalších dekolorizačních metod.

Přesto však konverzní metoda *Xiong17* nemusí být správnou volbou v oblasti dekolorizace videosekvencí, jelikož pro tyto účely není dostatečně výpočetně rychlá 4.2 a je tedy vhodné ji zvolit především tehdy, pokud je kvalita (viz příloha A) požadovaného výsledku nekompromisně důležitým faktorem.



Obrázek 3.7: Ukázka výsledných převodů na obrázkovém datasetu doc. Čadíka[28]; vlevo původní snímek, uprostřed *rgb2gray* viz vzorec 3.1, vpravo *Xiong17* 3.4

3.5 Zachování vnímaného kontrastu v dekolorizaci pomocí optimalizovaných barevných řad (Preserving perceptual contrast in decolorization with optimized color orders)

V předchozích metodách dochází k hledání ideální funkce zajišťující nejnižší možnou ztrátovost kontrastu mezi barevným snímkem a jeho reprezentací ve stupních šedi. Zde[18] (dále také jako metoda *Jin17*) je využíváno jiné metodiky – funkce na ztrátovost kontrastu slouží jako jádro pro hledání ideálních vstupních hodnot zastupující váhové koeficienty. Rovněž na získání kontrastu pro stupně šedi používáme nelineární funkci tak, aby výsledek lépe odpovídal vnímání kontrastu lidským okem. Tyto vylepšení rozšiřují princip oblíbené dekolorizační metody pro zachování kontrastu[13], implementované v OpenCV 3.1.1. Metoda, kterou se tato kapitola zabývá rovněž patří mezi globální 3.2.2.

Z algoritmu[13] se převezme polynomiální funkce (3.13), pomocí které se na závěr algoritmu převede barevný snímek na smínek ve stupních šedi.

$$g = f(v_r, v_g, v_b; \omega) = \sum_i \omega_i m_i, \tag{3.13}$$

kde $\Omega = \{v_r, v_g, v_b, v_{rg}, v_{rb}, v_{gb}, v_{r^2}, v_{g^2}, v_{b^2}\}$ udává parametr pro mapovací funkci a m_i představuje i-tý prvek v množině Ω.

Cílem algoritmu je získání vhodných hodnot pro devítiprvkovou množinu ω s počátečními hodnotami $\{0.33, 0.33, 0.33, 0, 0, 0, 0, 0, 0\}$, které jak je možno vidět, přikládají pro první iteraci metody stejnou váhu základním třem RGB kanálům, kdežto jejich kombinaci pak váhu nulovou.

Pro zjištění váhových koeficientů je nejprve nutno počítat důležitou funkci pro ztrátovost kontrastu mezi vstupním barevným obrazem a obrazem ve stupních šedi:

$$E = \sum_{(x,y)\in P} (\delta^g_{(x,y)} - \delta^c_{(x,y)})^2, \tag{3.14}$$

kde x a y značí dva rozdílné pixely, δ^g (3.16) je kontrast mezi pixely x a y pro obraz ve stupních šedi, δ^c (3.15) potom znázorňuje kontrast mezi pixely x a y pro snímek barevný.

$$\delta^c_{(x,y)} = \sqrt{(L_x - L_y)^2 + (a_x - a_y)^2 + (b_x - b_y)^2}, \tag{3.15}$$

kde L, a, b značí kanály CIELab modelu 2.2.2, s kterým metoda pracuje.

$$\delta^g_{(x,y)} = s^g_{(x,y)}(g_x - g_y), \tag{3.16}$$

přičemž znaménková funkce $S^g_{(x,y)}$ je definována jako:

$$s^g_{(x,y)} = \begin{cases} 1 & \text{pokud } r_x \geq r_y, g_x \geq g_y, b_x \geq b_y \\ -1 & \text{pokud } r_x < r_y, g_x < g_y, b_x < b_y \\ |\cdot| & \text{jinak} \end{cases}$$

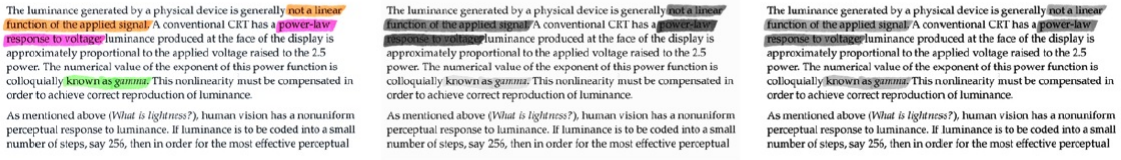
Nelineární vnímání kontrastu

Weberův-Fechnerův zákon[10] poukazuje na fakt, že vnímání aktuální změny v porovnání se změnou skutečnou je reprezentováno logaritmickou křivkou. Malý přírustek intenzity jasu, především ve snímcích s vysokými průměrnými hodnotami luminance, totiž nemůsí být lidským okem viditelný a obraz se může jevit zcela beze změny.

S cílem lepšího vnímání kontrastu lidským okem byla přidána nelinearita obsažená v logaritmických operacích. Avšak jelikož CIELab model 2.2.2 už takovou situaci zajišťuje, byla nelineární funkce aplikována pouze na snímek ve stupních šedi.

Aplikace takového pravidla pak vede na úpravu původní rovnice (3.14):

$$E = \sum_{(x,y)\in P} \left(s^g_{(x,y)} \left(\frac{1}{\log(1+\beta)} \cdot \frac{\beta}{\beta g_{c_i} + 1} \cdot (g_{c_i} - g_{c_j}) \right) - \delta^g_{(x,y)} \right)^2. \tag{3.17}$$



Obrázek 3.8: Zavedení nelinearity do stupňů šedi vedoucí k lepším vizuálním výsledkům; vlevo originální barevný snímek; uprostřed linearita; vpravo nelinearita; Převzato z [18].

