



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

**UŽIVATELSKÉ ROZHRANÍ SYSTÉMU PRO PRÁCI S HDR
OBRAZEM**

USER INTERFACE FOR HDR TONE MAPPING SYSTEM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JAN JEDLIČKA

VEDOUcí PRÁCE

SUPERVISOR

doc. Ing. MARTIN ČADÍK, Ph.D.

BRNO 2021

Zadání diplomové práce



Student: **Jedlička Jan, Bc.**
Program: Informační technologie
Obor: Informační systémy
Název: **Uživatelské rozhraní systému pro práci s HDR obrazem**
User Interface for HDR Tone Mapping System
Kategorie: Uživatelská rozhraní
Zadání:

1. Seznamte se s problematikou HDR fotografie, proveďte rešerši existujících uživatelských rozhraní pro zpracování HDR fotografie a videa.
2. Modifikujte existující uživatelské rozhraní systému Tone Mapping Studio (TMS) s ohledem na uživatelské požadavky. Definujte požadavky na funkčnost a jednotlivé prvky rozhraní prototypujte a testujte.
3. Se systémem experimentujte, posuďte jeho vlastnosti, proveďte uživatelský experiment a diskutujte možnosti budoucího vývoje.
4. Existující zásuvné moduly systému TMS porovnejte v uživatelské studii, příp. implementujte další moduly.
5. Dosažené výsledky prezentujte formou videa a plakátu, příp. článku.

Literatura:

- dodá vedoucí práce
- <https://github.com/cadik/TMS>

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Čadík Martin, doc. Ing., Ph.D.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2020

Datum odevzdání: 30. července 2021

Datum schválení: 13. listopadu 2020

Abstrakt

Tato práce si klade za cíl zprovoznění a vylepšení uživatelského rozhraní programu Tone Mapping Studio (TMS), který je vyvíjen na FIT VUT v Brně pod vedením doc. Ing. Čadíka, PhD. Současné uživatelské rozhraní je realizováno pomocí frameworku Qt3, který je již zastaralý a přestává být podporován. Je tedy zapotřebí převod do aktuální verze Qt5. V rámci vylepšení uživatelského rozhraní bude provedeno porovnání s již existujícími programy pro práci s HDR obrazem. Na základě porovnání budou navrženy změny, které budou prototypovány a uživatelsky testovány. Druhou částí této práce bude porovnání existujících zásuvných modulů systému TMS pro převod snímků do šedotónové barvy.

Abstract

The goal of this thesis is to improve graphical user interface of Tone Mapping Studio (TMS) program. This program is being developed on the Faculty of Information Technology (FIT), Brno University of Technology (BUT) by doc. Ing. Martin Čadík, PhD. The current program is using framework *Qt3*, which is old and not compatible with modern libraries. This program has to be rewritten to support current version *Qt5*. I will analyze other programs in the area of working with High Dynamic Range (HDR) images and video. Changes for improving the interface will be proposed and UX tests will be done. Second part will consist of comparing plug-ins for converting images to grayscale that already exists in TMS.

Klíčová slova

Uživatelské rozhraní, Qt, High Dynamic Range, HDR, fotografie, Tone Mapping Studio, TMS

Keywords

User Interface, UI, Qt, High Dynamic Range, HDR, photography, Tone Mapping Studio, TMS

Citace

JEDLIČKA, Jan. *Uživatelské rozhraní systému pro práci s HDR obrazem*. Brno, 2021. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. Ing. Martin Čadík, Ph.D.

Uživatelské rozhraní systému pro práci s HDR obrazem

Prohlášení

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

.....

Jan Jedlička
30. července 2021

Poděkování

Tímto bych rád poděkoval vedoucímu mé práce doc. Ing. Martinu Čadíkovi, PhD. za odborné vedení a rady poskytnuté během řešení této práce.

Obsah

| | | |
|----------|---|-----------|
| 1 | Úvod | 3 |
| 2 | Stávající řešení | 4 |
| 2.1 | Tone Mapping Studio | 4 |
| 2.1.1 | Problémy současného řešení <code>tmogui</code> | 5 |
| 2.2 | Jiné programy | 5 |
| 2.2.1 | Picturenaut 3 | 5 |
| 2.2.2 | Luminance HDR | 6 |
| 2.2.3 | Photomatix Pro | 7 |
| 2.2.4 | Adobe Lightroom | 8 |
| 2.3 | Výsledek analýzy | 9 |
| 3 | Implementace | 10 |
| 3.1 | Převod <i>Qt3</i> na <i>Qt4</i> | 10 |
| 3.2 | Převod <i>Qt4</i> na <i>Qt5</i> | 11 |
| 3.3 | Uživatelské testování | 12 |
| 3.4 | Plán testování | 12 |
| 3.5 | Výsledky uživatelského testování | 12 |
| 3.6 | Úprava rozhraní v <i>Qt5</i> | 13 |
| 3.6.1 | Živý náhled | 13 |
| 3.6.2 | Zjednodušení prostředí | 13 |
| 3.6.3 | Ukázka výsledků aplikace operátorů | 14 |
| 3.7 | Překlad a spuštění programu | 16 |
| 3.8 | Závěr | 17 |
| 4 | Uživatelská studie | 18 |
| 4.1 | Teorie | 18 |
| 4.1.1 | Barevné modely | 18 |
| 4.1.2 | Převod na stupně šedi | 20 |
| 4.1.3 | Převod videa | 21 |
| 4.1.4 | Možné chyby převodu | 21 |
| 4.2 | Analýza podobných studií | 21 |
| 4.2.1 | Porovnání časově závislých metod pro převod barevného obrazu na šedotónový (2018) | 21 |
| 4.2.2 | Assessment of video tone-mapping (2013) | 22 |
| 4.2.3 | Perceptual Evaluation of Color-to-Grayscale Image Conversions (2008) | 22 |
| 4.3 | Operátory knihovny TMO | 23 |
| 4.3.1 | Contrast Enhancing Colour to Grey (2009) | 23 |

| | | |
|----------|---|-----------|
| 4.3.2 | Laplacian-Guided Image Decolorization (2016) | 24 |
| 4.3.3 | Variance Maximization for Decolorization (2014) | 24 |
| 4.3.4 | Robust Color-to-gray via Nonlinear Global Mapping (2009) | 25 |
| 4.3.5 | Reducing Dimensions (2010) | 26 |
| 4.3.6 | Spectral Image Decolorization (2010) | 27 |
| 4.3.7 | Efficient Color-to-Gray Conversion for Digital Images in Gradinet Domain (2015) | 27 |
| 4.3.8 | Efficient decolorization preserving dominant distinctions (2015) | 28 |
| 4.4 | Plán uživatelské studie | 29 |
| 4.4.1 | Dataset | 29 |
| 4.4.2 | Metodika uživatelské studie | 30 |
| 4.4.3 | Vyhodnocování preferencí | 31 |
| 4.4.4 | Realizace | 34 |
| 4.5 | Výsledky uživatelské studie | 37 |
| 4.5.1 | Výsledky pro celý dataset | 37 |
| 4.5.2 | Výsledky pro jednotlivá videa | 38 |
| 5 | Závěr | 40 |
| | Literatura | 41 |
| A | Obsah přiloženého paměťového média | 43 |
| B | Příklad zadání uživatelského testování | 44 |
| C | Dataset | 45 |
| D | Výsledky pro jednotlivá videa | 49 |
| E | Odkazy na videa | 52 |

Kapitola 1

Úvod

Běžné formáty využívané ve fotografii a videu dokáží uložit obsah snímku pouze s omezeným rozsahem hodnot (Low Dynamic Range - dále LDR). Pro možnost uložit celý rozsah hodnot jasu a barev reálného světa však existuje možnost využít formátu s vysokým dynamickým rozsahem. High Dynamic Range (dále HDR) snímky jsou využitelné v mnoha jiných směrech než jen fotografie. Dají se například využít v 3D grafice jako zdroj osvětlení scény nebo v lékařství. HDR snímek není zobrazitelný na běžném zobrazovacím médiu a proto je potřeba metod pro komprimaci na LDR snímek, které se označují jako tone mappingové operátory.

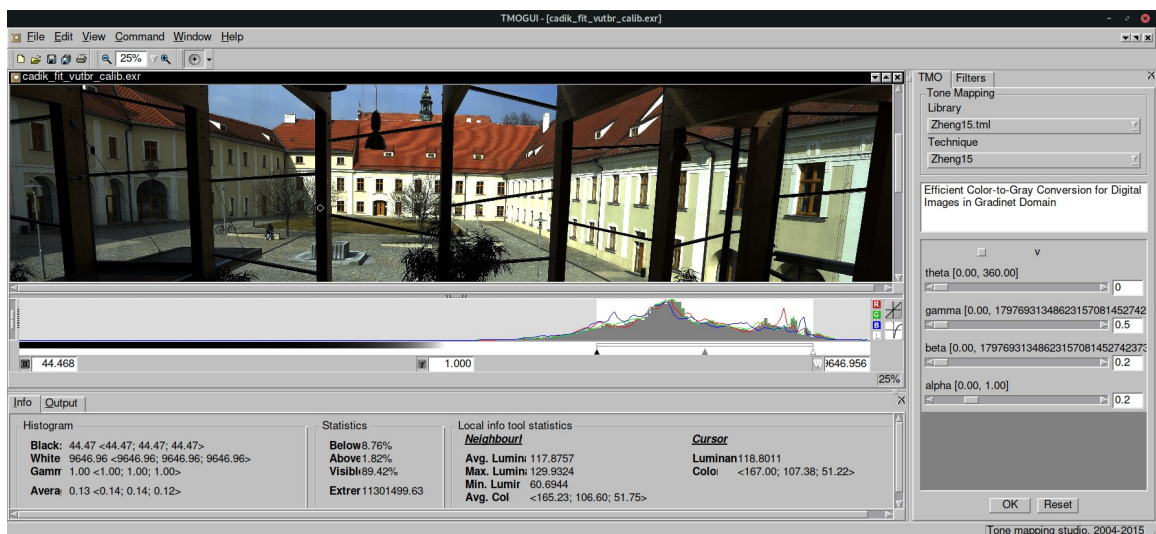
Pro jednodušší implementaci těchto operátorů byl vyvinut program *Tone Mapping Studio* (dále TMS). Cílem této práce je modernizace a vylepšení grafického uživatelského prostředí tohoto programu. V kapitole 2 se nejdříve zaměřím na program samotný a analyzuji rozhraní podobných existujících programů. Kapitola 3 pojednává o samotné implementaci a využitých technologiích. V kapitole 3.3 se poté konkrétněji zaměřím na uživatelské testování a porovnání zásuvných modulů knihovny TMS, zaměřených na převod snímků do černobílé.

Kapitola 2

Stávající řešení

V této kapitole se budu zabývat současným stavem grafického uživatelského rozhraní programu *Tone Mapping Studio*¹ a analýzou uživatelského rozhraní oblíbených programů určených pro práci s HDR obrazem. Vzhledem k oblíbenosti HDR fotografie se výběr značně rozšířil. Z analyzovaných programů jsem vybral tři zástupce, které popíši níže.

2.1 Tone Mapping Studio



Obrázek 2.1: Uživatelské rozhraní programu TMS v Qt3.

*Tone Mapping Studio*¹ (dále *TMS*) je framework, který od roku 2014 vzniká v rámci výzkumné skupiny CPhoto@FIT pod vedením doc. Ing. Martina Čadíka, PhD. na fakultě informatiky VUT v Brně. Poskytuje prostředí pro vývoj metod (implementaci operátorů) pro práci s HDR obrazy a videem[19]. Framework poskytuje nástroje pro načítání a ukládání souborů ve vícero formátech. Dále se například stará o zpracování barevných profilů. Je použitelný jako konzolová aplikace nebo lze využít grafického uživatelského rozhraní *tmogui*.

¹<https://github.com/cadik/TMS>

2.1.1 Problémy současného řešení *tmogui*

V současné době je grafické uživatelské rozhraní implementováno pomocí knihovny *Qt3*² (obr. 2.1), která vznikla v roce 2001. Hlavní problém tohoto řešení je zastaralá verze, která již není kompatibilní s nejnovější verzí knihovny *OpenCV 4*³, která je určená pro počítačové vidění a strojové učení. Program nelze spustit a končí segmentační chybou. Hlavním úkolem této práce tedy je zprovoznění funkcionality *tmogui* převodem z *Qt3* na aktuálně nejnovější verzi *Qt5*. Popisem tohoto procesu se budu zabývat v kapitole 3. Dalším podnětem pro převod je vzhled prostředí, který působí velice zastaralým dojmem, což snižuje použitelnost. Rozvržení prvků rozhraní není špatné, ale při zobrazení potřebných panelů nezůstává dostatek místa pro samotnou fotografii.

2.2 Jiné programy

Obliba HDR fotografie a práce s tímto formátem značně roste. Využití HDR principů se dostalo téměř do všech mobilních telefonů. Na trhu se tedy objevilo hned několik programů sloužících zejména pro vytvoření HDR souboru ze sekvence různě exponovaných snímků či pro řízený převod HDR snímku do LDR zobrazení pomocí metod tone mappingu. V dnešní době není problém zpracovávat HDR fotografie ve známých a hojně používaných grafických programech jako například Adobe Photoshop, nebo Adobe Photoshop Lightroom. V mé analýze se však zaměřím pouze na volně dostupné aplikace specializované na práci s HDR daty.

Jedním z hlavních problémů, které musí program řešit, je způsob zobrazení zdrojového HDR obrazu. Jedná se o data, která nejsou zobrazitelná na běžných LDR monitorech. Možností jak k tomuto problému přistoupit se nabízí několik a každému uživateli může vyhovovat jiná. Také je potřeba zmínit možnost zobrazení hodnot pro určitý pixel obrazu. Dále je důležitou součástí možnost výběru operátoru pro tone mapping a nastavení parametrů pro zvolenou metodu. V mé analýze se také soustřeďuji na rozšířitelnost funkcionality a složitost ovládání rozhraní.

2.2.1 *Picturenaut 3*

*Picturenaut 3*⁴ je program pro operační systémy Mac OS a Windows s open-source licencí. Hlavní předností programu je možnost vytváření vlastních zásuvných modulů. Je zaměřený především na vytváření HDR obrazů pomocí skládání většího množství různě exponovaných LDR fotografií a jejich tone mapping. Pomocí zásuvných modulů však lze jeho funkcionality značně rozšířit. Výhodou je kompatibilita se zásuvnými moduly vyvinutými pro program *HDRshop*⁵. Program také poskytuje možnost změny velikosti snímku, jeho otočení nebo vygenerování html stránky s interaktivním náhledem fotografie. Podporuje velké množství formátů jak HDR fotografií, tak například RAW formátů od několika výrobců fotoaparátů.

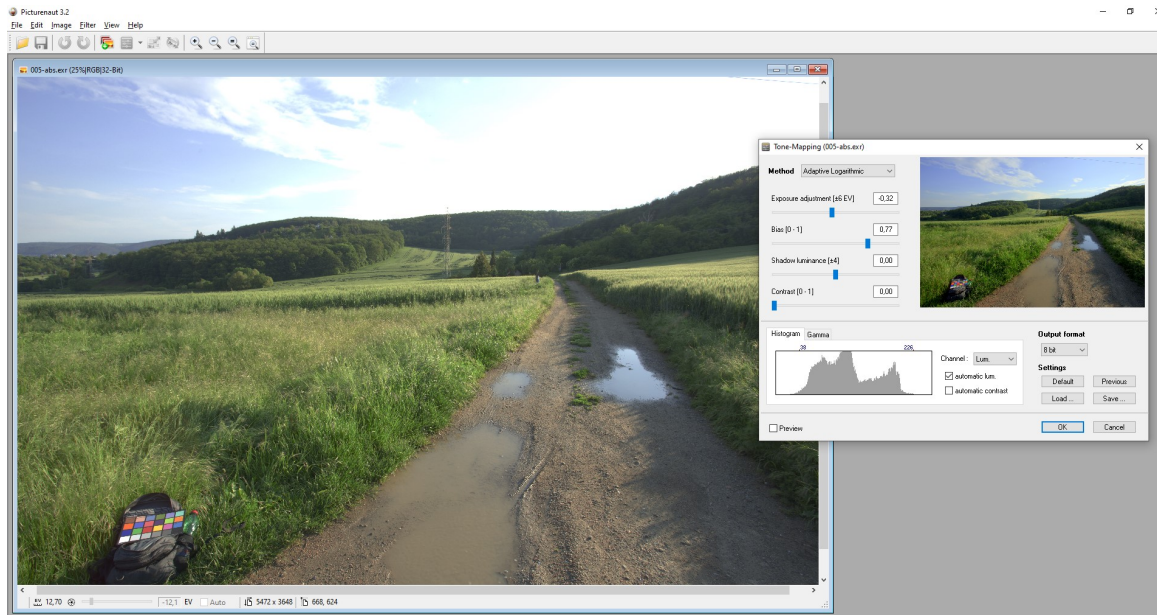
Uživatelské rozhraní (viz obr. 2.2) je velice strohé a jednoduché. Hlavním prvkem je okno se zdrojovým souborem. Horní lišta nám pak dává možnost otevřít okna pro jednotlivé úkony, které chceme aplikovat. Náhled zdrojového HDR obrazu je vypočítán automaticky a uživatel má na liště pod obrázkem možnost upravit expoziční hodnotu (EV), podle které se náhled vypočítává. Dále je možné v menu nastavit konkrétní hodnoty *gamma* korekce a

²<https://www.qt.io>

³<https://opencv.org>

⁴<http://www.hdrilabs.com/picturenaut>

⁵<http://gl.ict.usc.edu/HDRShop/> - program za jehož vývojem stojí prof. Paul Debevec.

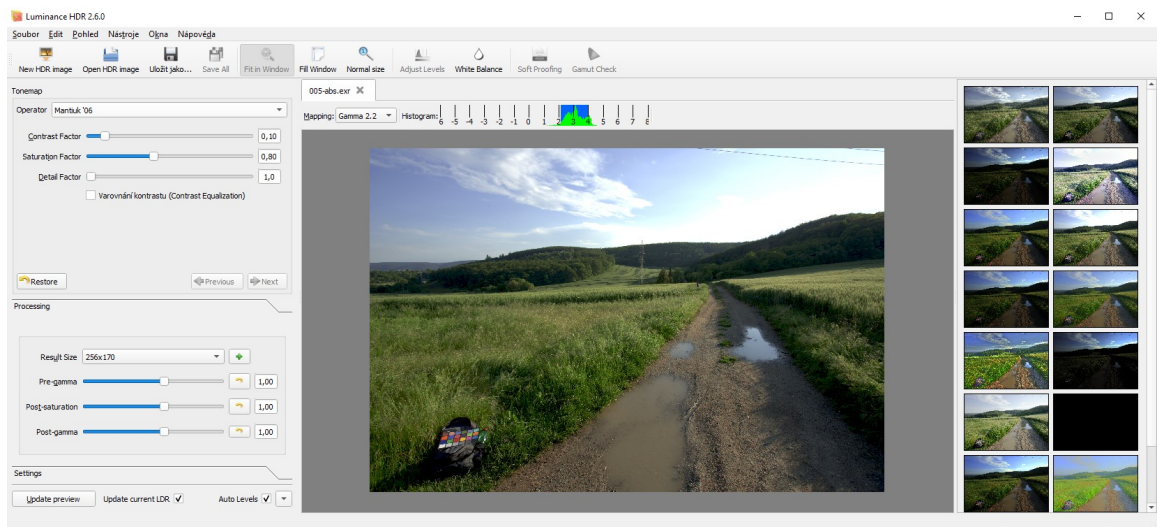


Obrázek 2.2: Uživatelské rozhraní programu *Picturenaut 3*.

offset. Výběr metody pro tone mapping a zadání parametrů je situováno do vlastního okna, kde je k dispozici i náhled výsledku zpracování a výsledný histogram s možností upravit hodnoty pro výstup.

Práce s tímto programem je velmi jednoduchá a intuitivní. Prvky uživatelského rozhraní jsou jednoduché a věcné. Bez zásuvných modulů má uživatel k dispozici pouze čtyři metody tone mappingu. Je nejspíše určen pro jednoduchou práci s HDR daty a pro vývojáře, kteří potřebují platformu pro tvorbu vlastních algoritmů.

2.2.2 Luminance HDR

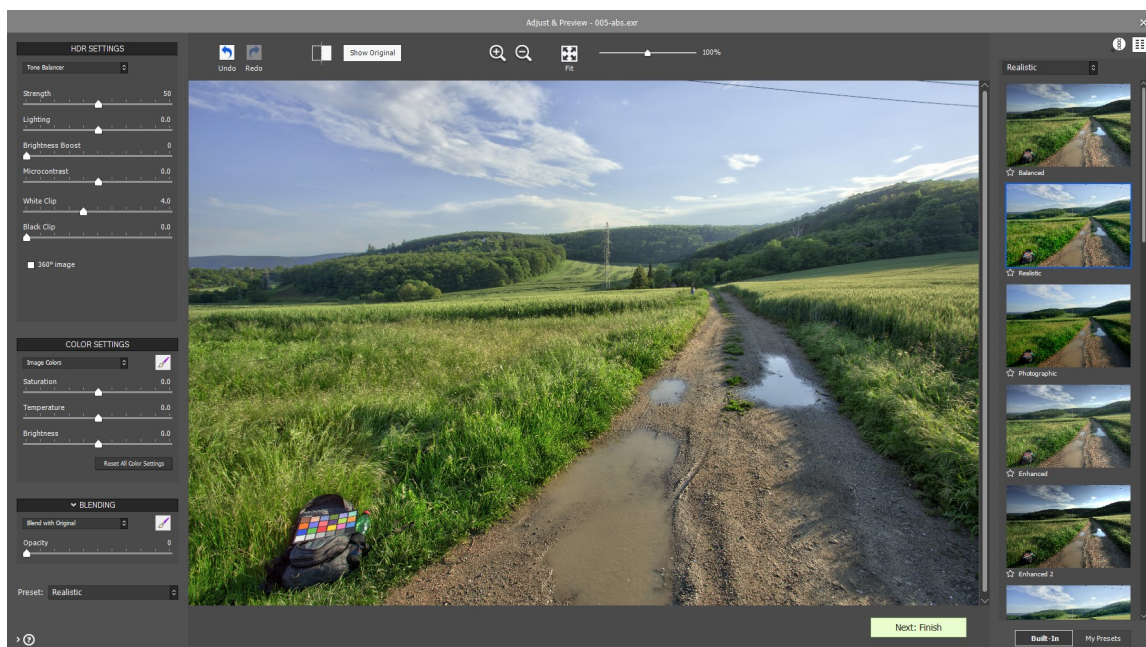


Obrázek 2.3: Uživatelské rozhraní programu *LuminanceHDR*.

*Luminance HDR*⁶ je program pro operační systémy Linux, Mac OS a Windows s open-source licencí. Podporuje skládání LDR snímků do HDR obrazu, tone mapping a práci s HDR obrazem. Náhled zdrojového obrazu je řízen posuvníkem nad histogramem hodnot jasu s možností volby gamma korekce. Tento přístup dává uživateli možnost seznámit se s rozsahem jasových hodnot obrázku a umožňuje větší kontrolu nad zobrazením LDR náhledu. Výběr tone mappingových operátorů je situován v levé části. Ve verzi 2.6.0 je implementováno patnáct operátorů. Výrazným prvkem uživatelského rozhraní je pravý sloupec poskytující náhled výsledků tone mappingu jednotlivých operátorů. Po nastavení parametrů a stisknutí tlačítka **Update preview** se vytvoří náhled ve zvoleném rozlišení, který je poté možné uložit. Živý náhled úprav parametrů je realizován pouze na miniaturách v pravém sloupci.

Uživatelské rozhraní je přímočaré a jednoduché na ovládání. Program poskytuje možnost výběru nastavení vzhledu a pozice některých prvků rozhraní. Cílí především na zkušenější uživatele. Využívá knihovny *Qt* stejně jako *TMS*.

2.2.3 Photomatix Pro

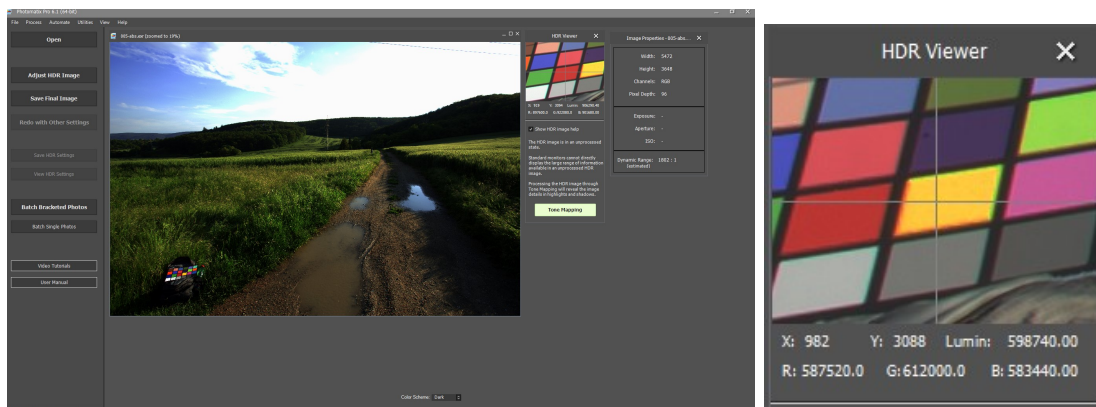


Obrázek 2.4: Uživatelské rozhraní programu *Photomatix pro*.

*Photomatix Pro*⁷ je placený program, který nabízí možnost vyzkoušení zdarma. Je navržen pro operační systémy Mac OS a Windows. Nabízí také možnost integrace do prostředí Adobe Photoshop Lightroom (viz 2.2.4). Umožňuje dávkové zpracování fotografií. Zajímavým prvkem při náhledu HDR fotografie je *HDR Viewer* (obrázek 2.5). Zobrazuje část HDR obrázku pod kurzorem myši spolu s informacemi o hodnotách jasu a barev pixelu na tomto místě. Tyto informace jsou velmi užitečné a dají se využít například při fotometrické kalibraci HDR snímků.

⁶<http://qtpfsgui.sourceforge.net>

⁷<https://www.hdrsoft.com>



Obrázek 2.5: *Photomatix pro* - zobrazení části fotografie s informacemi.

Uživatelské prostředí je mířeno výrazně na fotografy. K dispozici je velké množství přednastavených konfigurací s náhledy. Použité operátory nejsou přímo prezentovány a uživatel upravuje výstup pomocí vlastních parametrů programu, které jsou pojmenovány přímo podle změn, které působí (např. "*Detail Contrast*"). Ovládání je v celku jednoduché a program obsahuje nápovědu vysvětlující funkci všech prvků rozhraní a parametrů.

2.2.4 Adobe Lightroom



Obrázek 2.6: Uživatelské rozhraní programu *Adobe Lightroom*.

*Adobe Photoshop Lightroom*⁸ je placený program určený pro správu a editaci fotografií. Přímo nepodporuje práci s HDR snímky, ale je do něj možné integrovat již zmíněný program *Photomatix Pro*. Jeho uživatelské rozhraní je velmi dobře rozvržené a intuitivní. V režimu úpravy fotek se fotografie zobrazuje ve středu obrazovky. Na levé straně je panel obsahující přednastavené filtry a malý náhled snímku na kterém můžeme při výběru filtrů pozorovat

⁸<https://www.adobe.com>

výsledek po aplikaci filtru. V pravém panelu jsou pouze ovládací prvky pro přímou úpravu fotografie a histogram. Zajímavým detailem je vzhled jednotlivých posuvníků, který koresponduje s vlivem daného posuvníku na fotografii. Další ovládací prvky jsou umístěny v liště pod fotografií. Jednotlivé panely je možné skrýt a získat tak větší prostor pro samotnou fotografii. V horní liště programu si pak uživatel může měnit prostředí programu podle toho jakou činnost chce právě vykonávat. Rozhraní má tedy několik verzí, které jsou specializované na konkrétní úkony.

Možnost skrývání panelů je určitě přínosná a měla by být přítomna i v *TMS. Lightroom* je důkazem toho, že i velké množství ovládacích prvků lze zobrazit takovým způsobem aby stále zbylo dostatek místa na fotografii. To by však nebylo možné, pokud by měly všechny ovládací prvky všech funkcí programu být viditelné na jedné obrazovce.

2.3 Výsledek analýzy

Za hlavní výstup analýzy považuji nutnost definovat cílovou skupinu uživatelů. Existující řešení se dají rozdělit na dvě skupiny. První cílí na fotografy (z uvedených nejvíce *Photomatrix Pro* a *Adobe Lightroom*) a nejdůležitějším prvkem rozhraní je živý náhled výsledku tone mappingu. Druhou skupinou jsou programy vhodné pro vývojáře, které poskytují možnost rozšíření pomocí zásuvných modulů (*Picturenaut 3*) a obsahují implementace známých operátorů s možností upravovat přímo jejich parametry. Mezi nejvíce implementované operátory patří například práce Erika Reinharda[17], nebo Rafala Mantiuka[14][13].

Tone Mapping Studio je určené více pro druhou zmiňovanou skupinu. Vzhledem k předpokládané technické zdatnosti uživatele se po analýze jeví upravování náhledu zdrojového souboru pomocí posuvníku pod histogramem jako nejlepší volba interakce. Poskytuje plnou kontrolu nad náhledem a možnost analyzovat obsažená data.

Ve většině programech naprosto chyběla možnost zobrazení hodnot jednotlivých pixelů. Zajímavý přístup má v tomto směru program *Photomatrix Pro*, který poskytuje prvek *HDR Viewer* (viz obrázek 2.5), který uživateli zobrazí část snímku nacházející se pod kurzorem myši i s hodnotami pixelu ve středu.

Dalším prvkem rozhraní, který se často v programech vyskytoval, jsou náhledy výsledků tone mappingu pomocí poskytovaných operátorů. Tento prvek je použitelný hlavně pro fotografy a v kontextu Tone Mapping Studia nemá tak velký význam. Některé operátory TMS jsou bohužel náročné na výpočet. Přijatelným vylepšením by mohla být možnost zobrazit miniaturu náhledu právě zpracovávaného snímku, která by živý náhled poskytovala.

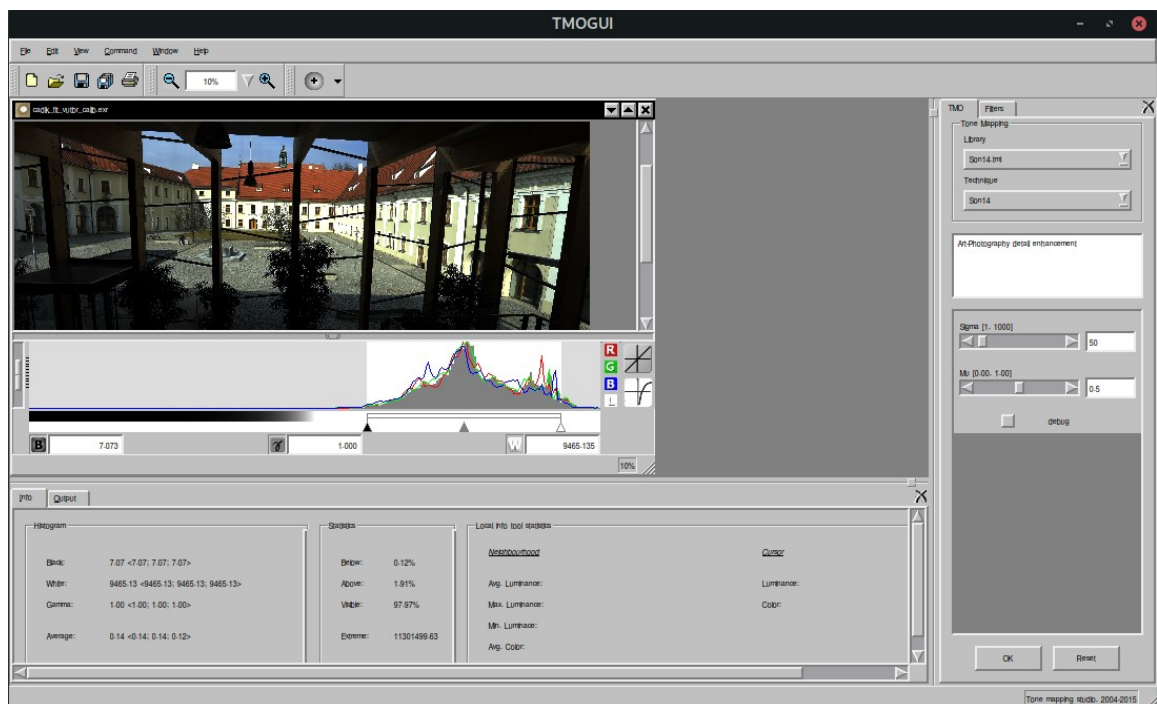
Hlavním rozdílem mezi Tone Mapping Studiem a většinou analyzovaných programů byl celkový vzhled. Tone Mapping Studio působí nemoderně a nepříliš efektivně využívá místo. To nejen kazí dojem z programu, ale také snižuje přehlednost a intuitivnost ovládání.

Kapitola 3

Implementace

Implementaci nového uživatelského rozhraní jsem rozdělil na tři hlavní části. Nejdříve využiji existujících automatizovaných nástrojů pro převod mezi verzemi *Qt3* a *Qt4*. Dále je pro převod do *Qt5* potřeba odstranit závislost na modulu pro zpětnou podporu *Qt3*. Třetí fáze spočívá v odstranění chyb způsobených převodem a vylepšení nového rozhraní na základě uživatelského testování. V průběhu implementace bylo potřeba využít starší verzi knihovny OpenCV, která není závislá na *Qt5*. Vývojovým prostředím pro mě byly programy *Qt Creator* a *Qt Designer*. Vyvíjel jsem v operačním systému Linux (Fedora).

3.1 Převod *Qt3* na *Qt4*

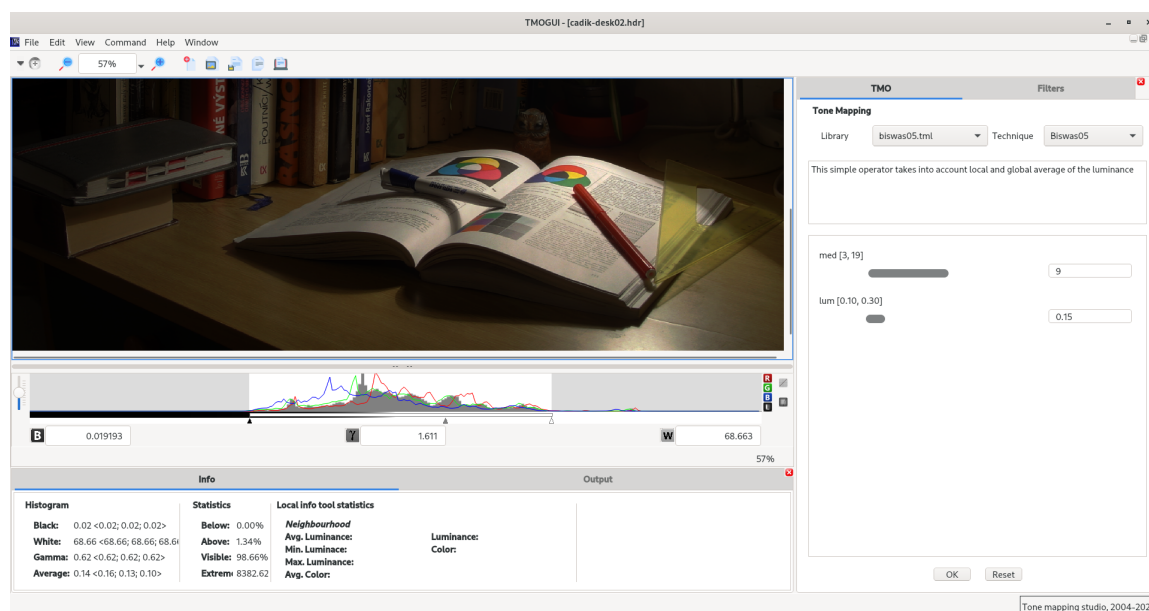


Obrázek 3.1: TMS po převodu z *Qt3* na *Qt4*. Velikost písma bude opravena až s podporou *Qt5*

Pro přepsání rozhraní do verze Qt4 jsem využil postupu v dokumentaci frameworku¹ a projekt od Ing. Tomáše Popely², který se již o transformaci na novější verzi pokoušel. Qt poskytuje nástroj `qt3to4`, který automaticky transformuje zdrojový kód programu. Tento nástroj využívá modulu `Qt3Support`, který byl přidán do `Qt4` právě pro podporu tříd a metod z `Qt3`. Tento modul není přítomen v `Qt5`. Pro naše účely tedy pomůže nástroj pouze identifikovat třídy, které bude zapotřebí ručně přepsat. Odstraněním závislosti na tomto modulu se zabývá následující sekce 3.2.

Zároveň bylo potřeba předělat všechny soubory s definicí prvků rozhraní (`.ui`), jelikož s `Qt4` přišla také nová verze programu `Qt Designer`. Pro převod těchto souborů byl použit nástroj `uic3`. Také byl vytvořen nový soubor se zdroji (`.qrc`), do kterého se vložily zdroje obrázků pro ikony programu. Cílem této části byla úspěšná kompilace pomocí verze `Qt4`. Chyby vzniklé převodem a již existující chyby programu budou opraveny až po převodu do `Qt5`.

3.2 Převod Qt4 na Qt5



Obrázek 3.2: TMS po převodu z `Qt4` na `Qt5`.

Pro podporu `Qt5` bylo potřeba zbavit program závislosti na modulu `Qt3Support`. Největší změny vyžadovaly třídy využívající `Q3ToolBar`, `Q3GridLayout`, `Q3VBox`, `Q3HBox`, `Q3ScrollView`, `QCustomEvent`, `Q3FileDialog` a `Q3PopupMenu`.

Po odstranění těchto závislostí jsem postupoval podle oficiální příručky na stránkách knihovny Qt³. Šlo zejména o úpravu souboru `CMakeList`, opravu importů knihoven a následně přepisování již nefunkčních metod a tříd. Celkem se jednalo o téměř pět tisíc řádků kódu. Některé konstrukce widgetů již nebyly možné a musel jsem přistoupit k jejich mo-

¹<https://doc.qt.io/archives/qt-4.8/porting4.html>

²<https://github.com/tpopela/TMS>

³https://wiki.qt.io/Transition_from_Qt_4.x_to_Qt5

difikaci. Velkým problémem se ukázala změna v definici rozložení prvků widgetu, jejich umístěním a velikostí.

3.3 Uživatelské testování

Uživatelské testování je důležitou součástí vývoje uživatelského rozhraní. V této části se zaměřím na plán uživatelských testů, které využiji na identifikaci slabých míst programu. Pro plán testování jsem čerpal především z knihy Stevea Kruga „*Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability Problems*“. Jedná se o příručku pojednávající o přístupech k uživatelskému testování a jednoduchých testech v malém měřítku. V každém programu lze najít něco k vylepšení.

3.4 Plán testování

Testování probíhalo v několika iteracích. V každé iteraci jsem k testování vybral pět lidí, kteří se s rozhraním ještě nesetkali. Testování obdrželi seznam úkolů k vypracování a byli požádáni o komentování svých činů a rozhodnutí. Do jejich rozhodování jsem v zásadě nezasahoval a po splnění úkolů jsme debatovali nad možnými vylepšeními, či problémy na které narazili.[10]

Uživatelské testy by měly otestovat především nejpoužívanější funkce programu. Jako hlavní jsem definoval následujících pět:

- Načtení a zobrazení souboru.
- Získání dat z načteného snímku.
- Nastavení parametrů vybraného operátoru a jeho aplikace na snímek.
- Uložení souboru ve zvoleném formátu.
- Orientace v uživatelském rozhraní programu.

Jednotlivé úkoly byly poté navrženy tak, aby subjekt musel využít výše definované funkce. Příklad zadaných úkolů naleznete v příloze B. Úkoly se v každé iteraci měnili tak, aby uživatel musel využít nově přidávaných prvků rozhraní. Z uživatelských testů vyplynulo hned několik návrhů na vylepšení rozhraní.

3.5 Výsledky uživatelského testování

První testování proběhlo ihned po zprovoznění programu v *Qt5*. Rozvržení prvků rozhraní (viz obr. 3.2) bylo stejné jako v původní verzi. Z testování vzešly následující připomínky:

- Velké množství chyb. Zřejmě způsobených převodem. Segmentační chyby, chyby zarovnání prvků, zobrazení snímku a podobné.
- Prostor pro fotografii je malý. Spodní panel s informacemi zabírá příliš místa.
- Chybí živý náhled výsledku aplikace operátoru. Uživatel neví co způsobí změna určitého parametru.