Výpočet takové operace by byl časově náročný a jelikož má mnoho pixelů v barevných regionech stejnou barvu, nabízí se operace kvantizace barev ve snímku, vedoucí ke snížení barevné palety nebo jinak řečeno výrazná redukce celkového počtu využitých barev v obraze, což vede k značnému urychlení výpočtu ztrátového kontrastu. Pro tyto účely je využita metoda kvantizace barev využívající principy řízeného strojového učení 2.4 s názvem K-means 2.3.1. Vhodnou volbou počtu clusterů je podle autorů algoritmu hodnota 30.

Takto upravená rovnice je pak zapsána jako:

$$E = \sum_{(c_i,c_j)\in C} K(c_i,c_j) \cdot \left(s^g_{(c_i,c_j)} \left(\frac{1}{\log(1+\beta)} \cdot \frac{\beta}{\beta g_{c_i} + 1} \cdot (g_{c_i} - g_{c_j}) \right) - \sqrt{(L_{c_i} - L_{c_j})^2 + (a_{c_i} - a_{c_j})^2 + (b_{c_i} - b_{c_j})^2} \right)^2, \tag{3.18}$$

kde C reprezentuje množinu všech clusterů, váha clusterového páru $K(C_i, C_j)$ se spočítá pomocí:

$$K(c_i,c_j) = \frac{N_{c_i} N_{c_j}}{N_o^2}, \tag{3.19}$$

kde N_{c_i} resp. N_{c_j} značí počet pixelů v clusterech C_i resp. C_j; N_o je potom definováno jako $0,01 \cdot Z$, přičemž Z reprezentuje celkový počet pixelů.

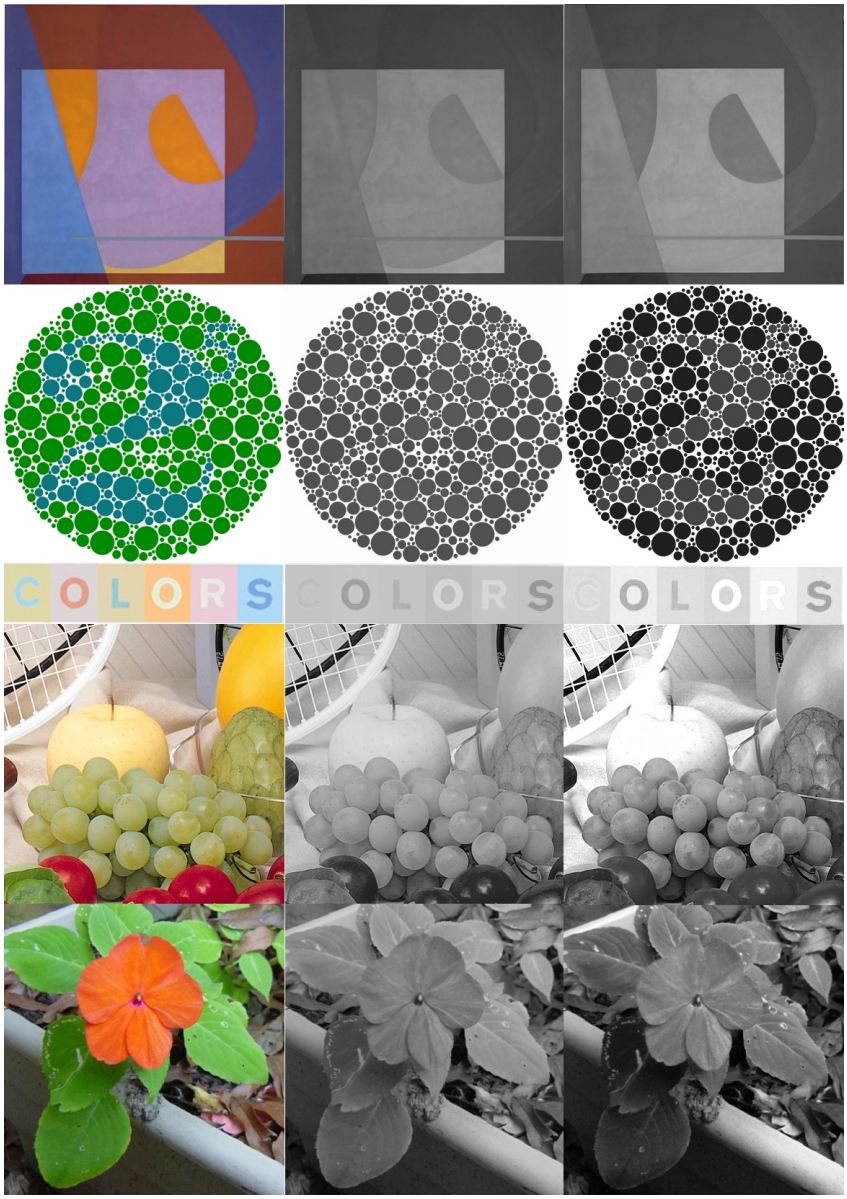
Závěrečný převod barevného snímku na stupně šedi pak probíhá v několika krocích. Nejprve pomocí váhových koeficientů Ω spočítáme vzorcem (3.13) obraz ve stupních šedi, který následně vstupuje do výpočtu ztrátovosti kontrastu (3.18). Poté jsou změněny hodnoty pro množinu Ω, kde je hledána taková kombinace koeficientů, pro kterou dojde k výskytu nejnižších ztrát v kontrastu a celý proces výpočtu opakujeme. Takto časově a výpočetně náročný úkol využívá optimalizačních technik popsaných v následující podkapitole.