- Pro jednoduchou práci s programem obsahuje až příliš ovládacích prvků a informací. Prostředí je přívětivější pro zkušené uživatele.
- Výběr operátorů pouze podle jména. Chybí ukázka výsledku aplikace operátoru.

Po implementaci jednotlivých připomínek, či prototypů jejich řešení, se testování opakovalo (viz 3.6).

3.6 Úprava rozhraní v Qt5

První uživatelské testování ukázalo velký počet chyb způsobených převodem z Qt3. Jednalo se hlavně o zobrazení histogramu a nastavení hodnot pro zobrazení snímku, problémy se synchronizací vláken a velké množství segmentačních chyb. Zřejmě nejvíce času implementace zabralo právě odstraňování těchto problémů. Také došlo k přesunu karty s informacemi o fotografii do pravého panelu. Další uživatelský test potvrdil, že přesunutí bylo přínosné. Místo pro fotografii se rozšířilo a umístění mezi karty na bočním panelu se neukázalo jako překážka. Naopak byly informace pro uživatele snáze k nalezení. Výpočet a zobrazení histogramu bylo opraveno a úprava vzhledu fotografie byla pro uživatele jednodušší.

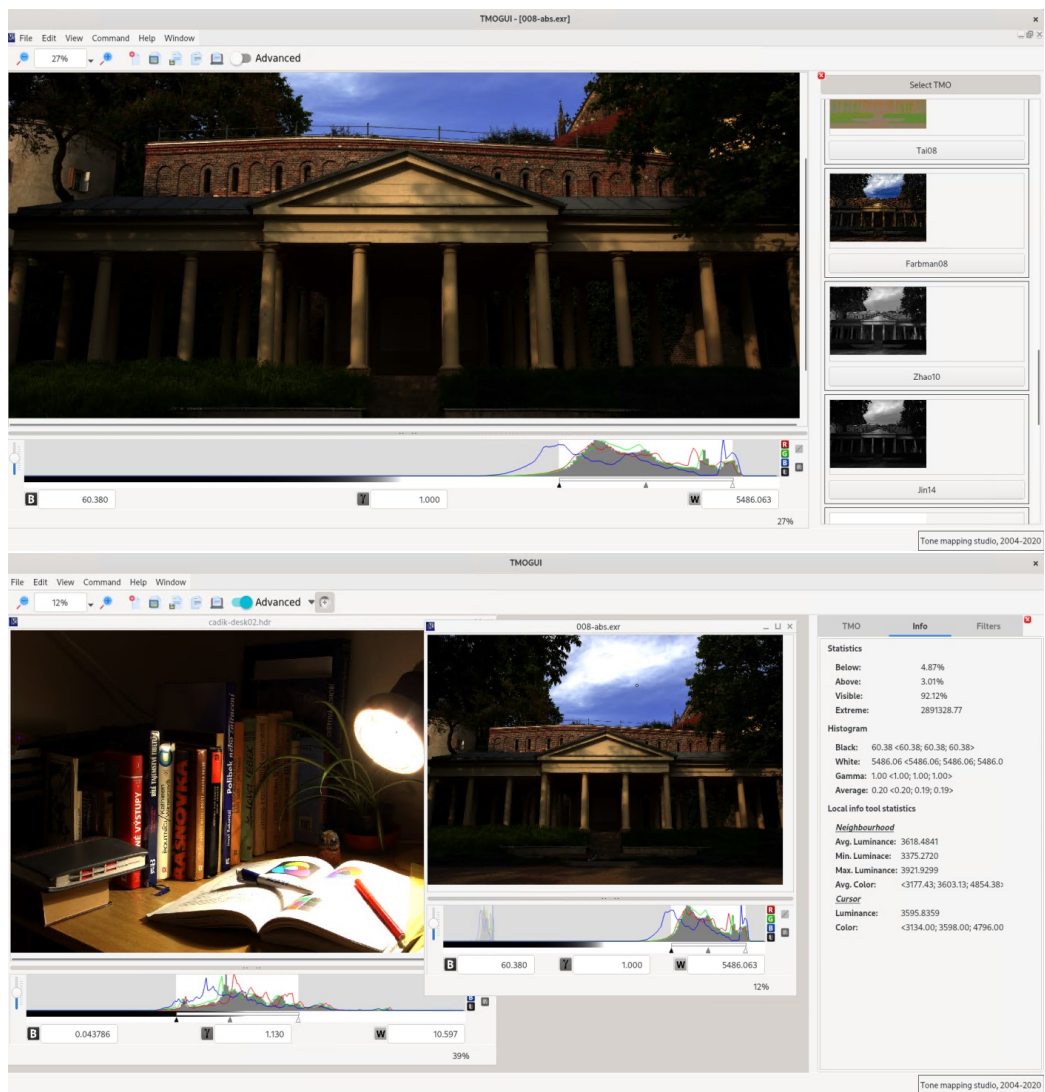
3.6.1 Živý náhled

Další připomínkou byla absence živého náhledu aplikace operátoru. Vzhledem k dlouhému času výpočtu některých operátorů nad rozměrnými fotografiemi jsem se rozhodl o vytvoření miniatury, na kterou se změny v nastavení parametrů budou živě promítat. Pro tento účel jsem vytvořil tři verze rozhraní s různým umístěním náhledu. Prvním prototypem byla miniatura bez ovládacích prvků umístěná v pravém panelu pod parametry. Druhým pak nové okno na pracovní ploše totožné s originálním oknem fotografie poskytující stejné ovládací prvky. Třetí prototyp nahrazoval originální obrázek náhledem.

Při testování prototypů jsem si všiml, že v případě třetího prototypu uživateli chyběla reference s originálním obrázkem (například při převodu obrázku do černobílé). U prvního prototypu chyběla většině uživatelů možnost náhled zvětšit a vzhledem k absenci histogramu nebylo možné upravit hodnoty pro zobrazení. Ne všechny operátory jsou implementované tak, aby jejich výstupní obrázek obsahoval hodnoty v intervalu $\langle 0, 1 \rangle$. Proto je možnost upravit rozsah hodnot nad histogramem nepostradatelná. Rozhodl jsem se tedy pro druhý prototyp, který uživateli poskytuje nejvíce možností a jeho umístění na pracovní ploše je volitelné. Náhled se automaticky aktualizuje při změně parametru a uživatel má tudíž možnost vidět vliv parametru na výsledek. Tento prvek výrazně zjednodušil práci s programem a vytvořil výhodu nad konzolovou aplikací, kde je výsledek vidět až na vygenerovaném výstupu.

3.6.2 Zjednodušení prostředí

Tone Mapping Studio je cílené hlavně pro vývoj operátorů a vědeckou práci. Obsahuje tedy mimo tone mapping také možnost analyzovat obrazová data souboru na úrovni pixelů. Uživatel má možnost barevně zvýraznit pixely s hodnotami v určitém rozmezí, nebo si nechat vypsát informace o konkrétní oblasti snímku. Tato funkcionality je žádoucí, ale pro potřeby tone mappingu není potřeba. Analytické prvky rozhraní pouze vytvářejí dojem složitosti a nový nezkušený uživatel se v něm může ztrácet. Vytvořil jsem tedy prototyp rozhraní bez zmíněných analytických prvků (viz obr. 3.3) a otestoval na uživateli, zda jejich absence



Obrázek 3.3: Porovnání jednoduchého (nahore) a složitějšího prostředí (dole), mezi kterými si může uživatel zvolit.

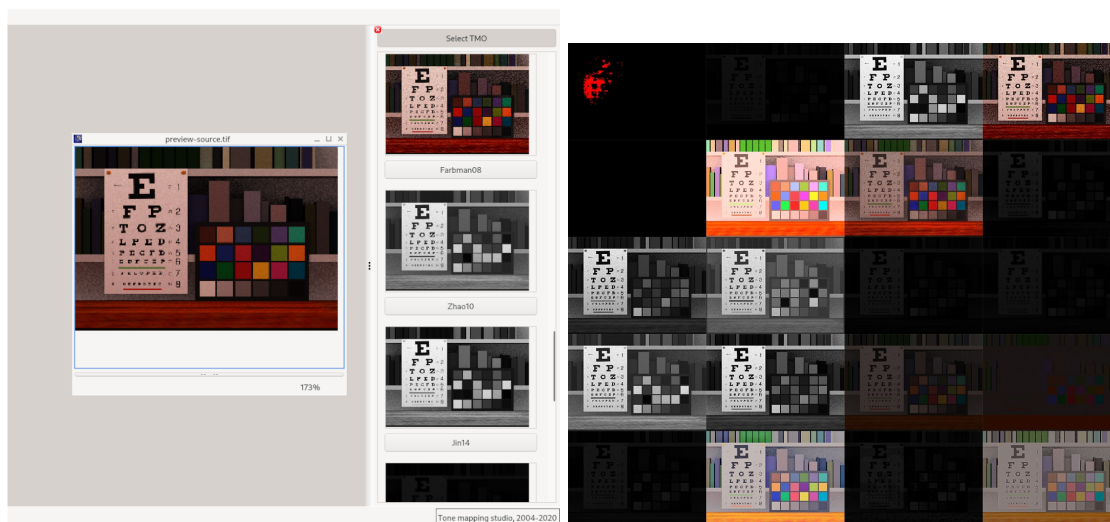
ulehčí orientaci v programu. Většině uživatelů trvalo splnění úkolů kratší dobu. Pro zobrazení analytických prvků jsem umístil výrazný přepínač na horní lištu. Většina testovaných neměla problém uvědomit si, k čemu tento přepínač slouží a že musí přepnout prostředí, aby mohli analytických prvků využít.

3.6.3 Ukázka výsledků aplikace operátorů

Pro ještě větší zjednodušení orientace bylo potřeba poskytnout uživateli možnost vidět výsledky jednotlivých operátorů před jejich samotným výběrem. Tato možnost je dostupná pouze ve zjednodušeném režimu, jelikož výpočet náhledů zbytečně snižuje výkon aplikace a pro pokročilejší uživatele není potřeba. Zde jsem ale narazil na problém. Vzhledem k tomu, že *TMS* je knihovna poskytující možnost implementovat vlastní operátory formou zásuvných modulů, tak není možné použít předem vytvořené náhledy. V případě přidání nového operátoru se musí náhled dynamicky dopočítat. To však znamená velkou výpočetní

zátěž a zpoždění při spuštění programu. Rozhodl jsem se tedy přistoupit k následujícímu řešení.

Program využívá jednoho HDR snímku, který má uložený ve své paměti, na který po prvním zapnutí programu aplikuje postupně všechny dostupné operátory. Výsledek si potom ukládá do lokální paměti. Zatížení programu je tedy omezeno pouze na první spuštění. Výpočty probíhají ve vlastních vláknech a uživatel má tedy možnost používat program i v průběhu kalkulace náhledů. V případě přidání nového operátoru se pak dopočítá náhled pouze pro něj. Komplikací je, že *TMS* obsahuje nejen operátory pro převod HDR obrázků na LDR, ale také operátory pro převod barevných LDR obrázků na stupně šedi a další, z nichž některé vyžadují na vstupu LDR soubor. Tato vlastnost však není nikde identifikována a program tedy neumí rozhodnout, zda pro vytvoření ukázky má poskytnout ten či onen vstup. *TMS* je zaměřeno hlavně na práci se snímky s vysokým dynamickým rozsahem. Proto jsem pro úvodní náhled zvolil jednu z HDR fotografií, která bohužel u některých operátorů nevytváří odpovídající výsledek. Dalším problémem je nejednotnost výstupu operátorů. Je snaha sjednotit výstup operátoru na hodnoty v rozmezí $\langle 0, 1 \rangle$, ale zatím tomu tak není a zobrazené výsledky tak mohou být zkreslené (viz obr 3.4).



Obrázek 3.4: Vlevo: Prvek s náhledy výsledků a zdrojový obrázek. Vpravo: Uložené vypočítané náhledy.

Tento prvek se při uživatelském testování ukázal, i přes ne příliš přesné zobrazení, jako největší pomocník v orientaci a pochopení programu. Několik testovaných osob však zmínilo, že by ještě více uvítaly, kdyby se náhledy zobrazovaly přímo pro právě vybraný snímek. Toto vylepšení jsem implementoval. Nyní se tedy v panelu náhledů zobrazují výsledky pro právě upravovaný snímek. To však přineslo další problém. Pokud aplikace některého z operátorů skončí segmentační chybou, tak celý program spadne. Stačí tedy jediný operátor, který nedokáže vybraný snímek zpracovat a uživatel nemá možnost použít ostatní. Toto lze obejít zapnutím pokročilého režimu, kde náhledy nejsou k dispozici a tedy se ani nevytvářejí.

3.7 Překlad a spuštění programu

Překlad a spuštění programu lze provést následujícími příkazy:

```
mkdir build; cd build
cmake ..
make
./tmogui
```

Pro překlad a spuštění programu je potřeba následujících knihoven:

- OpenExr <http://www.openexr.com/>
- LibTIFF <http://www.libtiff.org/>
- libjpeg <http://www.ijg.org/>
- FFTW <http://www.fftw.org/>
- Boost.MultiArray http://www.boost.org/doc/libs/1_56_0/libs/multi_array/doc/index.html
- CGAL <http://www.cgal.org/>
- OpenCV <https://opencv.org/>
- NLOpt <https://nlopt.readthedocs.io/>
- Qt5 <https://www.qt.io/offline-installers>

Pro Debian/Ubuntu jsou tyto knihovny obsaženy v balíčkách:

```
libopenexr-dev
libtiff4-dev
libjpeg-dev
qt5-default
qt5-doc
qttools5-dev-tools
libfftw3-3
libboost-all-dev
libopencv-dev
```

Pro Fedoru stačí spustit následující příkaz pro získání potřebných knihoven:

```
sudo dnf install OpenEXR OpenEXR-libs OpenEXR-devel \
libtiff libtiff-devel \
libjpeg-turbo libjpeg-turbo-devel libjpeg-turbo-utils \
fftw fftw-devel fftw-libs \
boost boost-devel \
CGAL-devel \
opencv opencv-devel \
nlopt nlopt-devel \
qt5 qt5-devel
```

3.8 Závěr

Vzhledem k velkému množství různých chyb, které se objevily po převodu rozhraní do *Qt5* byla implementace mnohem náročnější, než jen spuštění dvou skriptů pro automatický převod. Korektní zobrazování HDR snímků na LDR monitoru je složitá věc. Práci navíc komplikovala koronavirová krize, která značně ztížila hledání osob pro uživatelské testování.

Uživatelské rozhraní programu *Tone Mapping Studio* nyní poskytuje nástroje, které usnadňují práci s HDR obrázky a přibližuje tuto úžasnou technologii širšímu okruhu lidí. Díky uživatelskému testování se identifikovala slabá místa programu. Tyto nedostatky se posléze odstranily. Dalším možným vylepšením programu by mohlo být sjednocení formátu výstupu operátorů v rozsahu od nuly do jedné. Dále také oprava deklarací vstupních parametrů, které jsou u některých operátorů špatně zadány a tím pádem se nedají efektivně nastavit nebo zobrazit.

Kapitola 4

Uživatelská studie

Tone Mapping Studio v současnosti poskytuje dvacet dva operátorů dostupných v grafickém uživatelském rozhraní. Velká část z nich slouží pro převod barevných snímků na černobílé. V této kapitole se zaměřím na právě tyto operátory. Již bylo provedeno několik studií porovnávající operátory určené pro převod barevných snímků na černobílé. Tato studie se však zaměřuje na využití těchto operátorů na převod videa. Kvalita jejich výsledků je měřitelná dvojitým způsobem. Z jedné strany lze operátory porovnávat na základě podobnosti jejich výstupu se zdrojovým obrázkem. Z druhé je míra estetické líbivosti výstupu vůči jinému operátoru bez náhledu zdrojového snímku.[4]

V této kapitole nejdříve popíši proces převodu a možné chyby, které se mohou při převodu vyskytnout. Zanalyzuji již provedené studie zabývající se podobnou tematikou. Také popíši jednotlivé operátory, které byly v rámci studie porovnány. Dále uvedu způsob, jakým byla uživatelská studie realizována a na závěr shrnu výsledky této studie.

4.1 Teorie

V této části bych rád definoval některé pojmy vyskytující se v mé práci a obecně na převod barevných snímků na šedotónové. Dále pak popíšu specifika, které se objevují při převodu videa a chyby, které se mohou při převodu vyskytnout.

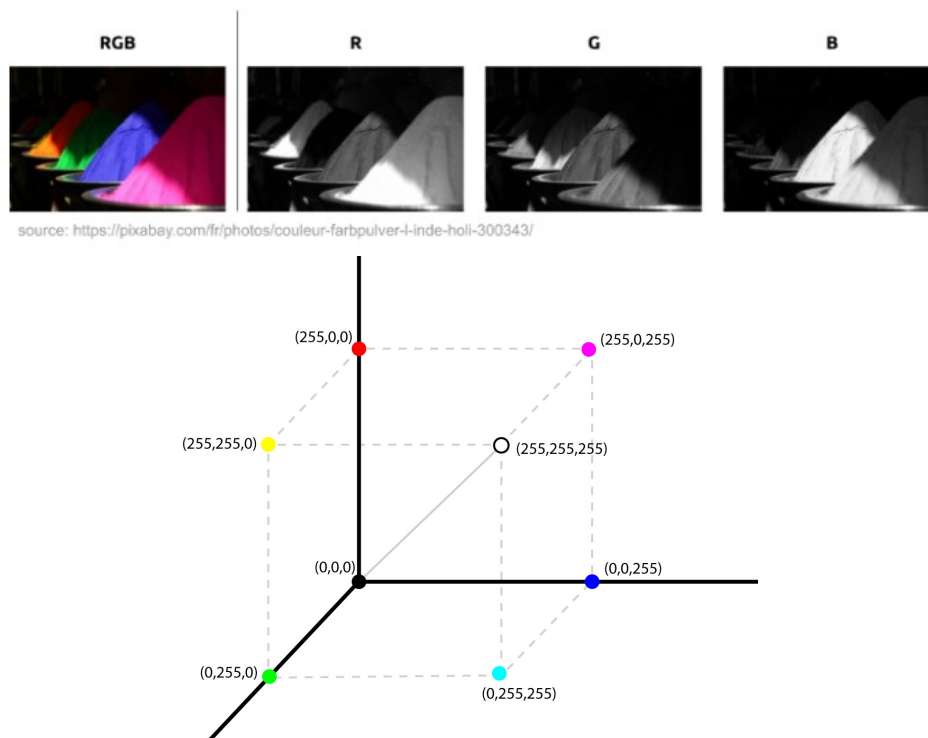
4.1.1 Barevné modely

Barva je výsledek našeho vnímání viditelného elektromagnetického záření v rozsahu $400nm$ - $700nm$. Lidské oko obsahuje tři druhy receptorů vnímajících barvu. Tudíž se pro uložení informace o barvě využívá tři numerických hodnot. Způsobů jak reprezentovat barvu je však nespočet. Každý takový způsob, který je definován, se označuje jako **barevný profil**. Barevné profily se mohou rozdělit na čtyři druhy:[16]

- **Kolometrické modely** - založeny na fyzickém měření v reálném světě
- **Psychofyzické modely** - založeny na lidském vnímání barev
- **Fyziologicky inspirované modely** - založeny na stavbě lidského oka
- **Oponentní modely** - založené na párových oponentních barvách (např. žlutá-modrá)

Zřejmě nejrozšířenějším barevným modelem je model **RGB**. Pracuje se třemi složkami v celočíselném rozsahu $< 0, 255 >$: červená (**Red**), zelená (**Green**) a modrá (**Blue**). Veškeré

barvy spektra jsou reprezentovány jako směs těchto tří základních barev. Celkem tedy dokáže rozlišit $256^3 = 16777216$ barev. Je obsažen například v JPEG a MPEJ standardu.[6]



Obrázek 4.1: Nahoře: Jednotlivé složky barevného prostoru RGB. Dole: Vizualizace barevného prostoru RGB.*

* Zdroje: <https://towardsdatascience.com>
https://commons.wikimedia.org/wiki/File:RGB_cube.jpg

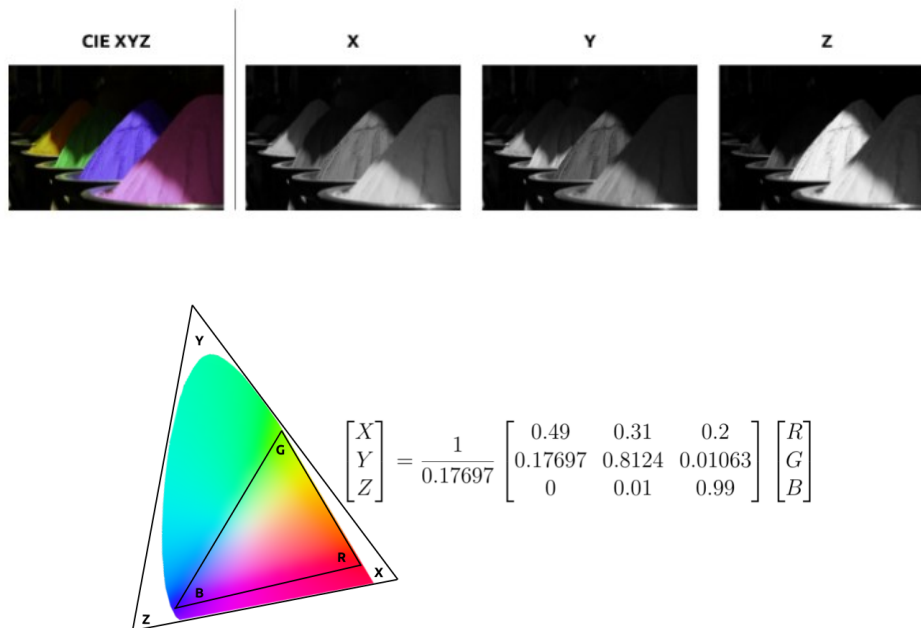
Jedním z nejzákladnějších barevných modelů je **CIE XYZ**, který byl vyvinutý *Mezinárodní komisí pro osvětlování* v roce 1931. Od modelu RGB se liší hlavně tím, že jeho Y složka reprezentuje jas. Složky X a Z lze popsat jako barevné souřadnice. Hodnoty těchto složek se pohybují ve spojitém rozmezí $< 0, 1 >$. RGB zabírá pouze 40% viditelného barevného prostoru. Porovnání rozsahu CIE XYZ s rozsahem RGB můžeme vidět na obrázku 4.2.[6]

Modely odvozenými od CIE XYZ jsou modely **CIE L*u*v*** a **CIE L*a*b***. Důvodem jejich vzniku byla snaha o lepší zachycení reálných percepčních hodnot. Složka L* v obou modelech reprezentuje jas v rozmezí $< 0, 100 >$. Hodnotu této složky definuje CIE následovně:[6]

$$L^* = \begin{cases} 116 \left(\frac{Y}{Y_n} \right)^{\frac{1}{3}} - 16 & \text{if } \frac{Y}{Y_n} > 0.00856 \\ 903.3 \left(\frac{Y}{Y_n} \right)^{\frac{1}{3}} & \text{if } \frac{Y}{Y_n} \leq 0.00856 \end{cases} \quad (4.1)$$

kde Y_n je fyzický jas bílého referenčního bodu.[6]

Posledním barevným modelem, který je v rámci této práce potřeba zmínit, je model **CIE L*C*h***. Složka L* reprezentuje jas podobně jako u předešlých modelů. Specifikem tohoto modelu je však složka C*, která reprezentuje sytost barvy v rozsahu $< 0, 100 >$ a složka h*, kterou můžeme vnímat jako úhel (nabývá hodnot $< 0, 360 >$).



Obrázek 4.2: Nahoře: Jednotlivé složky barevného prostoru XYZ. Dole: Porovnání prostoru XYZ s RGB.*

* Zdroj: <https://towardsdatascience.com>

4.1.2 Převod na stupně šedi

Převod barevného obrázku na černobílý je zahozením dvou dimenzí dat. Při převodu tedy zákonitě musíme přijít o informace. Cílem metod pro převod do odstínů šedi je tedy většinou využít data, která máme k dispozici, a vytvořit takový snímek, který v sobě bude obsahovat co nejvíce informací z původního barevného. Za informaci můžeme považovat například rozeznatelnost objektů na fotografii, nebo zachování kontrastu a to hlavně u snímků, které tyto informace při převodu obvyklým způsobem ztrácejí. Kromě komprimace informací může být motivací i estetický záměr.

Nejpřímočařejším přístupem je ponechání pouze jasové složky barevného profilu, což je velmi rychlé a v mnoha případech dostačující řešení. Za klasický přístup pro převod RGB snímku do stupňů šedi se dá označit zpracování každého pixelu následovně:

$$G_r(x, y) = \alpha R(x, y) + \beta G(x, y) + \gamma B(x, y) \quad (4.2)$$

kde α , β a γ jsou pozitivní váhy, které dávají v součtu jedna.

Váhy jsou většinou odvozeny z lidského vnímání a citlivosti oka na jednotlivé barvy. U snímků s malými rozdíly jasu je však naprosto nepoužitelná. Informace obsažené na snímku jsou často zahozeny, kontrast se ztrácí a například stíny nejsou rozeznatelné. Vyvíjí se tedy metody a algoritmy, které tyto informace dokáží zachovat.

Pro převod se ale také využívají vlastnosti jiných barevných modelů než RGB. Například lze využít barevných modelů, u kterých jedna ze složek přímo obsahuje informaci o jasu pixelu. Těmito modely mohou být například *CIE L*a*b**, *CIE L*C*h*, nebo *CIE Luv*.

4.1.3 Převod videa

Aplikace operátorů na videa přináší hned několik možností. Operátory určené pro převod snímků převádějí jednotlivé snímky jeden po druhém bez toho, aby braly v potaz předešlé či následující snímky. Tyto můžeme označit za *časově nezávislé operátory*. Některé operátory mohou být určené přímo pro převod videí a tím pádem využívat při převodu jednoho snímku i informace obsažené v ostatních. Tyto můžeme nazývat jako *časově závislé operátory*. V této studii je pro porovnání zařazena pouze jedna časově závislá metoda Kim09 (viz 4.3.4).

4.1.4 Možné chyby převodu

Při převodu videa se setkáváme s několika chybami, které se mohou negativně promítnout do kvality výsledku převodu. Převádění každého snímku zvlášť (časově nezávisle) tato rizika jen umocňuje. Jedním z problémů jsou **blikající artefakty**. Ty se dají rozdělit na lokální a globální. **Globální** blikající artefakty lze popsat jako náhlé změny celkového jasu snímku. Jsou většinou způsobeny operátory využívajícími parametry, které se adaptují v závislosti na obsahu snímku.[3] **Lokální** blikající artefakty jsou analogií toho samého problému v menší míře. Většinou jsou způsobeny lokálními operátory počítajícími s okolím pixelu[3].

Dalším problémem může být **temporální inkoherece**. Tím rozumíme nepřenesení určité informace z původního snímku. Například snímek může mít určitý celkový jas, který postupně slábne. Některé operátory mohou být nastaveny tak, aby udržovali hodnotu jasu na stále stejné úrovni. Tím se ale ztratí informace z původního videa. Stejně tak to může být i například s kontrastem.[3]

4.2 Analýza podobných studií

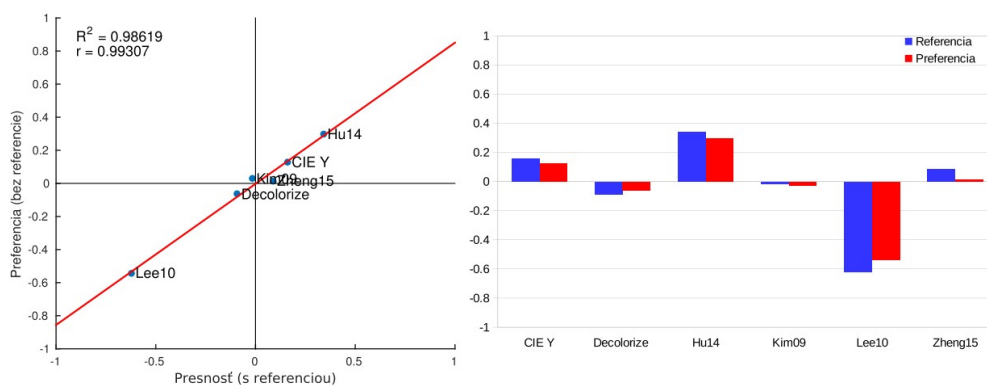
Vzhledem k malému počtu studií porovnávajících převod videa do šedotónu se zaměřím také na podobné studie, které se zabývají například porovnáváním tone mappingových operátorů pro převod HDR snímků, nebo metodami převodu statických snímků na šedotónové.

4.2.1 Porovnání časově závislých metod pro převod barevného obrazu na šedotónový (2018)

Za zmínku jistě stojí diplomová práce kolegy Vladimíra Vlkoviče: „*Porovnání časově závislých metod pro převod barevného obrazu na šedotónový*“ z roku 2018. Uživatelská studie porovnává tři časově závislé a tři časově nezávislé metody na dvaceti sedmi videích. Využívá metodu 2AFC (viz 4.4.2). Studie obsahuje dva typy testů. Prvním je test bez reference na zdrojový obrázek, který ukazuje subjektivní líbivost výsledku převodu (preferenci). Druhým testem je test s referencí původního barevného videa. Tento ukazuje přesnost převodu a podobnost s původním souborem.[19]

Z celkového počtu 7200 párových porovnání vzešlo hned několik zajímavých poznatků. Výsledky viz obrázek 4.3. Prvním bylo zjištění korelace mezi testy s referencí a bez reference. Přidání reference pouze lehce zvětšilo rozdíl mezi metodami, které si v testu bez reference vedly dobře a metodami, které byly slabší. Toto zjištění vedlo k mému rozhodnutí zabývat se v mé studii pouze testem s referencí.[19]

Dalším zjištěním bylo, že ačkoliv by se dalo očekávat, že časově závislé metody budou mít velkou výhodu nad časově nezávislými, tak tomu tak nebylo. Dalo by se říci, že pokud časově nezávislá metoda nevytváří artefakty a dodržuje temporální koherenci, tak se může bez problémů měřit s metodami časově závislými.[19]



Obrázek 4.3: Výsledky studie „Porovnání časově závislých metod pro převod barevného obrazu na šedotónový“. Vlevo: Porovnání výsledků obou částí studie. Vpravo: Celkové skóre zvlášť pro obě části studie.[19]

Také bylo zjištěno, že metoda *CIE Y*, která byla pouhým zanecháním jasové složky barevného prostoru, byla u některých videí velice konkurence schopná.[19] Jako referenci k této studii jsem do studie zařadil časově závislou metodu Kim09 (viz 4.3.4) a dvě časově nezávislé metody Zheng15 (viz 4.3.7) a Lee10 (viz 4.3.5).

4.2.2 Assessment of video tone-mapping (2013)

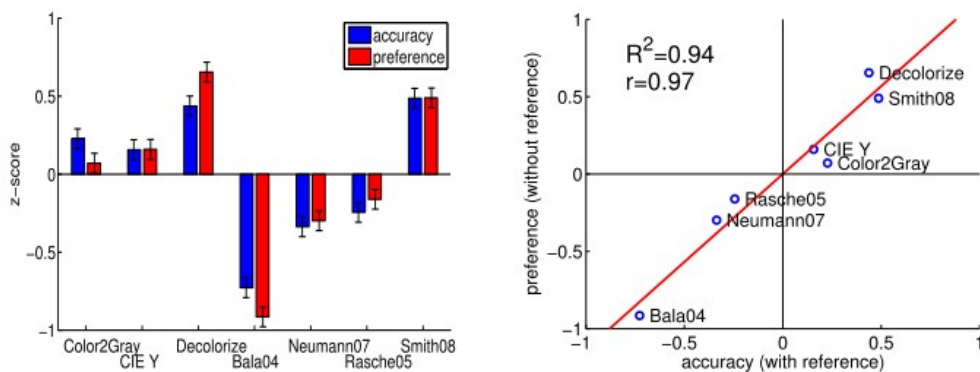
Jedna z prvních studií zabývající se převodem videa. Rafal Mantiuk a Josselin Petit ve své práci: „*Assessment of video tone-mapping: Are cameras' S-shaped tone-curves good enough?*“ využívají HDR videí pro porovnání tone mappingových operátorů. Využívají metody hodnocení, kdy u každého převedeného videa má testovaný subjekt přiřadit skóre z jedenácti hodnot. Párové porovnávání bylo zváženo, ale vzhledem k časové náročnosti se nevyužilo, jelikož by u každého videa muselo být 190 porovnání. Studie se zúčastnilo deset lidí, rozdělených do dvou skupin po pěti, v kontrolovaném prostředí.[15]

Výsledkem studie bylo zjištění, že není moc velký rozdíl mezi třemi nejlepšími operátory. Také že fixní S-křivka používaná ve fotoaparátech není nejlepším řešením pro složitější scény. V rámci studie se mezi sebou porovnávaly i stejné operátory s různým nastavením parametrů. Bylo zjištěno, že automatické nalezení parametrů operátorů nebo využití defaultních parametrů může být při studiích tohoto typu problém a může to zkreslovat výsledky.[15]

4.2.3 Perceptual Evaluation of Color-to-Grayscale Image Conversions (2008)

Na závěr bych rád zmínil práci doc. Čadíka: „*Perceptual Evaluation of Color-to-Grayscale Image Conversions*“. Tato studie porovnává sedm metod na převod barevných snímků do stupňů šedi na dvaceti čtyřech obrázcích. Využívá metodu *2AFC* (viz 4.4.2). Tímto způsobem jsou mezi sebou porovnány obrázky každý s každým (celkem 504 porovnání pro 7 operátorů). Jelikož množství porovnání nebylo vhodné pro jednoho účastníka testu, byl dataset rozdělen do tří skupin po osmi obrázcích (to je 168 porovnání na jedno sezení).[4]

Bylo provedeno porovnání jak s referencí, tak bez reference (stejně jako 4.2.1). Výsledkem tedy bylo ohodnocení přesnosti a preference výstupu operátoru. Na studii se podílelo 121 účastníků. Byly rozděleny na poloviny a každá podstoupila jednu ze dvou verzí testu.[4]



Obrázek 4.4: Výsledky studie „*Perceptual Evaluation of Color-to-Grayscale Image Conversions*“. Vlevo: Celkové skóre zvlášť pro obě části studie. Vpravo: Porovnání výsledků obou částí studie.[4]

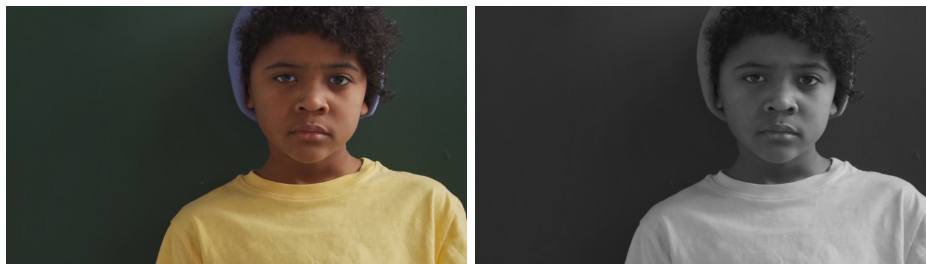
Byla nalezena korelace mezi testem s referencí a bez reference. Výsledky studie viz obrázek 4.4.

4.3 Operátory knihovny TMO

Pro uživatelskou studii jsem vybral následujících osm operátorů, které jsou implementovány v rámci knihovny *TMS*. Sedm operátorů je časově nezávislých. Do této sady jsem přidal časově závislou metodu Kim09 (4.3.4) pro porovnání jak s časově nezávislými metodami, tak s již provedenou studií časově závislých metod (viz 4.2.1).

4.3.1 Contrast Enhancing Colour to Grey (2009)

Tato část čerpá z článku *Contrast Enhancing Colour to Grey*[1] z roku 2009. Tuto metodu budu dále označovat jako *Alsam09*.



Obrázek 4.5: Šedotónový snímek po aplikaci operátoru *Alsam09*.

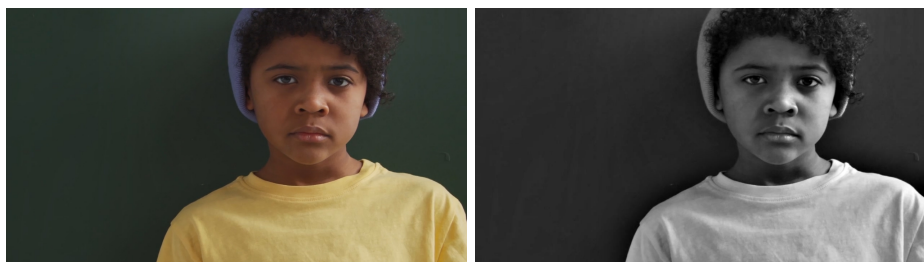
Jedná se o lokální prostorovou metodu, která je rychlá a jejím výsledkem je zvýšení lokálního a globálního kontrastu snímku. Na rozdíl od klasické rovnice pro převod RGB snímku (4.2), kde jsou váhy skaláry, *Alsam09* vypočítává tyto váhy pro každý pixel s ohledem na kontrast v $n \times n$ okolí pixelu tak, aby byl kontrast zachován. Každá váha je tedy vektor reprezentující váhovou mapu pro jednotlivé RGB kanály. Tyto váhové mapy označuje jako $\alpha(x, y)$, $\beta(x, y)$, $\gamma(x, y)$ Výsledný vzorec pro převod barevného snímku tedy vypadá následovně:

$$G_r(x, y) = \alpha(x, y)R(x, y) + \beta(x, y)G(x, y) + \gamma(x, y)B(x, y), \quad (4.3)$$

kde G_r je výsledný stupeň šedé pro daný pixel a $R(x, y)$, $G(x, y)$, $B(x, y)$ jsou jednotlivé složky RGB prostoru daného pixelu.

4.3.2 Laplacian-Guided Image Decolorization (2016)

Tato část čerpá z článku *Laplacian-Guided Image Decolorization*[2] z roku 2016. Tuto metodu budu dále označovat jako `Ancuti16`.



Obrázek 4.6: Šedotónový snímek po aplikaci operátoru `Ancuti16`.

Tato nelineární metoda využívá spojení hned dvou typů váhových map pro každou barevnou složku, které jsou před samotným převodem vypočítány ze zdrojového snímku. Prvním typem je **Laplaceova váhová mapa**, která využívá Laplaceův operátor pro identifikaci míst s vysokým lokálním kontrastem. Tato mapa se počítá podle vzorce (4.4):

$$W_L^k(x, y) = \overline{L\{I^k(x, y)\}} + \left| L\{I^k(x, y)\} \right|, \quad (4.4)$$

kde k reprezentuje index vstupních kanálů (R, G, B).

Tato mapa však není dostačující, jelikož nebere v potaz celkový vzhled originálního snímku. Právě k tomu slouží druhý typ mapy, kterou můžeme označit jako **váhu globálního kontrastu**. Tuto váhovou mapu lze vyjádřit jako vzorec (4.5):

$$W_G^k(x, y) = \left[I^k(x, y) - \overline{L\{I^k(x, y)\}} \right]^\delta \quad (4.5)$$

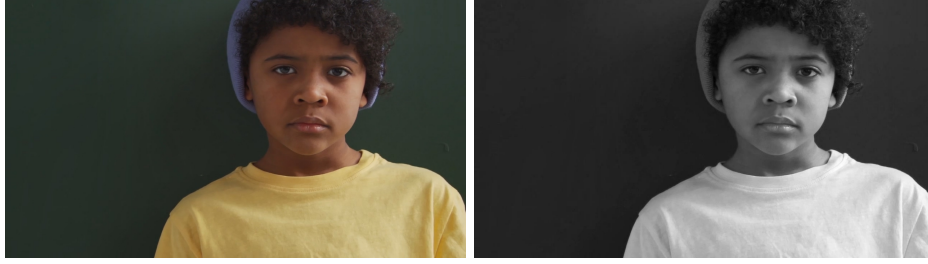
Tyto dvě váhové mapy se posléze spojí dohromady pomocí tzv. *multi-scale fusion* s jednotlivými kanály zdrojového snímku.

V rámci studie byla nastavena defaultní hodnota $\delta = 2$, která je napevno implementována v zásuvném modulu *TMS*.