3.5.1 Optimalizace

Jelikož znaménková funkce z rovnice (3.5) obsahuje i absolutní hodnotu, funkce pro výpočet ztrátového kontrastu je nekonvexní, tedy neobsahuje jediné lokální minimum a z toho důvodu je výrazně těžší najít globální minimum. Pro řešení takových optimalizačních úloh je využito algoritmu pro strojové učení 2.4 ze skupiny metod pro sestup gradientů 2.5, například Odhad momentu adaptace (Adam) 2.5.1. Autoři práce uvádí[18], že je vhodné nastavit rychlost učení na 0.0005, parametr Beta na hodnotu 0.5 a provést minimálně tisíc iterací. Tento algoritmus zjišťuje, pro jakou vstupní množinu Ω získáme nejnižší hodnotu z funkce ztrátovosti kontrastu, tedy hledáme globální minimum v 9D prostoru. A právě takovou množinu Ω poté aplikujeme v dekolorizačním vzorci (3.13).

3.5.2 Dosažené výsledky

Tato metoda[18] je založená na opačném postupu hledání váhových koeficientů a poskytuje uspokojivé výsledky viz příloha A, které je možné vidět v pravém sloupci na obrázku 3.9. V kapitole 4.4 bude vysvětleno, proč je při obecném použití dekolorizace videosekvencí nejlepší volbou z trojice implementovaných metod.



Obrázek 3.9: Ukázka výsledných převodů na obrázkovém datasetu doc. Čadíka[28]; vlevo původní snímek, uprostřed *rgb2gray* viz vzorec 3.1, vpravo algoritmus *Jin17* 3.5

Kapitola 4

Porovnání implementovaných metod

Součástí práce je porovnání implementovaných metod, a právě tím se podrobně zabývá tato kapitola. Pozornost je věnována především parametrům výpočetní rychlosti, paměťové náročnosti a tyto výstupní statistiky jsou prezentovány formou grafů. V závěru následuje krátké zhodnocení a doporučená metoda nejen pro převod jednoho barevného snímku na stupně šedi, ale také popis využitelnosti dekolorizačních metod v oblasti videosekvencí. Převod barevných videí na stupně šedi je prováděn snímek po snímku, a přestože se nepředpokládá proces dekolorizace v reálném čase, musí být značná pozornost věnována rychlosti takové konverze.

4.1 Implementační náročnost metod

Každý z implementovaných algoritmů byl jistým způsobem výzvou. Za nejjednodušší z této trojice lze považovat *Efektivní dekolorizace zachovávající významné rozdíly (Zhongping15)* 3.3, kde se vyskytují především známé operace v oblasti zpracování obrazu, mezi které patří například detekce hran nebo rozostření obrazu.

Náročné se však ukázaly býti především zbylé dvě implementované metody, jelikož předpokládají nutnost využití optimalizačních algoritmů. Operace optimalizace zvyšuje efektivitu nebo cílí na úsporu využitých zdrojů. Jedná se však o nejednoduchý proces, který předpokládá značné předchozí znalosti v této oblasti.

Za další obtíž zvyšující náročnost implementace metody *Parametrická poměrově založená metoda pro efektivní zachování kontrastu (Xiong17)* 3.4 je možné považovat ve výchozí studii nedostatečné množství mezikrokových ilustračních obrázků sloužících ke kontrole správnosti.

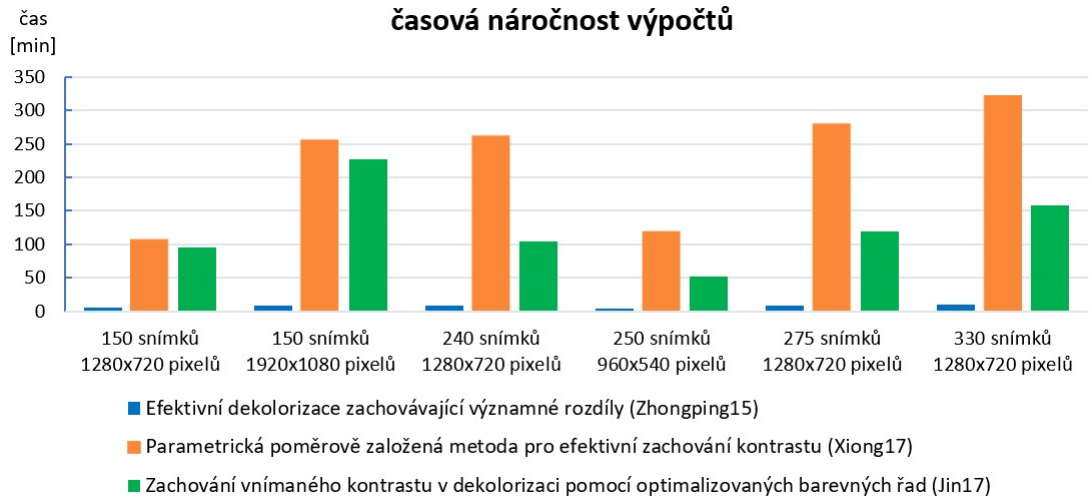
4.2 Časová náročnost metod

Testy časové náročnosti tří naimplementovaných metod proběhly s použitím výchozích hodnot vstupních parametrů a na stroji s procesorem 4x Intel(R) Core(TM) i7-5500U CPU @2.40 GHz, s pamětí 8GB RAM a grafickou kartou Intel(R) HD Graphics 5500 v prostředí operačního systému Linux Ubuntu 16.04 Xenial 64bitové verze.

Experiment byl prováděn na sérii videosekvencí s odlišným celkovým počtem obsažených snímků a rovněž také s rozdílným rozlišením obrazu. Tuto skutečnost a výsledky měření zachycuje graf 4.1 poukazující na výrazné rozdíly v rychlosti dekolorizačních metod.