4.3.3 Variance Maximization for Decolorization (2014)

Tato část čerpá z článku *A Variational Approach for Image Decolorization by Variance Maximization*[8] z roku 2014. Tuto metodu budu dále označovat jako `Jin14`.

Pro zachování detailů kontrastu a co nejvíce detailů zdrojového snímku vypočítává tato metoda váhovou mapu s parametry pro převod jednotlivých pixelů snímku pro každý z barevných kanálů. Podobně jako metoda `Alsam09` (4.3.1) bere při výpočtu v potaz okolí pixelu a hledá vhodné váhy α , β a γ pro rovnici (4.2). Vektor těchto vah je zde označen jako



Obrázek 4.7: Šedotónový snímek po aplikaci operátoru Jin14.

$\mathbf{u}(\mathbf{x})$. Z hodnot okolních pixelů získává metoda vektor středních hodnot $\mathbf{m}(\mathbf{x})$ a kovarianční matici $\mathbf{K}(\mathbf{x})$, jež je následně využita pro výpočet lokální odchylky podle vzorce:

$$\begin{aligned} \mathbf{Var}_\omega(L(x)) &= \mathbf{Var}_\omega(\alpha(x)R(x) + \beta(x)G(x) + \gamma(x)B(x)) \\ &= \alpha^2(x)k_{r,r}(x) + \beta^2(x)k_{g,g}(x) + \gamma^2(x)k_{b,b}(x) \\ &\quad + 2\alpha(x)\beta(x)k_{r,g}(x) + 2\alpha(x)\gamma(x)k_{r,b}(x) + 2\beta(x)\gamma(x)k_{g,b}(x) \\ &= \mathbf{u}(x)\mathbf{K}(x)\mathbf{u}^T(x), \end{aligned} \quad (4.6)$$

kde ω reprezentuje zvolenou velikost okolí pixelu.

Pro zvolené pozitivní konstanty τ , λ a μ vypadá výsledná rovnice pro převod následovně:

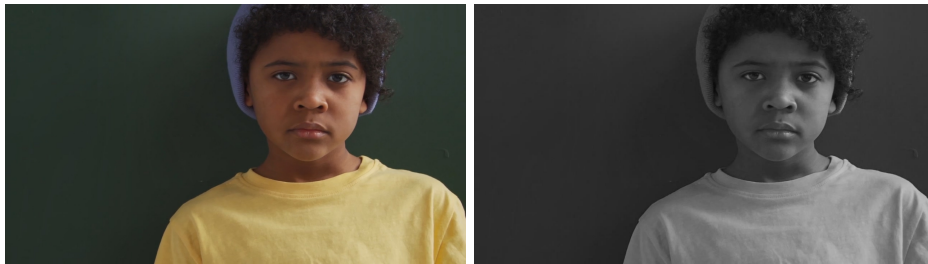
$$\begin{aligned} \min_{\mathbf{u} \in \Lambda} \left(\int_{\Omega} |D\mathbf{u}(x)| dx - \frac{\tau}{2} \int_{\Omega} \mathbf{u}(x)\mathbf{K}(x)\mathbf{u}(x)^T dx \right. \\ \left. + \frac{\lambda}{2} \int_{\Omega} (\mathbf{m}(x)\mathbf{u}(x)^T - 1)^2 dx + \frac{\mu}{2} \int_{\Omega} |\mathbf{u}(x)|^2 dx \right), \end{aligned} \quad (4.7)$$

kde

$$\begin{aligned} \Lambda := \{ \mathbf{u}(x) = (\alpha(x), \beta(x), \gamma(x)) \mid \mathbf{u}(x) \in BV(\Omega; \mathbb{R}^3), \\ \alpha(x) + \beta(x) + \gamma(x) = 1, 0 \leq \alpha(x), \beta(x), \gamma(x) \leq 1 \} \end{aligned} \quad (4.8)$$

4.3.4 Robust Color-to-gray via Nonlinear Global Mapping (2009)

Publikováno v rámci *SIGGRAPH ASIA 2009* [9]. Tuto metodu budu nadále označovat jako Kim09.



Obrázek 4.8: Šedotónový snímek po aplikaci operátoru Kim09.

Jedná se o rychlou a robustní globální metodu. Jejím cílem je zachování rozlišitelnosti významných rysů snímku a světlosti jednotlivých bodů. Výhodou této metody proti ostatním zahrnutým metodám v uživatelské studii je fakt, že je tato metoda rozšířena o možnost zpracování videa.

Metoda pracuje s barevným prostorem $CIE L^*C^*h$. Pro převod využívá globální mapovací funkci (4.9).

$$g(x, y) = L + f(\theta)C, \quad (4.9)$$

kde g je výsledný odstín pixelu se souřadnicemi (x, y) a L (*lightness - světlost*), C (*chroma - sytost*) a $\theta = H$ (*hue - odstín*) odpovídají jednotlivým složkám modelu $CIE L^*C^*h^*$.

V rámci této metody je funkce $f(\theta)$ definována následovně:

$$f(\theta) = \sum_{k=1}^n (A_k \cos(k\theta) + B_k \sin(k\theta)) + A_0, \quad (4.10)$$

kde A_k , B_k a A_0 jsou neznámé, které si tato metoda klade za cíl optimalizovat.

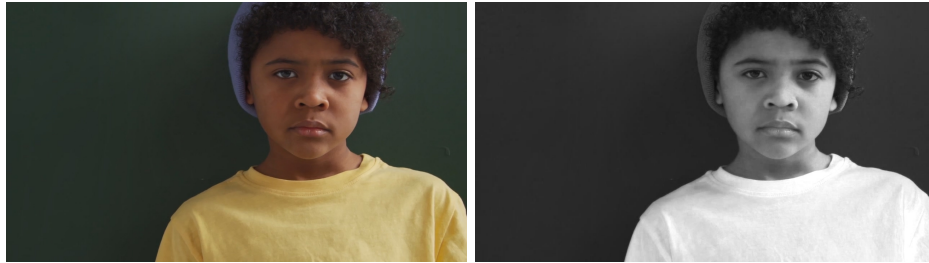
K optimalizaci těchto parametrů využívá hodnot z barevného prostoru $CIE L^*a^*b^*$ a bere v potaz vzdálenost jednotlivých barev v gradientu. Minimalizací rozdílů mezi gradientem původního snímku a snímku šedotónového se dosáhne požadovaného výsledku. Pro převod videí se energie vypočítá podle vzorce (4.11), který je vytvořen tak, aby se minimalizovala temporální koherence.

$$E_t = \{(g_c(x, y) - g_p(x, y)) - (c_c(x, y) \ominus c_p(x, y))\}^2, \quad (4.11)$$

kde $g(x, y)$ je pixel v šedotónovém obraze, $c(x, y)$ je pixel v původním barevném snímku a dolní indexy c a p určují aktuální a předchozí snímek videa.

4.3.5 Reducing Dimensions (2010)

Tato část čerpá z článku *Converting color images to grayscale images by reducing dimensions*[12] z roku 2010. Tuto metodu budu dále označovat jako **Lee10**.



Obrázek 4.9: Šedotónový snímek po aplikaci operátoru Lee10.

Při vyvíjení metody bylo využito znalostí o lidském vnímání barev. Načervenalejší barvy jsou vnímány jako teplejší a dominantnější než modravé. Teplejší barvy jsou podvědomě spojovány s lehčími objekty. Lidem přijdou světlejší objekty lehčí. Spojením těchto pozorování dostáváme jasný indikátor toho, že při převodu barev do stupňů šedi by červenější barvy měly být převedeny na světlejší odstín, než barvy spíše modré. Tento princip je celkem jasně vidět na obrázku 4.9, kde žluté tričko je výrazně světlejší, než modrá čepice, nebo zelená zeď. Tato metoda využívá barevného prostoru YCB_C_R , který nejlépe zastupuje výše zmíněné vlastnosti barev. Tento prostor lze získat převodem z RGB následovně:

$$\begin{aligned} Y &= 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B, \\ C_B &= -0.169 \cdot R - 0.331 \cdot G + 0.500 \cdot B + 128, \\ C_R &= 0.500 \cdot R - 0.419 \cdot G - 0.081 \cdot B + 128, \end{aligned} \quad (4.12)$$

kde Y vyjadřuje jasovou složku, C_B vyjadřuje celkovou modrost a C_R celkovou červenost pixelu.

Výsledek převodu touto metodou se získá pomocí vzorce (4.13).

$$P = Y + k \cdot \operatorname{sgn}(C_R - C_B) \cdot |C_R - C_B|^\alpha, \quad (4.13)$$

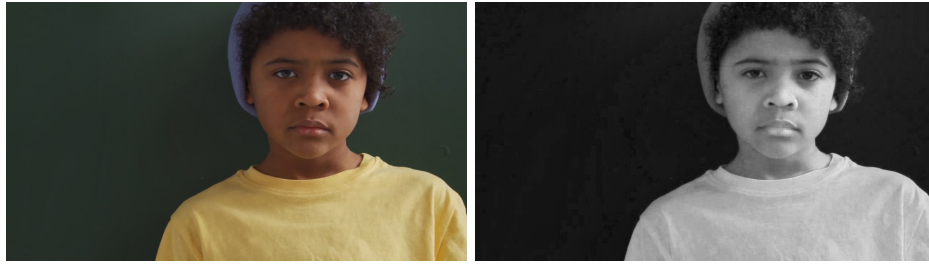
kde k je volitelný parametr upravující kontrast a α je volitelný parametr rozsahu.

Tato metoda byla vyvinuta tak, aby splnila následující 4 požadavky:

1. Zachovat barevný kontrast snímku
2. Zachovat rozdíly v jasu snímku
3. Pixely stejné barvy převést na stejný odstín šedé
4. Složitost algoritmu jednoduchá stejně jako prostá lineární kombinace barevných složek

4.3.6 Spectral Image Decolorization (2010)

Tato část čerpá z článku *Spectral Image Decolorization*[20] z roku 2010. Tuto metodu budu dále označovat jako Zhao10.



Obrázek 4.10: Šedotónový snímek po aplikaci operátoru Zhao10.

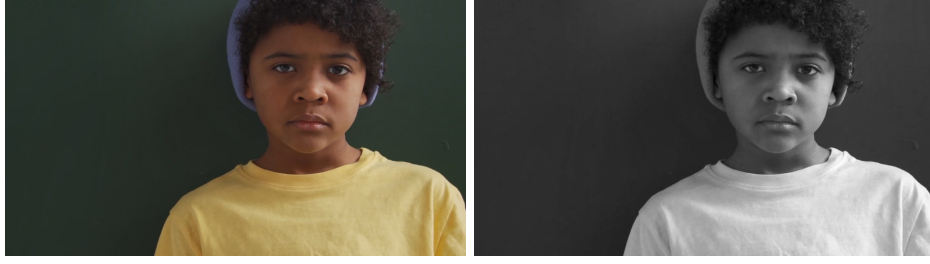
Metoda, která pracuje s Fourierovou transformací nad složkami barevného prostoru $CIEL^*a^*b^*$. Transformací jednotlivých složek (L,a,b) získáváme spektrální snímky \hat{L} , \hat{a} a \hat{b} . Výsledný snímek je pak získán inverzní Fourierovou transformací jasové složky \hat{L} , která je modifikována pro zvýšení kontrastu. Vzorec pro výpočet šedotónového snímku \bar{G}_I tedy vypadá následovně:

$$\begin{aligned} \bar{G}_I &= IF(\hat{E}), \\ \hat{E} &= H(\hat{L}, \hat{a}, \hat{b}), \\ H(\hat{L}, \hat{a}, \hat{b}) &= (1 - \theta)\hat{L} + \theta(\phi\hat{a} + (1 - \phi)\hat{b}) \end{aligned} \quad (4.14)$$

kde ω a ϕ jsou volitelné parametry, které lze dopočítat i automaticky.

4.3.7 Efficient Color-to-Gray Conversion for Digital Images in Gradinet Domain (2015)

Tato část čerpá z článku *Efficient Color-to-Gray Conversion for Digital Images in Gradinet Domain*[21] z roku 2015. Tuto metodu budu dále označovat jako Zheng15.



Obrázek 4.11: Šedotónový snímek po aplikaci operátoru Zheng15.

Jedná se o metodu, která cílí na shodu intenzit výstupu s původním barevným snímkem tak, aby kopírovala vlastnosti vnímání lidského oka. Zaměřuje se také na to, aby isoluminantní složky snímku zůstaly i po převodu rozeznatelné. Pracuje v barevném modelu $CIE L^*a^*b^*$. Myšlenkou je možnost reprezentace šedotónového snímku pomocí gradientu, jako například v rovnici (4.15).

$$\nabla I = (I_x, I_y) = (I(x + \Delta x, y) - I(x, y), I(x, y + \Delta y) - I(x, y)), \tilde{\nabla}_C(a, b) = (C_x, C_y), \quad (4.15)$$

Ze zdrojových dat získává gradientní pole $\tilde{\nabla}_C$, které se skládá z jasové složky ∇L a barevné složky $\tilde{\nabla}_C(a, b)$.

$$\begin{aligned} \tilde{\nabla}_C(a, b) &= (C_x, C_y), \\ C_x &= \sqrt{(a(x + \Delta x, y) - a(x, y))^2 + (b(x + \Delta x, y) - b(x, y))^2}, \\ C_y &= \sqrt{(a(x, y + \Delta y) - a(x, y))^2 + (b(x, y + \Delta y) - b(x, y))^2}, \end{aligned} \quad (4.16)$$

Z těchto komponent se posléze vypočítá gradientní pole snímku:

$$\begin{aligned} \tilde{\nabla}C &= \\ &\left(\text{sign}(L_x, a(x + \Delta x, y), a(x, y), b(x + \Delta x, y), b(x, y)) \cdot \sqrt{L_x^2 + A^2(C_x)}, \right. \\ &\left. \text{sign}(L_y, a(x, y + \Delta y), a(x, y), b(x, y + \Delta y), b(x, y)) \cdot \sqrt{L_y^2 + A^2(C_y)} \right) \end{aligned} \quad (4.17)$$

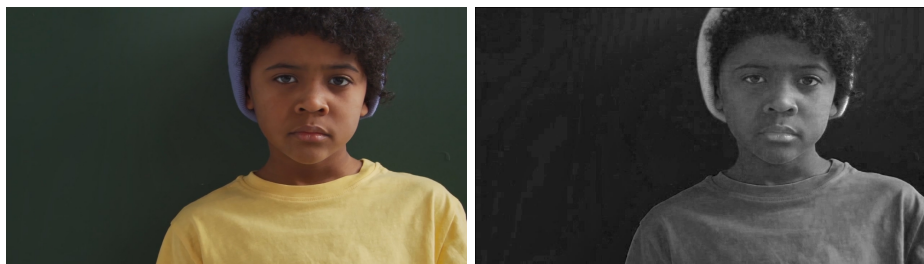
To je poté předloženo PES solveru (*Poisson Equation Solver*), který vypočítá výslednou podobu výstupu.

4.3.8 Efficient decolorization preserving dominant distinctions (2015)

Tato část čerpá z článku *Efficient decolorization preserving dominant distinctions*[7] z roku 2015. Tuto metodu budu dále označovat jako **Zhongping15**.

Metoda využívající barevného profilu $CIE L^*a^*b^*$. Nejdříve je nalezen barevný směr (*chromatic orientation*), který vyjadřuje informace o kontrastu snímku v rovině a^*b^* . Každý pixel je převeden do roviny $\Delta a^*\Delta b^*$. Pomocí vzorce (4.18) je s ohledem na dvojice osmi sousedních pixelů vypočítán hledaný směr.

$$G_p = \sum_{\langle i, j \rangle \in \mathcal{N}_p} s_{ij} \mathbf{g}(\Delta l_{ij}) (\Delta a_{ij}, \Delta b_{ij}), \quad (4.18)$$



Obrázek 4.12: Šedotónový snímek po aplikaci operátoru Zhongping15.

kde \mathcal{N}_p je množina čtyř dvojic sousedních osmi pixelů pixelu p , $(\Delta a_{ij}, \Delta b_{ij})$ označuje rozdíl barev v rovině $a*b*$ pro pixely $\langle i, j \rangle$, s_{ij} můžeme popsat jako polaritu barevného rozdílu (nabývá hodnot 1 a -1). G_p označuje barevný gradient pixelu p . Funkce $g(x) = 1 + x^2$ je váhovou funkcí.

Celková barevná orientace snímku je pak vypočítána jednoduše jako:

$$(O_a, O_b) = \sum_p G_p / \left| \sum_p G_p \right|, \quad (4.19)$$

Dále je aplikován filtr pásmové propusti $\mathcal{L}_{s1,s2}(x, y)$ (*Band-pass filter*), který odhalí informace o jasů. Veškeré získané informace jsou posléze využity při samotném převodu.

4.4 Plán uživatelské studie

Při plánování uživatelské studie jsem se inspiroval hlavně prací doc. Čadíka: „*Perceptual Evaluation of Color-to-Grayscale Image Conversions*“ [4]. Nejdříve bylo zapotřebí vytvořit dataset zdrojových videí, dále se rozhodnout pro metodiku testování, samotné provedení uživatelské studie a nakonec vyhodnocení výsledků.

4.4.1 Dataset

Pro vytvoření datasetu videí jsem využil stránky www.pexels.com, kde je velké množství videí s otevřenou licencí. Odkazy na videa stažené z této stránky jsou uvedeny v příloze E. Celkem bylo v uživatelské studii použito 16 videí. Při výběru videí jsem dbal na různorodost. Co se obsahu týče, dataset obsahuje videa s portréty lidí, záběry zvířat, abstrakci, animaci, rychle se pohybující objekty a statické objekty, záběry z blízka i z dálky. Pro otestování všech aspektů metody převodu do stupňů šedi byla zároveň potřeba různorodost v barevnosti. Video tedy obsahují výrazně barevné prvky, světlé i tmavé oblasti, jemné barevné detaily, míchání barev, barvy se stejnou hodnotou jasů atp. (viz obrázek 4.13) Ukázky všech vybraných a ve studii použitých videí jsou v příloze C spolu s náhledem převedených šedotónových variant.

Dalším aspektem byla délka videa. Pokud by videa byla příliš dlouhá, tak by bylo pro účastníka experimentu unavující projít jím až do konce, nebo by se účastník rozhodl dříve, než by viděl celé video. Video byla tedy zkrácena tak, aby celkový průchod experimentem byl kratší. Průměrná délka videa datasetu je 7 sekund. Většina videí má délku 5 sekund, přičemž nejdelší video má 28 sekund a nejkratší 4 sekundy. V jednom průchodu se porovnává 8 metod každá s každou, to je celkem 28 porovnání. Celkový čas testu se tedy dá odhadovat na $28 * 7 / 60 = 3.3 \text{minuty}$ (pokud by uživatel rozhodl v okamžiku konce videa a okamžitě přešel



Obrázek 4.13: Ukázka snímků z některých videí datasetu.

na další porovnání). Reálně nelze předpokládat takto rychlý průchod z důvodu načítání videí, nerozhodnosti uživatele a dalších faktorů.

Rozlišení videí jsem zvolil HD (720p). Toto rozlišení poskytuje dostatečnou kvalitu pro hodnocení. Vzhledem k časové náročnosti převodu do stupňů šedi a způsobu doručení experimentu uživatelům nebylo vyšší rozlišení vhodné. Načítání webové stránky s experimentem by trvalo příliš dlouho a uživatelé s pomalejším internetem by měli problém experiment dokončit.

4.4.2 Metodika uživatelské studie

Jako metodiku experimentu jsem zvolil nucený výběr ze dvou možností metodou *2AFC* s referencí původního barevného snímku. Z dalších variant se nabízela například metoda stupnice použitá v práci „*Assessment of video tone-mapping*“ (viz 4.2.2), která poskytne uživateli jeden snímek pro ohodnocení a ten posléze ohodnotí kvalitu převodu na přiložené stupnici. Výhodou této metody je menší časová náročnost. Pokud bychom chtěli v našem datasetu ohodnotit všechny metody na všech videích, celkově by se jednalo o $16 * 8 = 128$ ohodnocení. Pokud bychom chtěli toho samého dosáhnout metodou *2AFC*, pak by se jednalo o $128 * 127/2 = 8128$ porovnání. Pro metodu *2AFC* jsem se rozhodl, jelikož v

předchozích studiích vykazovala velmi dobré výsledky a je pro uživatele přijatelnější. Bylo však potřeba nějakým způsobem snížit počet porovnání na přijatelnou úroveň. Toho jsem docílil zaměřením se na porovnání operátorů. Při počtu osmi operátorů se počet porovnání všech při jednom průchodu rovná $8 * 7/2 = 28$ porovnání. Pro každý průchod se videa náhodně seřadily a každý uživatel tedy porovnával dva určité operátory aplikované na jiné video.