Přestože algoritmus *Efektivní dekolorizace zachovávající významné rozdíly (Zhongping15)* 3.3 nenabízí výsledky, které by se kvalitou mohly rovnat dvěma konkurenčním metodám, má v oblasti videosekvencí co nabídnout.

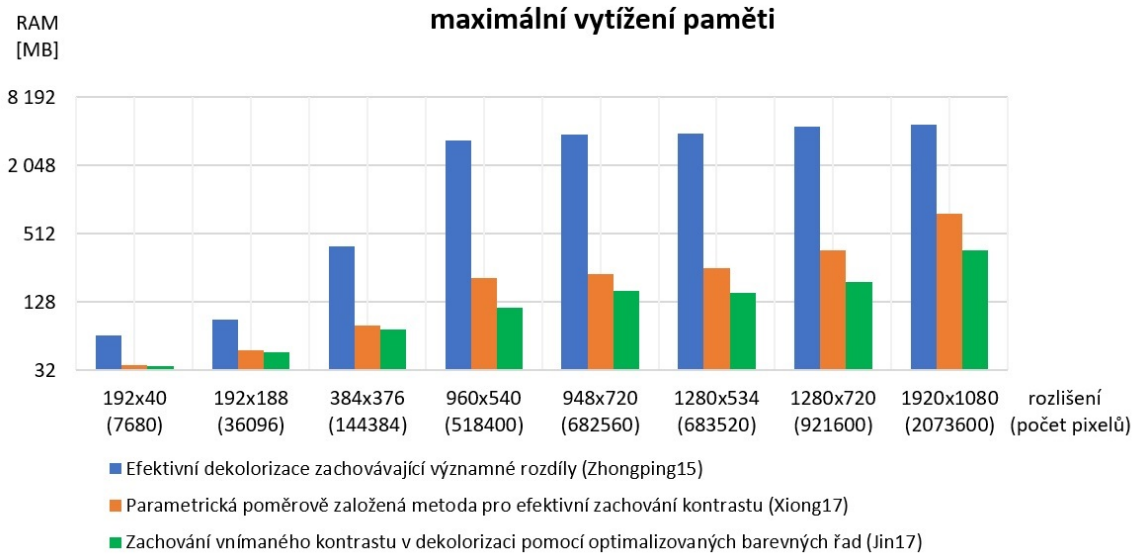
Zajímavou skutečností stojící za povšimnutí může být fakt, že metoda *Zachování vnímaného kontrastu v dekolorizaci pomocí optimalizovaných barevných řad (Jin17)* 3.5 poskytuje nad očekávání rychlé výsledky pro snímky s nižším počtem pixelů.



Obrázek 4.1: Graf zachycující porovnání časové náročnosti výpočtů pro jednotlivé metody

4.3 Paměťová náročnost metod

Tato podkapitola popisuje, jakých hodnot maximálního vytížení paměti RAM dosahovaly implementované metody při zvoleném rozlišení obrazu. Vertikální osa grafu 4.2 udávající paměťovou spotřebu je reprezentována logaritmicky a je tedy možné si povšimnout, že algoritmus *Efektivní dekolorizace zachovávající významné rozdíly (Zhongping15)* 3.3 poskytuje velmi neuspokojivé výsledky. I přestože se jedná o rychlou metodu 4.1, a tedy toto vysoké vytížení není dlouhodobou záležitostí, může znamenat podstatný problém při použití na méně výkonných strojích.



Obrázek 4.2: Graf zachycující porovnání časové náročnosti výpočtů pro jednotlivé metody

4.4 Hodnocení použitelnosti metod

V této sekci je zohledněna výsledná kvalita převodu viz příloha A, rychlost i paměťová náročnost implementovaných metod, kde vycházíme ze zjištěných hodnot zachycených grafy v předchozích podkapitolách.

Algoritmus *Efektivní dekolorizace zachovávající významné rozdíly (Zhongping15)* 3.3 je rychlý, avšak paměťově velmi náročný a navíc neposkytuje dostatečně dobré výsledky viz příloha A. Doporučenou volbou je pouze v situacích, kdy je kladen opravdu velký důraz na rychlost.

Nejlepší z této trojice, zlatý střed v poměru kvality a rychlosti představuje metoda *Zachování vnímaného kontrastu v dekolorizaci pomocí optimalizovaných barevných řad (Jin17)* 3.5. Tato metoda poskytuje uspokojivé výsledky a nízké maximální paměťové vytížení. Zajímavým faktorem je zde rychlost, na obrázku 4.1 si lze povšimnout, že při nižším celkovém počtu pixelů je metoda výrazně rychlejší než v opačném případě, například při Full HD rozlišení. V situaci práce se zmíněným Full HD rozlišením se již více vyplatí využití metody *Parametrická poměrově založená metoda pro efektivní zachování kontrastu (Xiong17)* 3.4.

Kapitola 5

Závěr

Tato bakalářská práce se zabývá převodem barevných snímků na stupně šedi, což je stále aktuální a přínosné téma v oblasti zpracování obrazu. Cílem práce byla implementace tří algoritmů sloužících právě pro tyto účely a následné porovnání a posouzení jejich použitelnosti v praxi pro dekolorizaci videosekvencí.

V úvodních částech práce je pozornost věnována teoretickému rozboru reprezentace digitálního obrazu a využitých operací pro jeho zpracování. Mezi které patří například kvantizace a rozostření obrazu, detekce hran a regionů, ale i optimalizační algoritmus Adam Gradient Descent spadající do kategorie strojového učení.