Metoda *2AFC* byla popsána Herbertem Aronem Davidem v práci: „*The method of paired comparisons*“.[5] Její celý název zní ***two-alternative forced choice***. Šedotónové verze videí jsou uživateli prezentovány po dvojicích a jeho úkolem je jednotlivá videa porovnat a vybrat to, které považuje za přesnější reprezentaci zdrojového snímku. Konkrétně byl uživatel instruován větou „*Klikněte prosím na černobílé video, které považujete za přesnější reprezentaci barevného videa.*“, respektive „*Please click on grayscale video that you find as a better representation of the colourful one.*“ v angličtině.

Uživatel měl k dispozici také původní barevné video pro porovnání. Pro tuto verzi experimentu jsem se rozhodl na základě výsledků předešlých studií (viz 4.2), kde se potvrdila korelace mezi výsledky s referencí a bez reference. Referenční video bylo zobrazeno ve středu nad porovnávanými šedotónovými videi (viz obrázek 4.14). Pozice porovnávaných videí se určovala náhodně. 4.14



Obrázek 4.14: Ukázka rozložení prvků uživatelské studie. Referenční barevné video nahoře uprostřed, porovnávaná pod ním.

4.4.3 Vyhodnocování preferencí

Pro uložení preferencí jednotlivých účastníků experimentu byla využita databáze, do které se ukládaly informace o tom, jaký operátor byl preferovaný vůči kterému a na jakém videu bylo porovnání provedeno. Z těchto dat lze posléze vytvořit **preferenční matice**. V našem případě má matice rozměry 8×8 , jelikož porovnáváme 8 operátorů.[5]

Každý řádek matice reprezentuje pozorovaný operátor. Sloupce matice odkazují na operátor, se kterým byl porovnáván. Za každou preferenci je do matice na odpovídající místo přičten jeden bod. Například pokud účastník experimentu preferoval operátor A nad operátorem B, pak na řádek odpovídající operátoru A se do sloupce operátoru B přičte 1. Jelikož nemá smysl porovnávat operátory samy se sebou, tak diagonála této matice zůstává nevyužitá. Preferenční matici lze zkonstruovat pro jednotlivá videa, nebo pro jednotlivé uživatele. Zároveň lze vytvořit preferenční matici napříč všemi uživateli a videi. Náhled preferenčních matic v tabulkách 4.1 a 4.2.

| | Als | Anc | Jin | Kim | Lee | Zha | Zhe | Zho | Skóre |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-------|
| Alsam09 | - | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 5 |
| Ancuti16 | 1 | - | 0 | 1 | 0 | 1 | 0 | 0 | 3 |
| Jin14 | 0 | 1 | - | 0 | 1 | 1 | 0 | 1 | 4 |
| Kim09 | 1 | 0 | 1 | - | 1 | 0 | 1 | 1 | 5 |
| Lee10 | 0 | 1 | 0 | 0 | - | 1 | 0 | 0 | 2 |
| Zhao10 | 0 | 0 | 0 | 1 | 0 | - | 1 | 1 | 3 |
| Zheng15 | 0 | 1 | 1 | 0 | 1 | 0 | - | 1 | 4 |
| Zhongping15 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | - | 2 |

Tabulka 4.1: Preferenční matice jednoho účastníka experimentu

| Colours | Als | Anc | Jin | Kim | Lee | Zha | Zhe | Zho | Preference |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|------------|
| Alsam09 | - | 4 | 4 | 10 | 9 | 12 | 3 | 8 | 71,43% |
| Ancuti16 | 1 | - | 3 | 4 | 7 | 10 | 3 | 6 | 48,57% |
| Jin14 | 7 | 10 | - | 6 | 7 | 8 | 4 | 9 | 65,38% |
| Kim09 | 4 | 5 | 7 | - | 8 | 11 | 3 | 7 | 56,96% |
| Lee10 | 1 | 3 | 1 | 1 | - | 4 | 1 | 6 | 21,52% |
| Zhao10 | 0 | 2 | 5 | 4 | 12 | - | 3 | 5 | 34,07% |
| Zheng15 | 5 | 7 | 5 | 7 | 10 | 8 | - | 9 | 72,86% |
| Zhongping15 | 2 | 5 | 2 | 2 | 9 | 7 | 2 | - | 36,71% |

Tabulka 4.2: Preferenční matice pro video *Colours*.

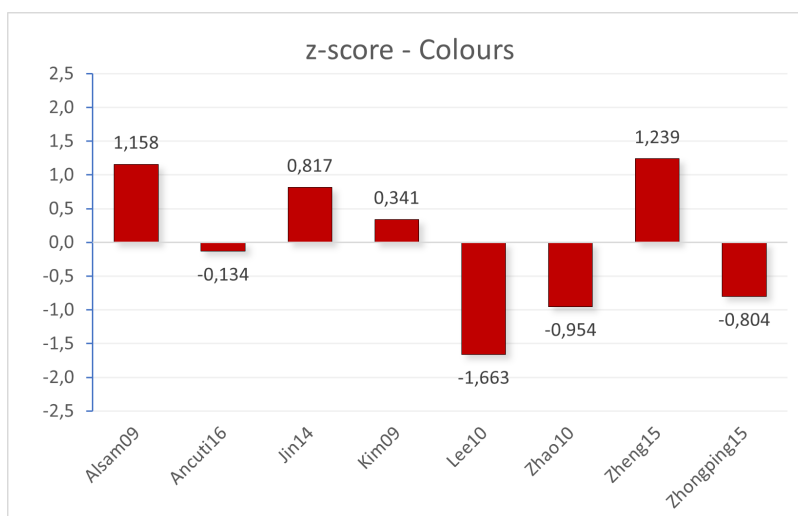
Vzhledem ke způsobu provedení experimentu se nedalo počítat přímo se skóre vypočítaném součtem všech hodnot jedné řádky preferenční matice. Skóre by odpovídalo výsledku pouze pokud by se počet porovnání každého jednoho operátoru rovnal. Jako rozhodující skóre jsem tedy použil **procenta preference** vypočítané vydělením skóre počtem porovnání daného operátoru (viz tabulka 4.3).

Pro lepší reprezentaci výsledků v grafu byla tato procenta standardizována převedením na *z-score*. **Z-score** představuje metriku, která ukazuje kolik směrodatných odchylek je skóre vzdáleno od průměru (viz obrázek 4.15), kde hodnota 0 znamená, že se jedná o průměrný výsledek. Tento způsob nám umožní porovnávat data z různých studií a experimentů mezi sebou.

Data bylo také potřeba statisticky vyhodnotit. Pro zjištění shody mezi účastníky experimentu jsem využil **Kendallův test**. Koeficient u vypočítaný podle vzorce 4.20 se pohybuje v rozmezí $< 0, 1 >$, kde hodnota 1 odpovídá případu, kdy by všichni účastníci experimentu odpovídali naprosto stejně. Hodnota blíží se nule by pak znamenala naprosto žádnou shodu mezi účastníky [11]

| Operátor | Porovnání | Skóre | Preference | z-score |
|-------------|-----------|-------|------------|---------|
| Alsam09 | 70 | 50 | 71,43% | 1,158 |
| Ancuti16 | 70 | 34 | 48,57% | -0,134 |
| Jin14 | 78 | 51 | 65,38% | 0,817 |
| Kim09 | 79 | 45 | 56,96% | 0,341 |
| Lee10 | 79 | 17 | 21,52% | -1,663 |
| Zhao10 | 91 | 31 | 34,07% | -0,954 |
| Zheng15 | 70 | 51 | 72,86% | 1,239 |
| Zhongping15 | 79 | 29 | 36,71% | -0,804 |

Tabulka 4.3: Ukázka výpočtu výsledků pro video *Colours*.



Obrázek 4.15: Standardizované výsledky pro video *Colours*.

$$\begin{aligned}
 u &= \frac{2\Sigma}{\binom{t}{2}\binom{s}{2}} - 1, \\
 \Sigma &= \sum_{i \neq j} \binom{p_{ij}}{2},
 \end{aligned}
 \tag{4.20}$$

kde t odpovídá počtu porovnávaných operátorů, s je počet hodnotitelů a p_{ij} je počet preferencí operátoru i před operátorem j . [11]

Experiment nebyl koncipován tak, aby každý účastník porovnal všechny operátory nad jedním videem. Tudíž bylo, při výpočtu u pro jednotlivá videa, potřeba zjistit hodnotu s takovou, aby odpovídala počtu případů, kdy jeden účastník vyhodnotí všechny možné kombinace nad konkrétním videem. Toho jsem dosáhl vydělením počtu všech porovnání provedených nad konkrétním videem číslem 28, což odpovídá potřebnému počtu porovnání pro dosažení všech kombinací operátorů. Jelikož počty porovnání pro jednotlivé operátory se lehce lišily, není tato hodnota pro jednotlivá videa naprosto přesná.

Pro zjištění významnosti výsledku Kendallova testu jsem využil chí-kvadrát (χ^2). Ten jsem postavil proti nulové hypotéze: „Mezi účastníky experimentu neexistuje shoda.“ Chí-kvadrát odpovídá vzorci [11]:

$$\chi^2 = \frac{t(t-1)(1+u(s-1))}{2}, \quad (4.21)$$

kde t je počet porovnávaných operátorů, s je počet účastníků a u odpovídá výsledku Kendallova testu.[11]

Pro zjištění statistické významnosti se posléze porovnála hodnota $\chi^2 t$ s tabulkou kritických hodnot chí-kvadrátu se stupni volnosti $t(t-1)/2 = 28$. [11][18, tabulka C] Výsledkem pak byla hodnota p vypovídající o potvrzení, či vyvrácení hypotézy.

4.4.4 Realizace

Původním plánem realizace studie bylo provedení experimentu v kontrolovaném prostředí na zkalirovaném monitoru. Výsledky by pak nebyly zkreslené vlastnostmi zařízení, na kterém účastník videa porovnává. Tato možnost však nebyla v době koronavirové krize proveditelná. Rozhodl jsem se tedy pro sdílení experimentu na internetu. Výsledky tedy mohou být zkreslené v závislosti na zobrazovacích zařízeních, které účastníci používali.

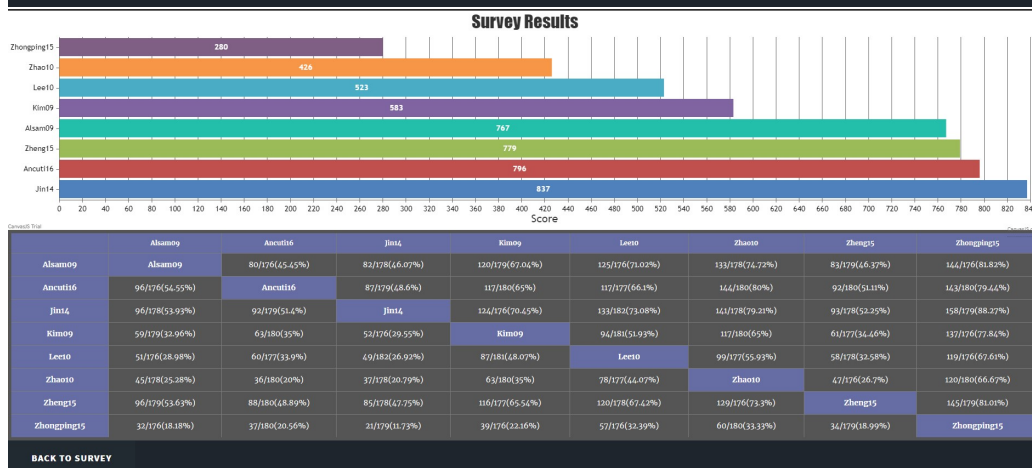
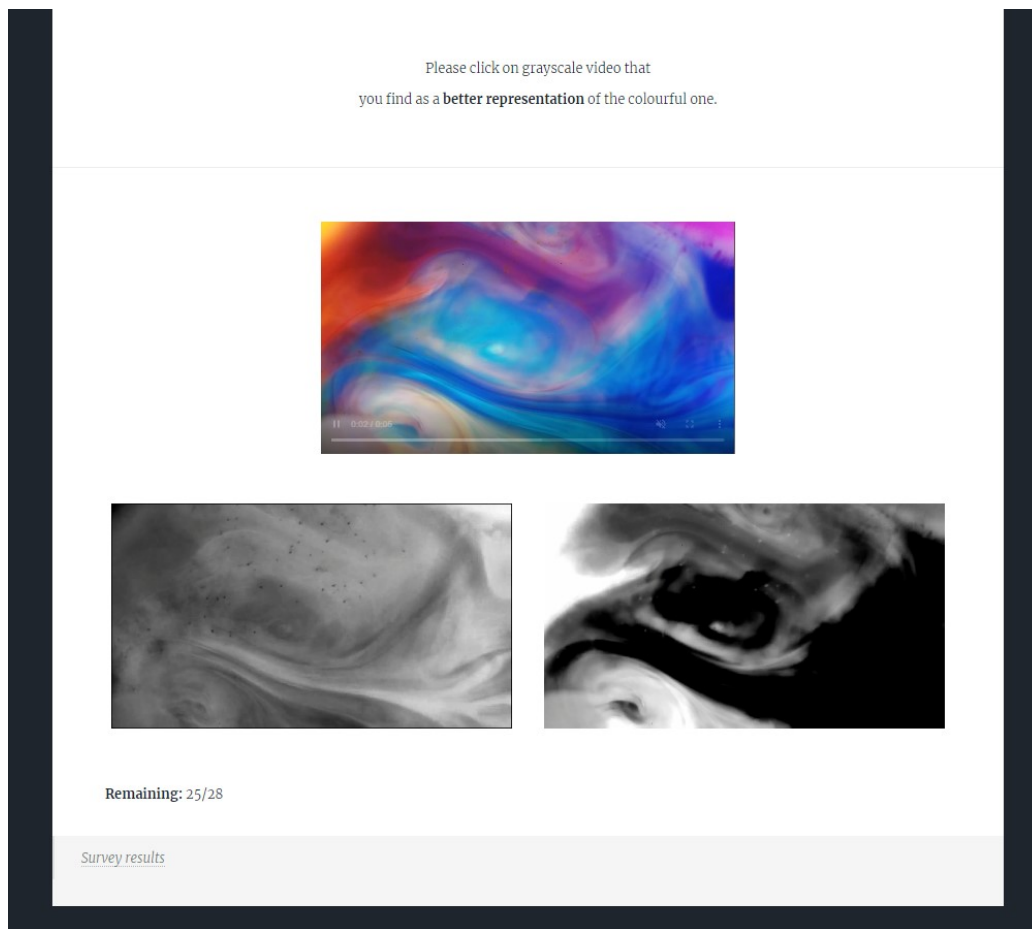
Pro účely studie nebylo možné využít běžných nástrojů pro vytváření a sdílení formulářů, jelikož žádný z nich neposkytoval možnost vložení a synchronizace videa. Bylo tedy potřeba vytvořit vlastní webovou stránku. Experiment se nachází na mé vlastní doméně na adrese <http://janjedlic.cz/survey.php>. Při první návštěvě stránky se zobrazí uvítací strana s popisem experimentu a instrukcemi (viz obrázek 4.17). Stránka je napsána v českém a anglickém jazyce. O tom jaký jazyk bude uživateli zobrazen se rozhoduje na základě nastavení prohlížeče. Český jazyk je přednostně zobrazen v případě, že je obsažen v seznamu uživatelem definovaných preferovaných jazyků použitého prohlížeče. Při absenci tohoto nastavení se automaticky stránka zobrazí v angličtině.

Po potvrzení je účastníkovi odhalena stránka s videi k porovnání. Na stránce se nachází zadání úkolu, referenční barevné video, dvě videa pro porovnání, počítadlo ukazující kolik porovnání z 28 požadovaných ještě uživateli zbývá a odkaz na vizualizaci průběžných výsledků studie. Při najetí myši na šedotónové video se video lehce zvětší, což indikuje možnost výběru. Náhled prostředí na obrázku 4.16.

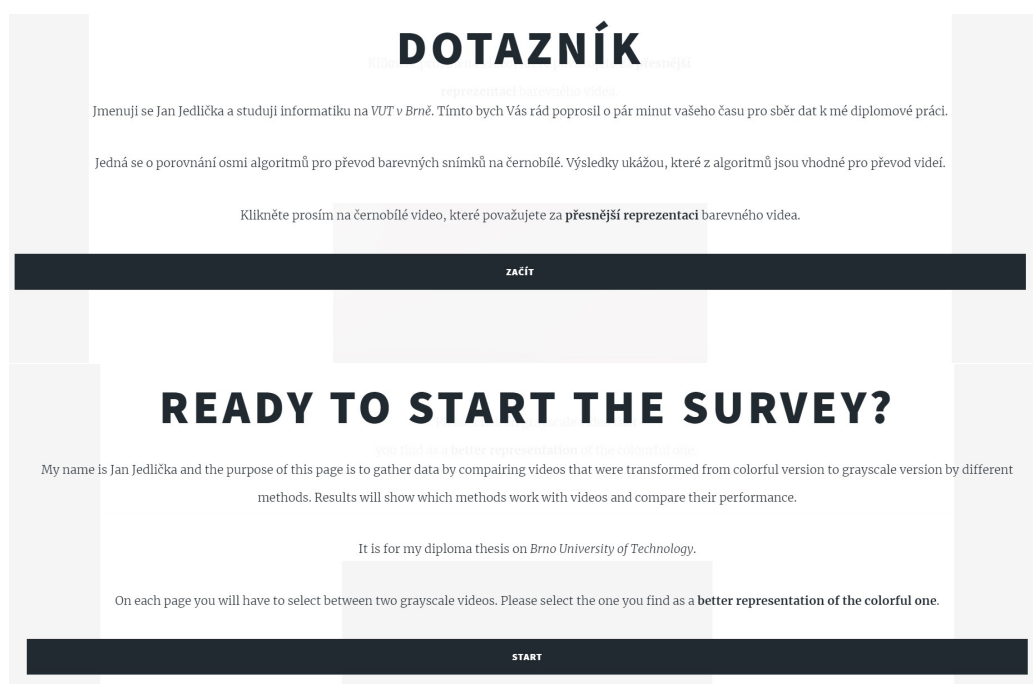
Videa jsou na stránce zobrazeny pomocí *HTML5* elementu `<video>`. Jedním z problémů, které bylo potřeba vyřešit je synchronizace. Ovládání videa (spuštění, pozastavení, přetáčení atp.) je přítomno pouze na referenčním barevném snímku. Dvě šedotónová videa jsou pomocí skriptu synchronizována s barevným referenčním videem a vše se spustí až po načtení všech tří videí. Jako identifikátor uživatele se používá `session id`, které se v průběhu celého experimentu nemění. Jednotlivé výběry preference uživatele se okamžitě ukládají do připojené MySQL databáze.

Po průchodu celým testem se uživateli zobrazí poděkování. Uživatel má možnost zobrazit si průběžné výsledky experimentu, nebo pokračovat dalším průchodem s jinými videi. Této možnosti využilo 35 uživatelů. Graf počtu odpovědí jednotlivých účastníků je k nahlédnutí na obrázku 4.18. Účastníci, kteří ukončili porovnávání dříve, než dosáhli jednoho průchodu, jsem se rozhodl do experimentu nezapočítávat.

Šíření dotazníku jsem nejdříve omezil na úzkou skupinu lidí v mém okolí s žádostí o zpětnou vazbu. Po opravení některých rušivých elementů a zapracování několika estetických připomínek jsem posléze sdílel dotazník veřejnosti. Pro sdílení jsem využil zejména



Obrázek 4.16: Nahoře: Hlavní stránka dotazníku s porovnáváním. Dole: Stránka s prezentací průběžných výsledků.



Obrázek 4.17: Ukázka uvítací stránky dotazníku v českém a anglickém jazyce.



Obrázek 4.18: Graf počtu provedených porovnání jednotlivých účastníků seřazených od nejvyššího. (Oranžové čáry reprezentují jeden průchod testem)

facebookových skupin včetně VUT FIT MIT 2017–2019¹, VUT FIT MIT 2018–2020², VUT FIT MIT 2019–2021³ a oslovil jsem také své kolegy v práci a přátele s prosbou o sdílení. Pokaždé bylo u odkazu na dotazník napsáno, aby každý kdo má nějakou vadu zraku svou účast zvážil a pokud má prostředky jak tuto vadu opravit (např. dioptrické brýle), aby je využil.