V prostřední části práce jsou velmi zblízka přiblíženy implementované metody, principy na kterých pracují doprovází vysvětlivky a matematické vzorce. Pro každý uvedený algoritmus jsou zde prezentovány dosažené výsledky formou několika snímků ze vstupního datasetu, který je celý dostupný až v sekci příloh.

Porovnání metod je podrobně zachyceno v předchozí kapitole, kde je důraz kladen nejen na kvalitu výsledné konverze, ale také na časovou a paměťovou náročnost. Závěrem této kapitoly jsou potom doporučené metody pro různé situace užití.

Slabým místem práce a případnou možností pro zdokonalení je metoda *Efektivní dekolorizace zachovávající významné rozdíly (Zhongping15)*, kde nedochází k úplnému odstranění barevných tónů z regionů objektů, jejichž detekce je součástí použitého interního postupu dekolorizace.

Literatura

[1] *About library OpenCV*. [Online; navštíveno 08.05.2018].
URL https://opencv.org/about.html

[2] *About machine learning library MLPack*. [Online; navštíveno 08.05.2018].
URL http://www.mlpack.org/about.html

[3] *Gaussian blur filter*. [Online; navštíveno 02.05.2018].
URL https://en.wikipedia.org/wiki/Gaussian_blur#/media/File:Cappadocia_Gaussian_Blur.svg

[4] *Image of 3D RGB cube*. [Online; navštíveno 20.04.2018].
URL https://commons.wikimedia.org/wiki/File:RGB_Cube_Show_lowgamma_cutout_a.png

[5] *Khan academy: Langrange multipliers, examples*. [Online; navštíveno 11.05.2018].
URL https://www.khanacademy.org/math/multivariable-calculus/applications-of-multivariable-derivatives/constrained-optimization/a/lagrange-multipliers-examples

[6] *Lab a CIELab color space*. [Online; navštíveno 20.04.2018].
URL https://ipfs.io/ipfs/QmXoypizjW3WknFiJnKLwHCnL72vedxjQkDDP1mXWo6uco/wiki/Lab_color_space.html

[7] *Lazy evaluation*. [Online; navštíveno 08.05.2018].
URL https://wiki.haskell.org/Lazy_evaluation

[8] *Low-pass filtering*. [Online; navštíveno 30.04.2018].
URL https://diffractionlimited.com/help/maximdl/Low-Pass_Filtering.htm

[9] *Mathematic library Eigen*. [Online; navštíveno 08.05.2018].
URL https://www.quantstart.com/articles/Eigen-Library-for-Matrix-Algebra-in-C

[10] *Weber-Fechner law*. [Online; navštíveno 09.05.2018].
URL http://www.cns.nyu.edu/~msl/courses/0044/handouts/Weber.pdf

[11] A. M. Saif, J.; H. Hammad, M.; A. A. Alqubati, I.: *Gradient based image edge detection*. [Online; navštíveno 26.04.2018].
URL https://pdfs.semanticscholar.org/10e3/65efcd0ed5dbdc2e1a38909ea68b07146b06.pdf

[12] Brownlee, J.: *Adaptive Moment Estimation*. [Online; navštíveno 02.05.2018].
URL https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

[13] Cewu, L.; Li, X.; Jiaya, J.: *Decolorization methon in OpenCV*. [Online; navštíveno 05.05.2018].
URL https://docs.opencv.org/3.0-beta/modules/photo/doc/decolor.html

[14] Fei-Fei, L.: *Quick Reviews on Descriptors and Detectors*. [Online; navštíveno 28.04.2018].
URL https://sensblogs.wordpress.com/2011/08/22/quick-reviews-on-descriptors-and-detectors-by-fei-fei-li/

[15] Girod, B.: *Edge detection types*. [Online; navštíveno 26.04.2018].
URL https://web.stanford.edu/class/ee368/Handouts/Lectures/2014_Spring/Combined_Slides/11-Edge-Detection-Combined.pdf

[16] Jacobs, D.: *Image gradients*. [Online; navštíveno 12.03.2018].
URL http://www.cs.umd.edu/~djacobs/CMSC426/ImageGradients.pdf

[17] Jiaojiao, X.; Hongyang, L.; Qiegen, L.; aj.: *Parametric ratio-based method for efficient contrast-preserving decolorization*. [Online; navštíveno 11.05.2018].
URL https://link.springer.com/article/10.1007/s11042-017-5143-6

[18] Jin, B.; Süsstrunk, S.: *Preserving perceptual contrast in decolorization with optimized color orders*. [Online; navštíveno 11.05.2018].
URL https://infoscience.epfl.ch/record/231738/files/ICIP-2017-Bin.pdf

[19] van Loon, R.: *Machine learning*. [Online; navštíveno 03.05.2018].
URL http://bigdata-madesimple.com/machine-learning-explained-understanding-supervised-unsupervised-and-reinforcement-learning/

[20] Muscato, C.: *Opponent process theory of color vision*. [Online; navštíveno 20.04.2018].
URL https://study.com/academy/lesson/opponent-process-theory-of-color-vision.html

[21] Navrátil, P.: *Počítačová grafika a multimédia*. Computer Media, 2007, ISBN 978-80-86686-77-6.

[22] Novák, M.: Pokročilé metody detekce hran v obraze. 2012, [Online; navštíveno 06.05.2018].
URL https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=118046

[23] Ozturk, C.; Hancer, E.; Karaboga, D.: *Color Image Quantization*. [Online; navštíveno 24.04.2018].
URL https://www.mii.lt/Informatica/pdf/INFO1023.pdf

[24] Piech, C.: *K-means method*. [Online; navštíveno 24.04.2018].
URL http://stanford.edu/~cpiech/cs221/handouts/kmeans.html

[25] Qiegen, L.; Peter Xiaoping, L.; Yuhao, W.; aj.: *Semiparametrická decolorizace s vnímáním metrické kvality založené na Laplacianu*. [Online; navštíveno 05.05.2018].
URL https://ieeexplore.ieee.org/document/7456206/

[26] Shaikh, F.: *Gradient Descent algorithm*. [Online; navštíveno 02.05.2018].
URL https://www.analyticsvidhya.com/blog/2017/03/introduction-to-gradient-descent-algorithm-along-its-variants/

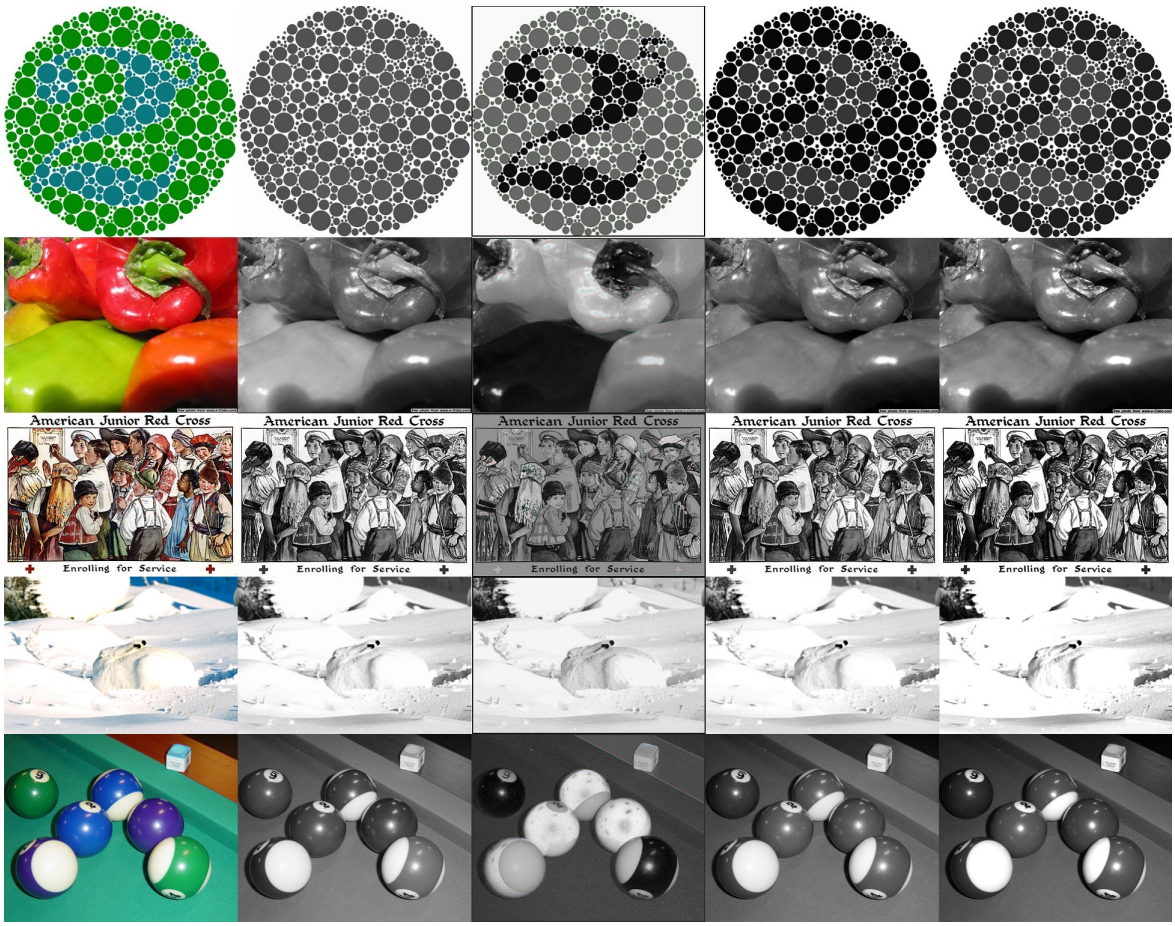
[27] Zhongping, J.; Mei-e, F.; Yigang, W.; aj.: *Efficient decolorization preserving dominant distinctions*. [Online; navštíveno 11.05.2018].
URL https://link.springer.com/article/10.1007/s00371-015-1145-4

[28] Čadík, M.: *Perceptual evaluation images*. [Online; navštíveno 11.05.2018].
URL http://cadik.posvete.cz/color_to_gray_evaluation/cadik08perceptualEvaluation-images.zip