4.5 Výsledky uživatelské studie

Do experimentu se celkem zapojilo 156 účastníků, kteří dohromady provedli 4988 jednotlivých porovnání. Do těchto počtů nejsou započítáni účastníci, kteří nedokončili alespoň jeden průchod experimentem (provedli méně než 28 porovnání). Jednalo se o 40 účastníků, z nichž většina provedla méně než 10 porovnání (viz obrázek 4.18).

V této části se zaměřím na zhodnocení výsledků uživatelské studie. Analyzuji celkové výsledky napříč celým datasetem, poté se zaměřím na jednotlivá videa a vyzdvihnu některá konkrétní data, které stojí za pozornost.

4.5.1 Výsledky pro celý dataset

| - | Als | Anc | Jin | Kim | Lee | Zha | Zhe | Zho | Pref. | z-score |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|--------|---------|
| Alsam09 | 0 | 80 | 82 | 120 | 125 | 133 | 83 | 144 | 61,80% | 0,726 |
| Ancuti16 | 96 | 0 | 87 | 117 | 117 | 144 | 92 | 143 | 63,58% | 0,835 |
| Jin14 | 96 | 92 | 0 | 124 | 133 | 141 | 93 | 157 | 66,93% | 1,041 |
| Kim09 | 59 | 63 | 52 | 0 | 94 | 117 | 61 | 137 | 46,71% | -0,201 |
| Lee10 | 50 | 60 | 49 | 87 | 0 | 99 | 58 | 119 | 41,89% | -0,498 |
| Zhao10 | 45 | 36 | 37 | 62 | 78 | 0 | 47 | 120 | 34,05% | -0,979 |
| Zheng15 | 96 | 88 | 85 | 116 | 120 | 129 | 0 | 145 | 62,47% | 0,767 |
| Zhongping15 | 32 | 37 | 21 | 39 | 57 | 60 | 34 | 0 | 22,49% | -1,690 |

Tabulka 4.4: Preferenční matice pro celý dataset spolu s výsledky. Pref. = procenta porovnání, kdy byl operátor preferován.

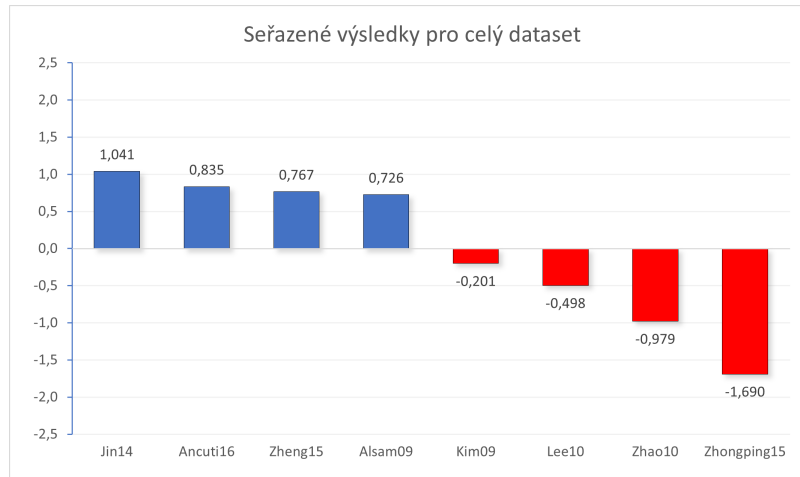
Pro celý dataset byla vytvořena preferenční matice (viz tabulka 4.4), ze které se vypočítalo z-score. Z grafu na obrázku 4.19 je jasně patrné že zkoumané operátory můžeme rozdělit na dvě poloviny. Výsledky operátorů Jin14, Ancuti16, Zheng15 a Alsam09 se v tomto pořadí od nejlepšího dostaly celkem vysoko nad průměr. Pod průměrem se postupně nacházejí operátory Kim09, Lee10, Zhao10 a na posledním místě operátor Zhongping15. Celkově se dá pozorovat, že operátory pracující lokálně s jednotlivými pixely snímku se umístily v horní polovině žebříčku, zatímco globální operátory vykazují spíše horší výsledky.

Výsledky horní poloviny žebříčku byly velmi těsné. Kromě operátoru Jin14, který byl nadprůměrný i v rámci výsledků všech jednotlivých videí datasetu, se operátory Ancuti16, Zheng15 a Alsam09 dají považovat z celkového hlediska za rovnocenné. Vzhledem rozdílu mezi výsledky posledních čtyř operátorů nemůžeme říct, že by nějaké patřily na stejnou úroveň.

¹<https://www.facebook.com/groups/vutfitmit2017/>

²<https://www.facebook.com/groups/2002665913382033/>

³<https://www.facebook.com/groups/fitmit2019/>



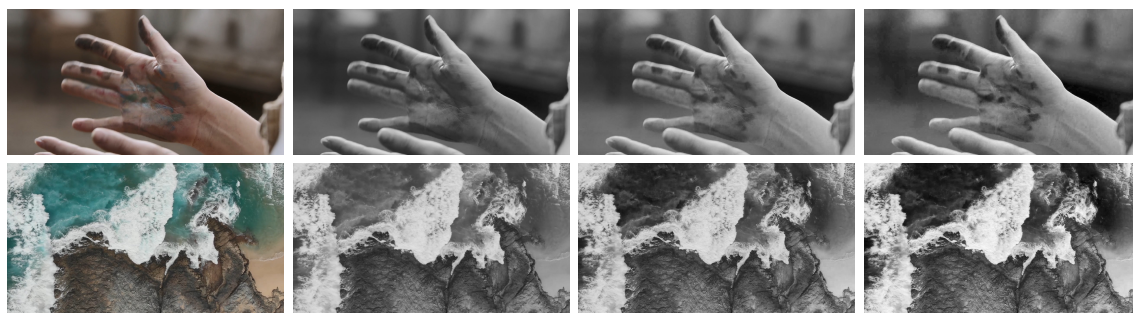
Obrázek 4.19: Graf celkových výsledků studie pro celý dataset. Operátory seřazené podle pořadí.

Mezi porovnávanými operátory byly i tři operátory (Kim09, Lee10 a Zheng15), které již byly zkoumány ve studii Vladimíra Vlkoviče (viz 4.2.1). V jeho studii se tyto tři operátory umístily ve stejném pořadí jako nyní, přičemž operátor Lee10 byl výrazně nejhorším (viz obrázek 4.3). Zde se ještě za ním umístily operátory Zhao10 a Zhongping15. Jako důvod se nabízí skutečnost, že u většiny videí se po aplikování těchto operátorů vyskytovaly blikající artefakty. Časově závislá metoda Kim09 se v obou studiích umístila nejbližší průměru a několik časově nezávislých metod se dostalo před ní. Potvrzuje se tedy Vlkovičův závěr, že časově nezávislé metody mohou při aplikaci na video konkurovat časově závislým metodám, pokud nevytváří artefakty a dodržují temporální koherenci[19].

4.5.2 Výsledky pro jednotlivá videa

Souhrn výsledků pro jednotlivá videa je uveden v tabulce 4.5. Jednotlivé grafy s výsledky jsou pak k nahlédnutí v příloze D spolu s porovnáním všech operátorů s celkovým výsledkem studie. První zajímavostí je video *Shore*, kde operátor Alsam09 naprosto selhal. Kromě videa *ColorHand* byl totiž u všech dalších videí nadprůměrný. Stejná dvojice videí naopak výjimečně uspěla u operátorů Lee10 a Zhao10. Při pohledu na obrázek 4.20 je patrné, že operátor Alsam09 zvládl zachovat méně kontrastu, ale hlavním důvodem bylo nejspíš jemné blikání. Obě videa spojuje rychlejší pohyb. Další zajímavostí je video *Horizon*, kde se u operátoru Kim09 objevily blikající artefakty, což zřejmě způsobilo jeho nízké skóre. Naopak si tento operátor vedl dobře u videa *Bear*.

U videí *Corals* a *Turtle* můžeme v tabulce 4.5 vidět celkem vysoký koeficient shody. Obě videa jsou natočena pod vodou a obsahují jak ostré detaily, tak rozmazané pozadí. Některé operátory vykreslily toto pozadí téměř černé, jiné skoro bílé. Lze vyzorovat, že oba případy vedly ke snížení skóre daného operátoru.



Obrázek 4.20: Porovnání operátorů u videí *ColorHand* a *Shore*. Zleva:Originál, Alsam09, Lee10, Zhao10

| Video | u | χ^2 | $p(DF=28)$ | Pořadí podle preference |
|-------------|-------|----------|------------|----------------------------|
| ColorHand | 0,504 | 176,649 | <0.001 | Ji An Le Zhe Zha Ki Al Zho |
| ColorsWater | 0,429 | 148,429 | <0.001 | Zhe Al Ki Ji An Zha Zho Le |
| Drone | 0,496 | 171,870 | <0.001 | An Ji Al Le Ki Zhe Zha Zho |
| Horizon | 0,751 | 235,970 | <0.001 | An Al Le Ji Zhe Zha Zho Ki |
| Turtle | 0,734 | 233,455 | <0.001 | An Ji Al Le Zhe Zha Ki Zho |
| Bear | 0,448 | 161,519 | <0.001 | Zhe Al Ki Ji An Zha Le Zho |
| Silueta | 0,317 | 117,995 | <0.001 | Ji Zhe Al Ki An Le Zho Zha |
| Shore | 0,317 | 113,397 | <0.001 | Le An Ji Zha Zhe Zho Ki Al |
| Crossroad | 0,345 | 126,455 | <0.001 | Ji Al Le Ki An Zhe Zha Zho |
| ManNature | 0,678 | 230,021 | <0.001 | Ji Zhe Ki Le Al An Zha Zho |
| Surf | 0,278 | 106,374 | <0.001 | An Ji Al Zhe Zha Zho Le Ki |
| Corals | 0,860 | 270,447 | <0.001 | An Al Zhe Ji Ki Zha Zho Le |
| Boy | 0,517 | 174,795 | <0.001 | Al Zhe An Ji Ki Le Zho Zha |
| Bridge | 0,504 | 169,091 | <0.001 | Ji Zhe An Al Le Ki Zha Zho |
| PinkSmoke | 0,448 | 151,095 | <0.001 | An Ji Zhe Ki Al Zho Le Zha |
| Colours | 0,248 | 97,455 | <0.001 | Zhe Al Ji Ki An Zho Zha Le |

Tabulka 4.5: Výsledky pro jednotlivá videa s metodami seřazenými podle preference zleva doprava. Al=Alsam09, An=Ancuti16, Ji=Jin14, Ki=Kim09, Le=Lee10, Zha=Zhao10, Zhe=Zheng15, Zho=Zhongping15

Kapitola 5

Závěr

Cílem této práce bylo zprovoznění, modernizace a vylepšení grafického uživatelského rozhraní programu Tone Mapping Studio, který je navržen pro práci s HDR snímky a videem. Z analýzy grafických uživatelských rozhraní podobných programů jsem navrhl možné změny a vylepšení. Provedl jsem několik iterací uživatelského testování. Výstupy z testování jsem zhodnotil a rozhraní ještě rozšířil o chybějící funkcionalitu. Rozhraní nyní funguje na verzi knihovny *Qt5*, která je kompatibilní s posledními verzemi knihoven, které program TMS využívá. Je k dispozici živý náhled výsledku aplikace operátoru a náhledy výstupů jednotlivých operátorů pro právě zvolený snímek. Rozhraní je přehlednější a poskytuje potřebné nástroje pro práci s nejen HDR obrazy. Provedl jsem studii, ve které jsem porovnal 8 metod převodu barevných snímků na šedotónové. Studie se zajímala o jejich aplikaci na video sekvence. Výsledky studie jsem porovnal s podobnou studií a rozšířil její výsledky o nové poznatky. Ve studii se jako nejuniverzálnější operátor ukázal *Jin14*, který byl konstantně nad průměrem bez ohledu na použité video.

V práci lze jistě pokračovat například provedením komplexnější studie v kontrolovaném prostředí. Implementací nových operátorů pro *TMS* a opravou definic a výstupů již existujících. Uživatelské prostředí programu lze jistě dále vylepšovat a přidávat nové funkce, které pomohou uživatelům při práci s HDR technologií.

Literatura

- [1] Alsam, A.: Contrast enhancing colour to grey. In *Scandinavian Conference on Image Analysis*, Springer, 2009, s. 588–596.
- [2] Ancuti, C.; Ancuti, C. O.: Laplacian-guided image decolorization. In *2016 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2016, s. 4107–4111.
- [3] Boitard, R.; Cozot, R.; Thoreau, D.; aj.: Survey of temporal brightness artifacts in video tone mapping. In *HDRi2014-Second international conference and SME workshop on HDR imaging*, ročník 9, 2014.
- [4] Čadík, M.: Perceptual Evaluation of Color-to-Grayscale Image Conversions. *Comput. Graph. Forum*, ročník 27, č. 7, 2008: s. 1745–1754.
- [5] David, H. A.: *The method of paired comparisons*, ročník 12. London, 1963.
URL <https://apps.dtic.mil/sti/pdfs/ADA417190.pdf#page=15>
- [6] Ibraheem, N. A.; Hasan, M. M.; Khan, R. Z.; aj.: Understanding color models: a review. *ARPN Journal of science and technology*, ročník 2, č. 3, 2012: s. 265–275.
- [7] Ji, Z.; Fang, M.-e.; Wang, Y.; aj.: Efficient decolorization preserving dominant distinctions. *The Visual Computer*, ročník 32, č. 12, 2016: s. 1621–1631.
- [8] Jin, Z.; Li, F.; Ng, M. K.: A variational approach for image decolorization by variance maximization. *SIAM Journal on Imaging Sciences*, ročník 7, č. 2, 2014: s. 944–968.
- [9] Kim, Y.; Jang, C.; Demouth, J.; aj.: Robust color-to-gray via nonlinear global mapping. In *ACM SIGGRAPH Asia 2009 papers*, ACM, 2009, s. 1–4.
- [10] Krug, S.: *Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability Problems*. New Riders, dec 2009, ISBN 0321657292.
- [11] Ledda, P.; Chalmers, A.; Troscianko, T.; aj.: Evaluation of tone mapping operators using a high dynamic range display. *ACM Transactions on Graphics (TOG)*, ročník 24, č. 3, 2005: s. 640–648.
- [12] Lee, T.-H.; Kim, B.-K.; Song, W.-J.: Converting color images to grayscale images by reducing dimensions. *Optical Engineering*, ročník 49, č. 5, 2010: str. 057006.
- [13] Mantiuk, R.; Daly, S.; Kerofsky, L.: Display adaptive tone mapping. In *ACM Transactions on Graphics (TOG)*, ročník 27, ACM, 2008, str. 68.
- [14] Mantiuk, R.; Myszkowski, K.; Seidel, H.-P.: A perceptual framework for contrast processing of high dynamic range images. *ACM Transactions on Applied Perception (TAP)*, ročník 3, č. 3, 2006: s. 286–308.

- [15] Petit, J.; Mantiuk, R. K.: Assessment of video tone-mapping: Are cameras' S-shaped tone-curves good enough? *Journal of Visual Communication and Image Representation*, ročník 24, č. 7, 2013: s. 1020–1030.
- [16] Plataniotis, K. N.; Venetsanopoulos, A. N.: *Color image processing and applications*. Springer Science & Business Media, 2013.
- [17] Reinhard, E.; Stark, M.; Shirley, P.; aj.: Photographic tone reproduction for digital images. In *ACM transactions on graphics (TOG)*, ročník 21, ACM, 2002, s. 267–276.
- [18] Siegel, S.: *Nonparametric statistics for the behavioral sciences*. 1956.
- [19] Vlkovič, V.: *Porovnání časově závislých metod pro převod barevného obrazu na šedotónový*. Diplomová práce, Vysoké Učení Technické v Brně, Fakulta informačních technologií, Brno, 2018, vedoucí práce Doc. Ing. Martin Čadík, Ph.D.
- [20] Zhao, Y.; Tamimi, Z.: Spectral image decolorization. In *International Symposium on Visual Computing*, Springer, 2010, s. 747–756.
- [21] Zheng, X.; Feng, J.; Zhou, B.: Efficient color-to-gray conversion for digital images in gradient domain. In *Proceedings of the 14th ACM SIGGRAPH International Conference on Virtual Reality Continuum and Its Applications in Industry*, 2015, s. 85–88.

Příloha A

Obsah přiloženého paměťového média

Obsah přiloženého paměťového média je následovný:

- **DP** - Diplomová práce
 - **DP-Latex** - Zdrojové soubory k diplomové práci pro \LaTeX
- **TMS** - Tone Mapping Studio
 - **TMOgui** - grafické uživatelské rozhraní
 - * - ostatní adresáře neobsahují mnou psaný kód
- **Data** - data použítá v uživatelské studii
 - **Video** - Dataset a operátory převedená videa
 - data.csv - Tabulka s výsledky experimentu