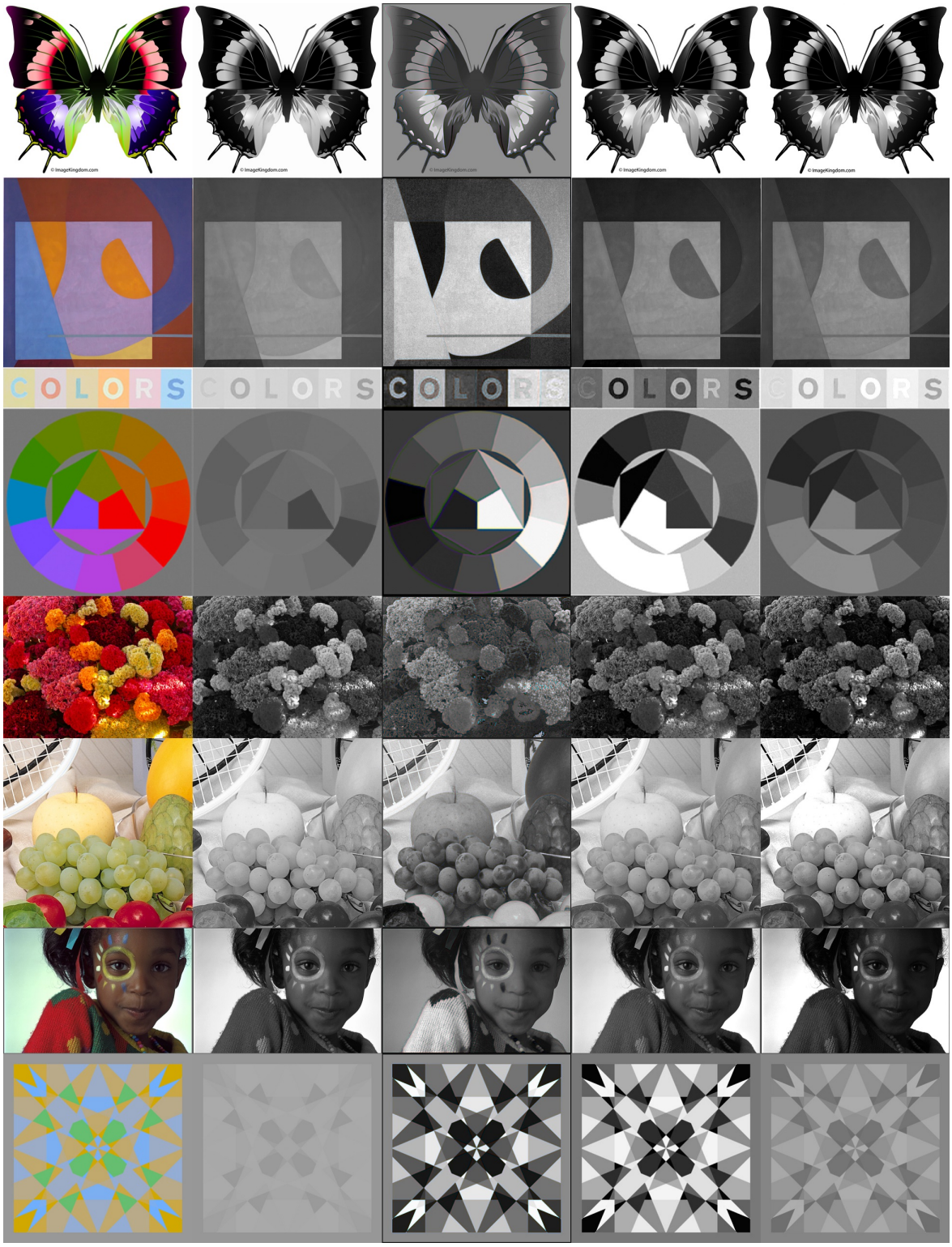
Příloha A

Porovnání dosažených výsledků jednotlivých metod

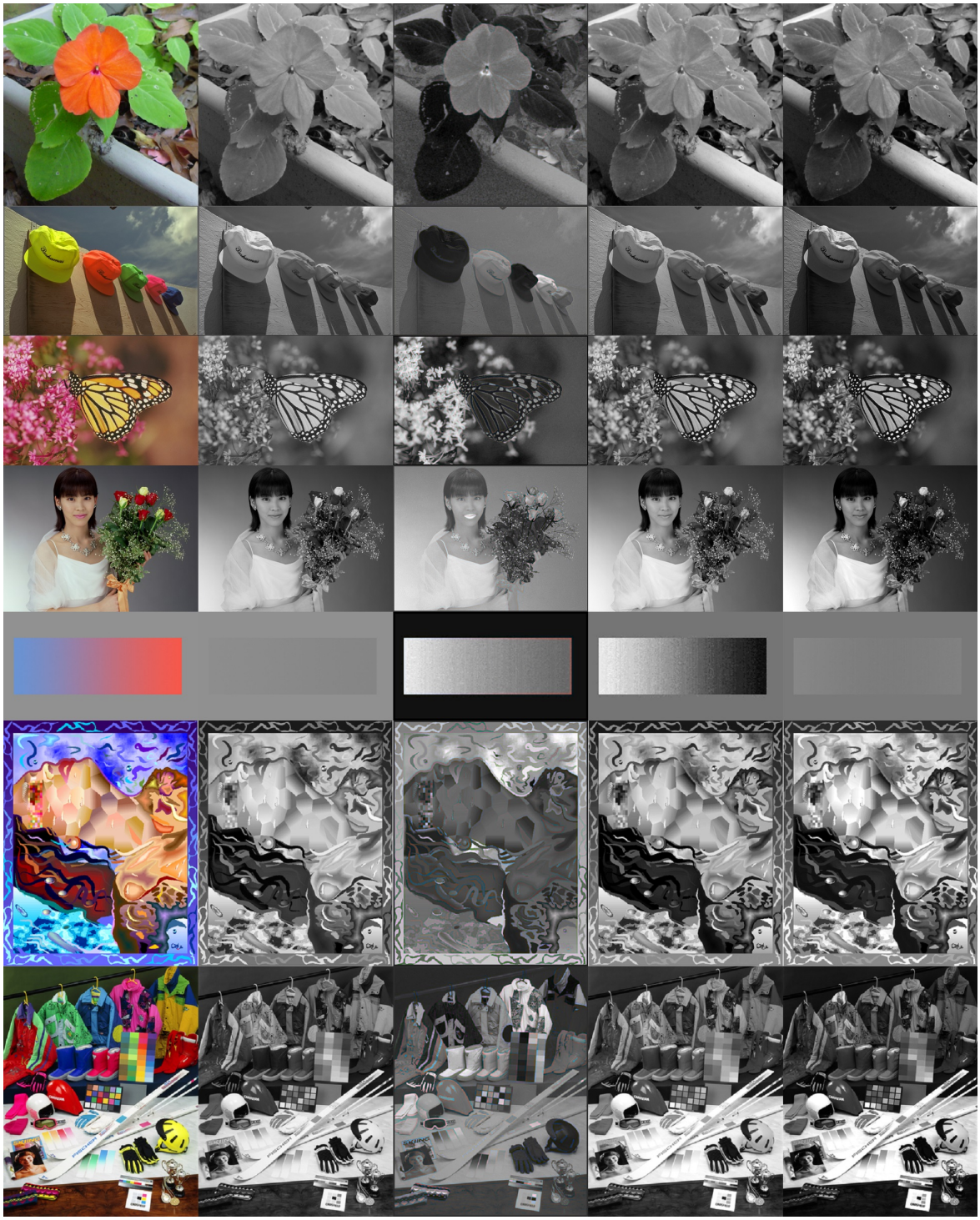
Reprezetace dosažených výsledků pro všechny tři implementované metody na vstupním obrázkovém datasetu doc. Martina Čadíka[28].



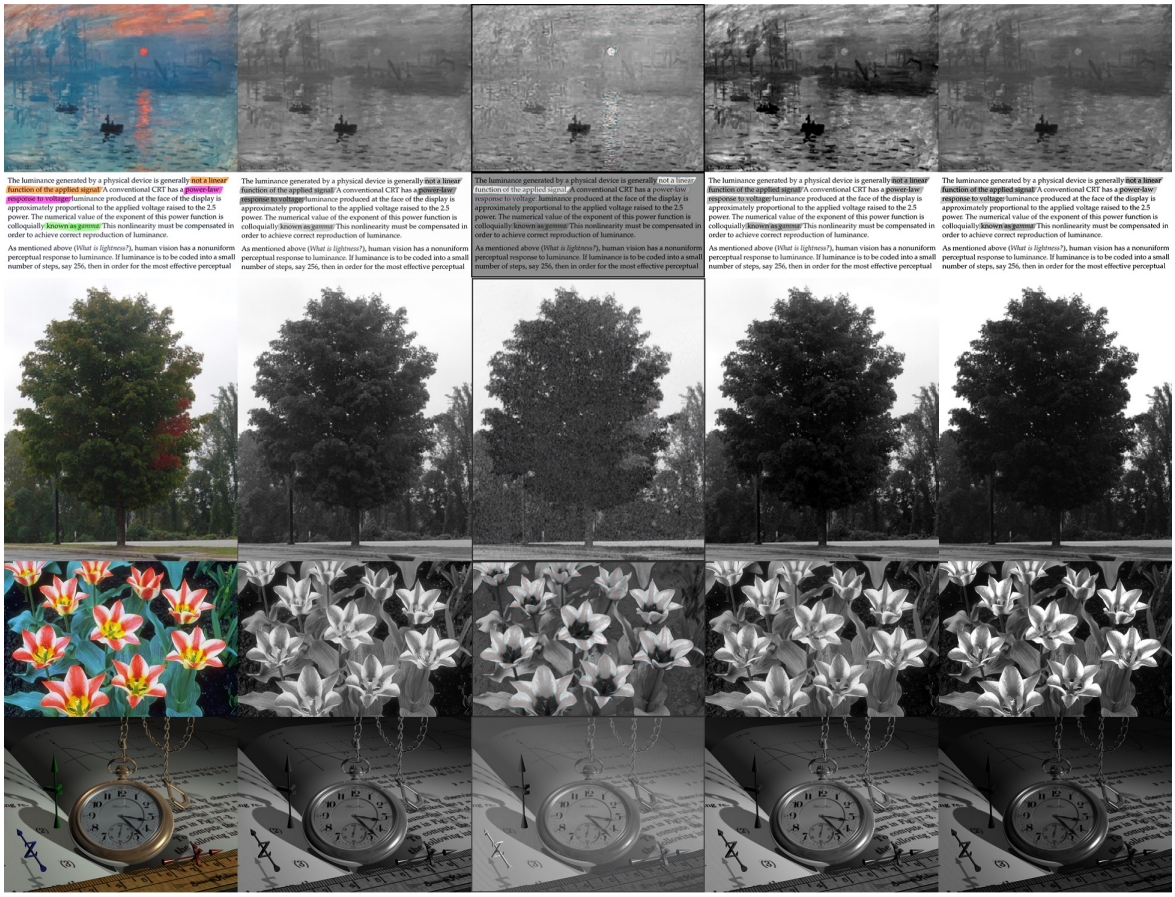
Obrázek A.1: vlevo původní snímek, druhý sloupec *rgb2gray* viz vzorec 3.1, třetí sloupec – *Zhongping15* 3.3, čtvrtý sloupec – *Xiong17* 3.4, pátý sloupec – *Jin17* 3.5



Obrázek A.2: vlevo původní snímek, druhý sloupec *rgb2gray* viz vzorec 3.1, třetí sloupec – *Zhongping15* 3.3, čtvrtý sloupec – *Xiong17* 3.4, pátý sloupec – *Jin17* 3.5



Obrázek A.3: vlevo původní snímek, druhý sloupec *rgb2gray* viz vzorec 3.1, třetí sloupec – *Zhongping15* 3.3, čtvrtý sloupec – *Xiong17* 3.4, pátý sloupec – *Jin17* 3.5



Obrázek A.4: vlevo původní snímek, druhý sloupec *rgb2gray* viz vzorec 3.1, třetí sloupec – *Zhongping15* 3.3, čtvrtý sloupec – *Xiong17* 3.4, pátý sloupec – *Jin17* 3.5