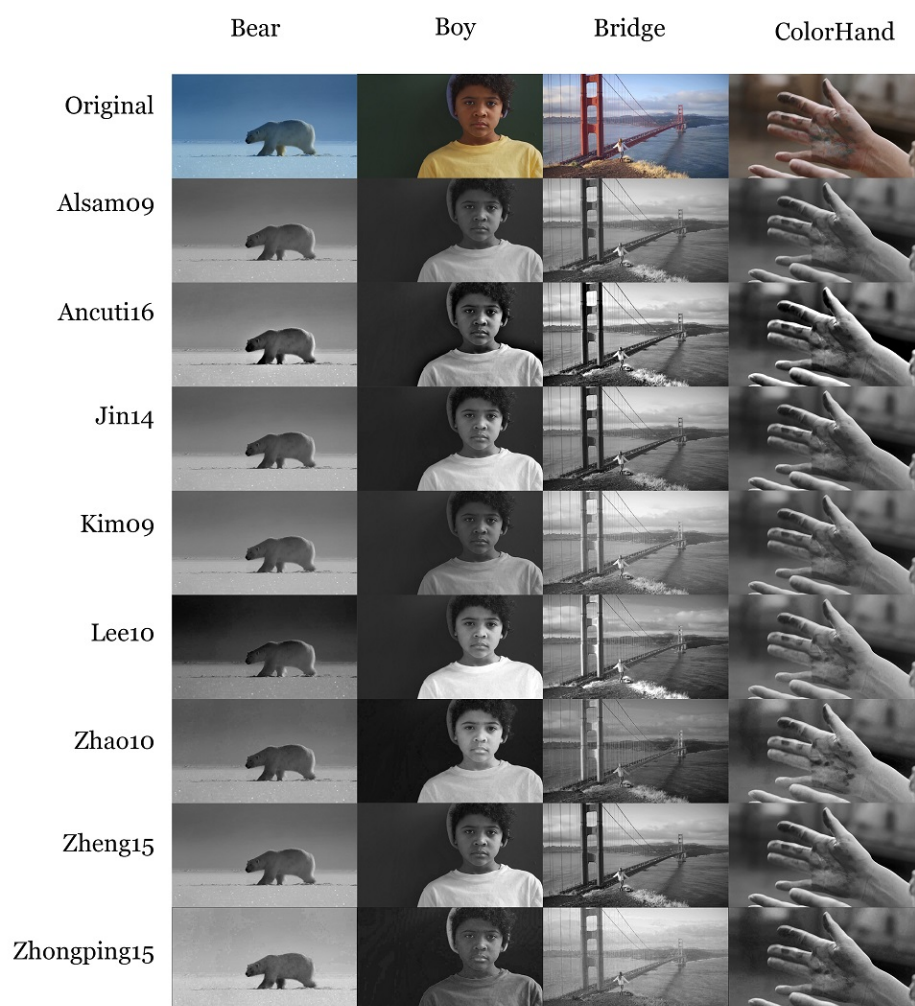
Příloha B

Příklad zadání uživatelského testování

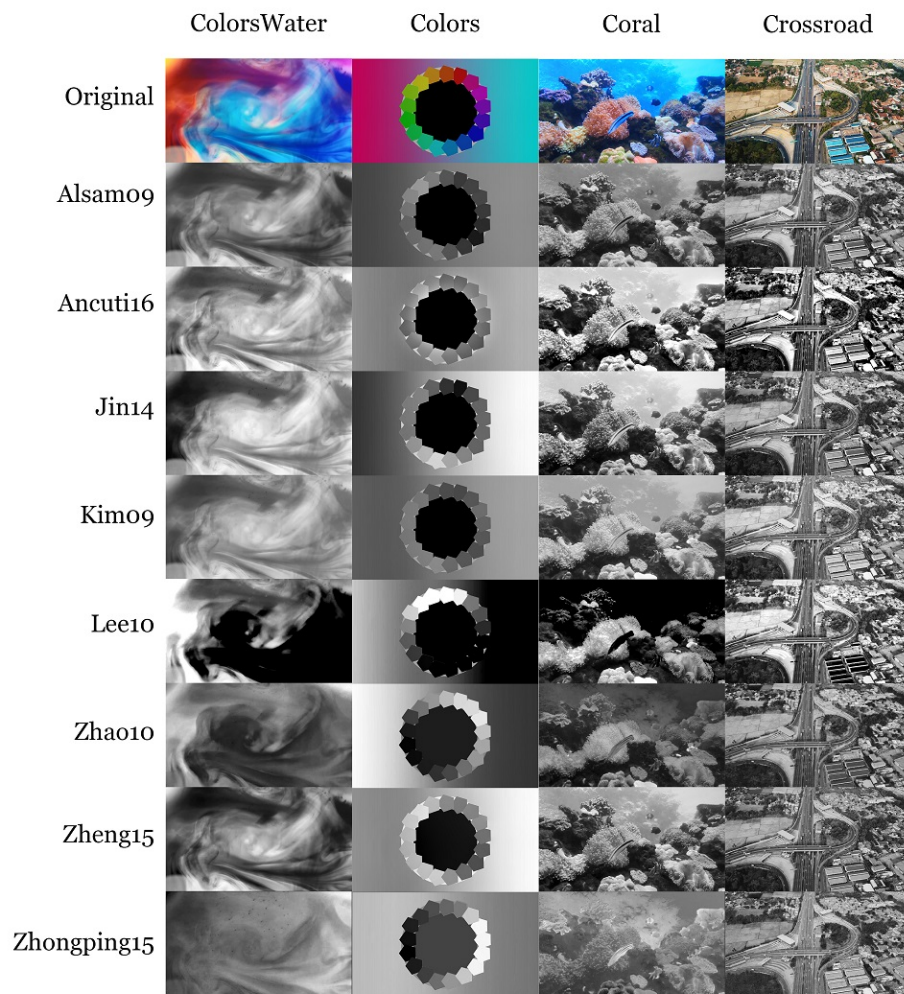
1. Bez interakce s programem popište jednotlivé prvky rozhraní a snažte se vysvětlit jejich funkci.
2. Otevřete soubor test.hdr nacházející se v adresáři `Pictures`.
3. Upravte hodnoty histogramu tak, aby snímek vypadal přirozeně.
4. Přečtěte nápis na propisce, která na snímku leží na knize.
5. Zjistěte průměrné hodnoty jasu modré barvy na levé stránce knihy.
6. Pomocí filtrů zvýrazněte zelenou barvou oblasti snímku, kde je hodnota jasu nižší než 10.2 a vyšší než 3.5.
7. Převedte snímek do černobílé tak, aby se výsledek co nejvíce podobal originálu.
8. Černobílý snímek uložte do adresáře `Pictures/BW` ve formátu TIF.
9. Změňte velikost snímku tak, aby se zachoval poměr stran a jeho výška byla 200 pixelů.
10. Upravte uživatelské rozhraní tak, aby byl snímek téměř přes celou obrazovku.

Příloha C

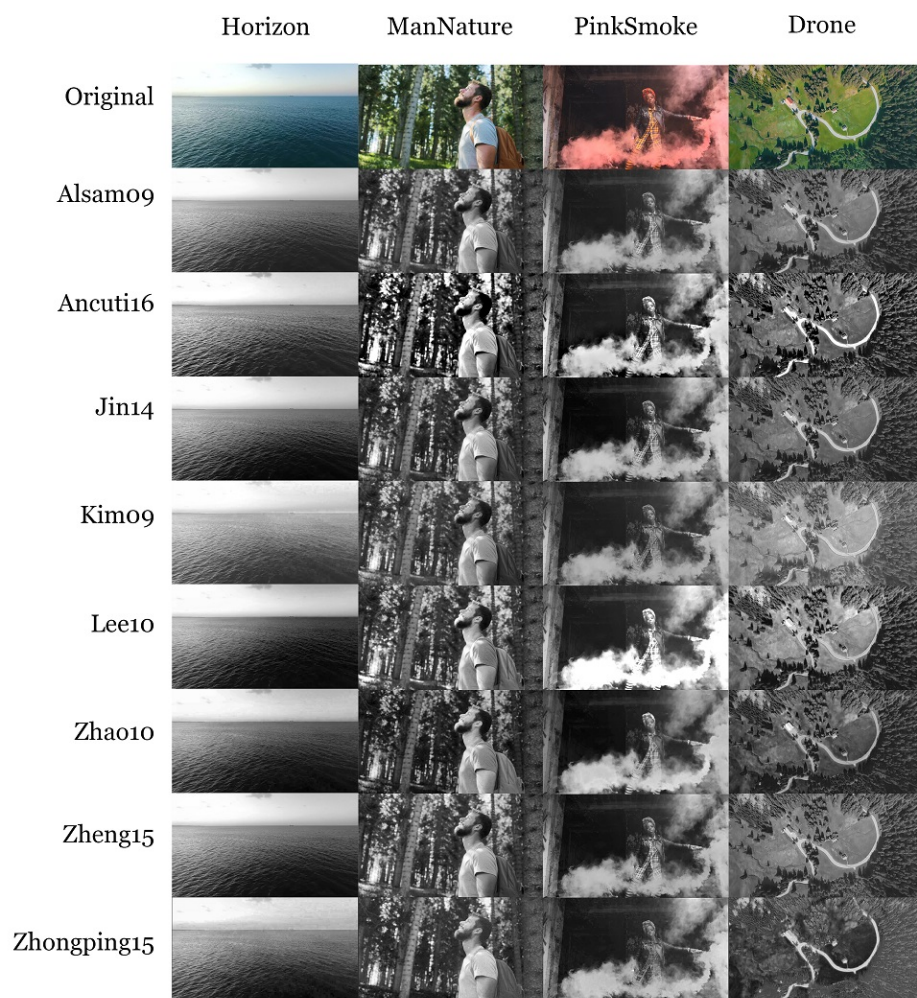
Dataset



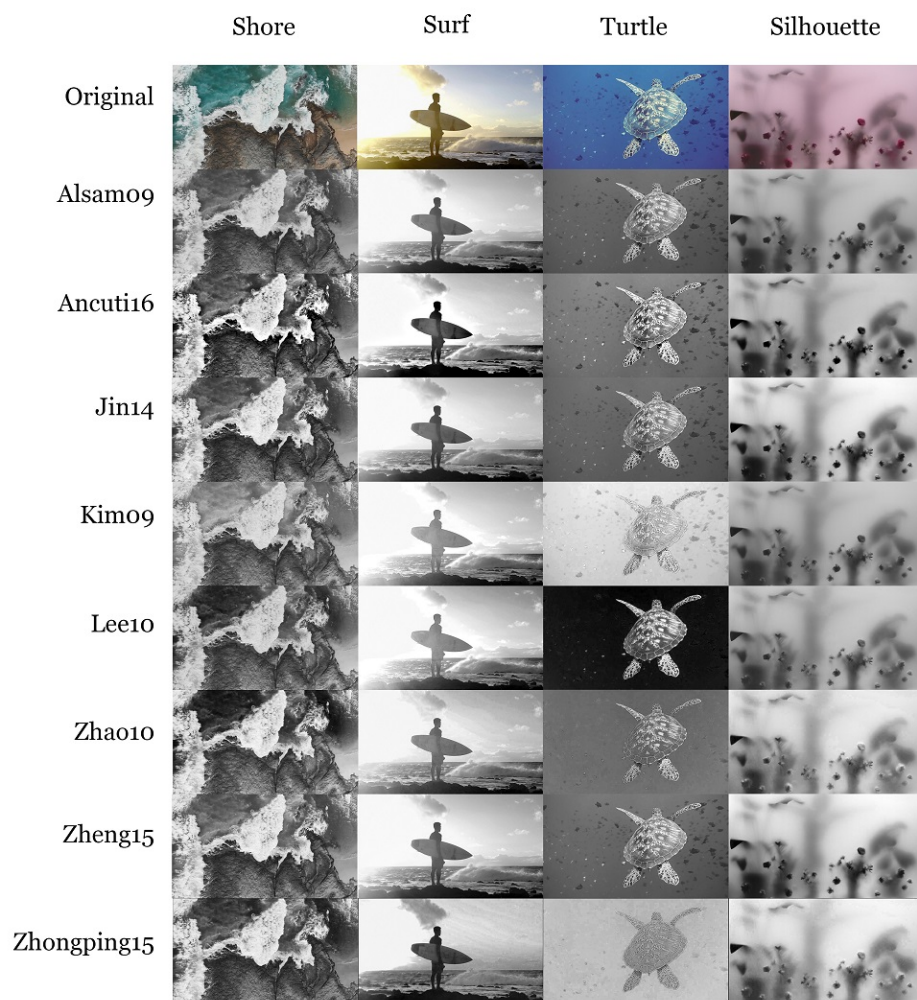
Obrázek C.1: Náhled videí z datasetu a jejich převedených šedotónových variant.



Obrázek C.2: Náhled videí z datasetu a jejich převedených šedotónových variant.



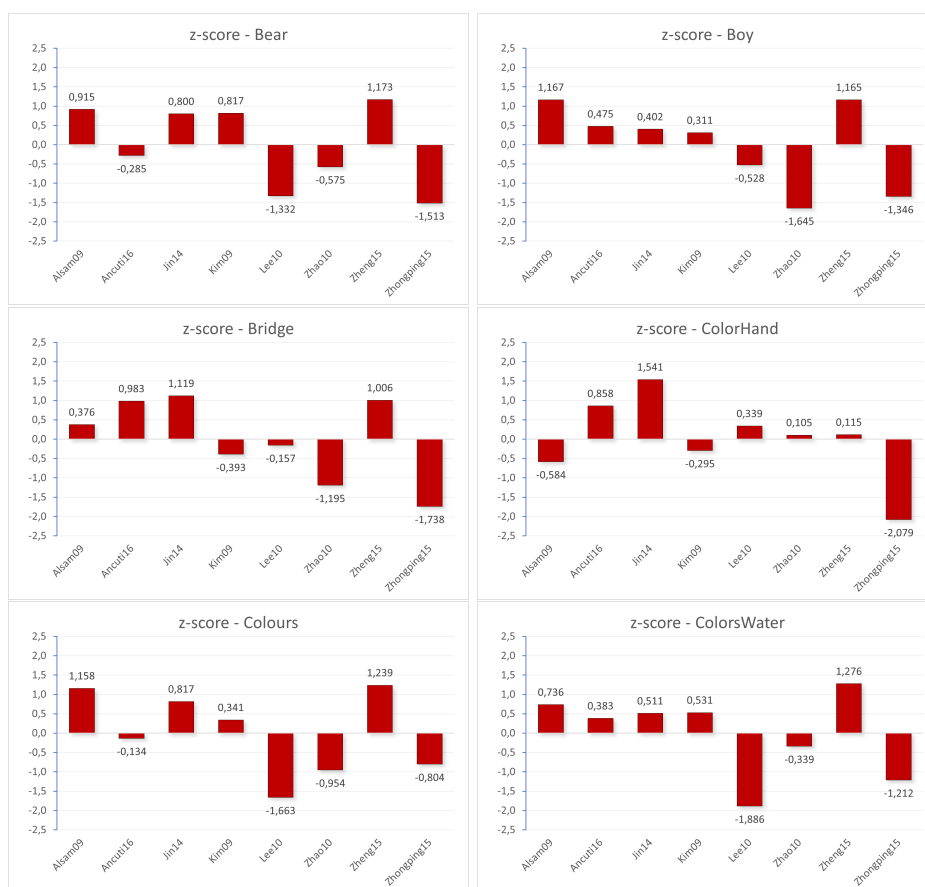
Obrázek C.3: Náhled videí z datasetu a jejich převedených šedotónových variant.



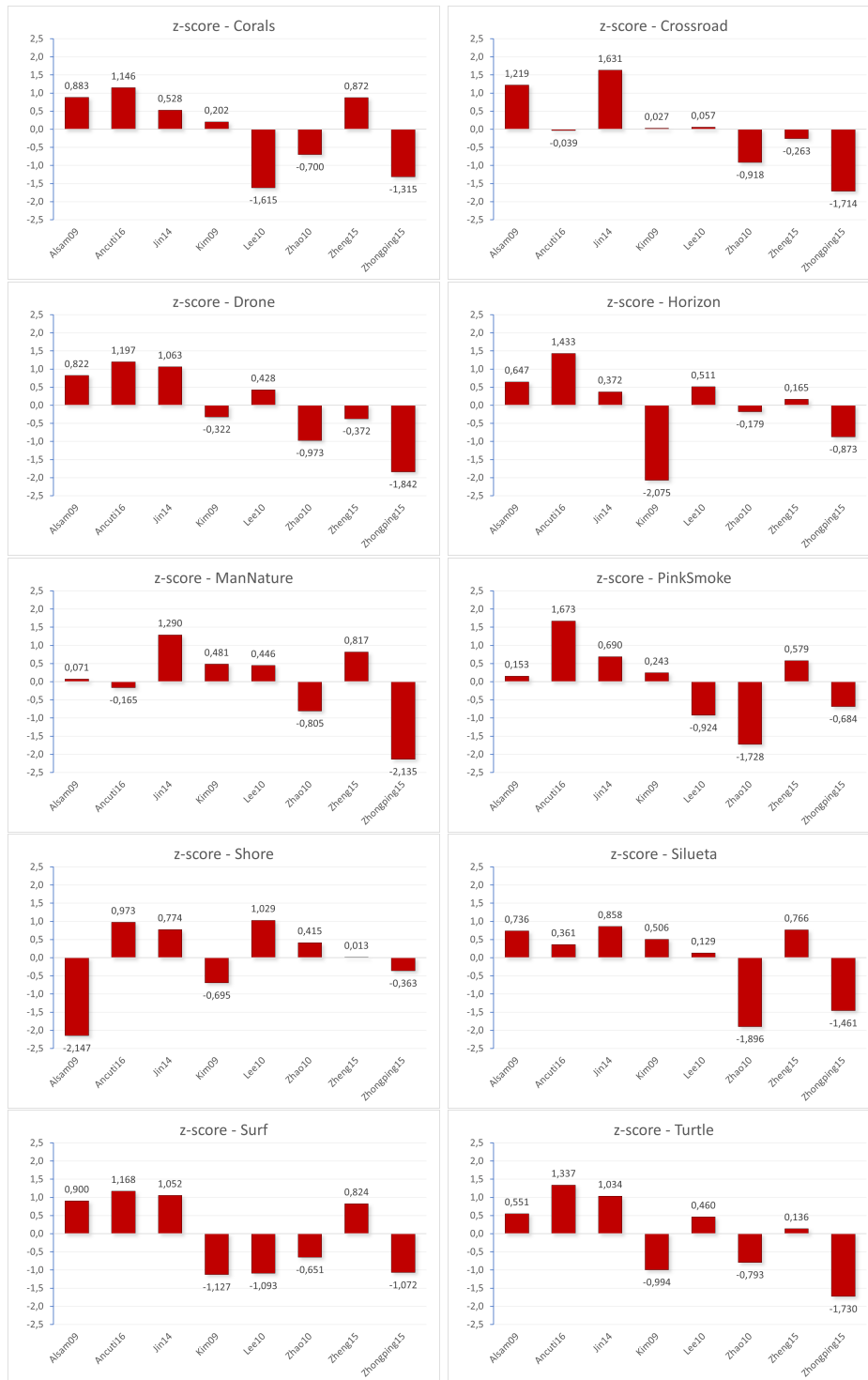
Obrázek C.4: Náhled videí z datasetu a jejich převedených šedotónových variant.

Příloha D

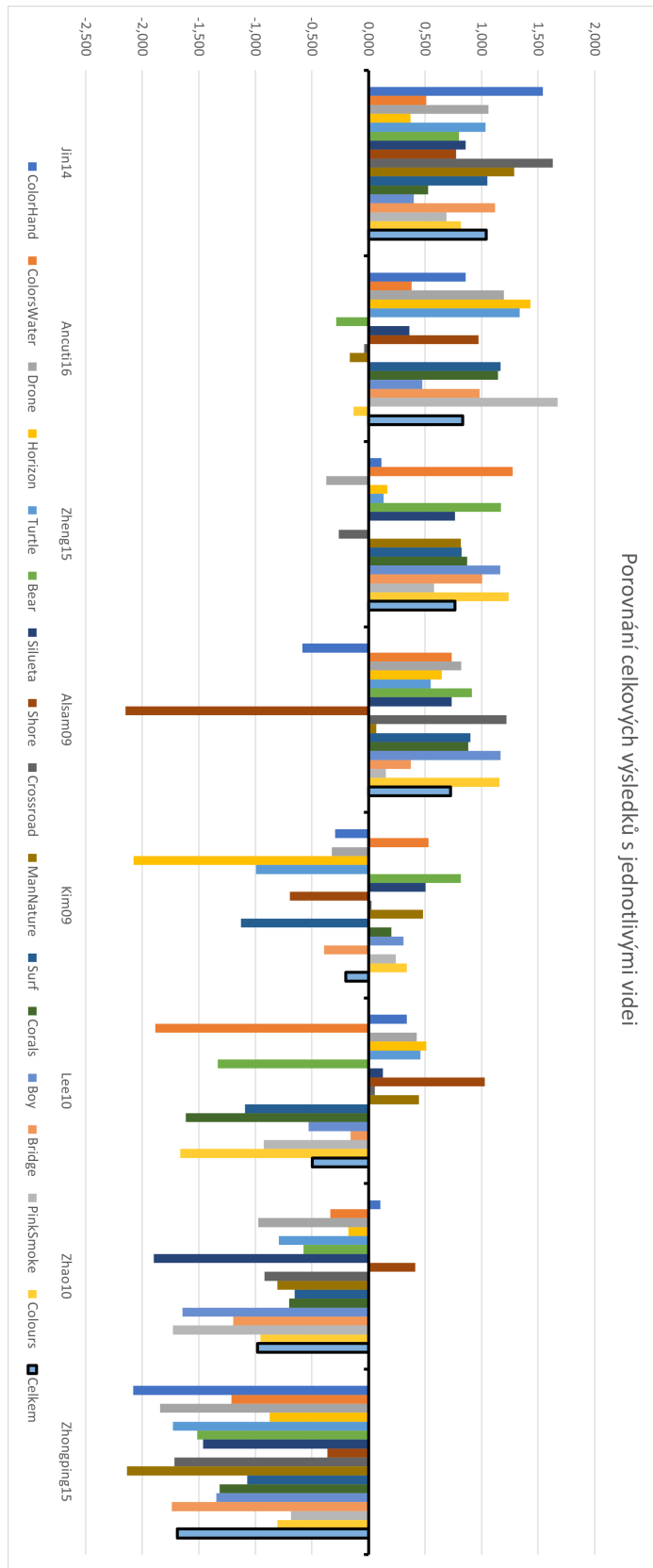
Výsledky pro jednotlivá videa



Obrázek D.1: Výsledky pro jednotlivá videa.



Obrázek D.2: Výsledky pro jednotlivá videa.



Obrázek D.3: Porovnání celkových výsledků operátorů s výsledky u jednotlivých videí.

Příloha E

Odkazy na videa

- Bear.mp4: <https://www.pexels.com/video/footage-of-polar-bear-4517474/>
- Boy.mp4: <https://www.pexels.com/video/video-of-a-boy-looking-at-camera-7653682/>
- Bridge.mp4: <https://www.pexels.com/video/woman-enjoying-the-great-view-in-slow-motion-3775287/>
- ColorHand.mp4: <https://www.pexels.com/video/hands-full-of-paint-3796261/>
- ColorsWater.mp4: <https://www.pexels.com/video/splatter-of-mixing-color-liquids-4379024/>
- Corals.mp4: <https://www.pexels.com/video/marine-life-of-fishes-and-corals-underwater-3765078/>
- Crossroad.mp4: <https://www.pexels.com/video/aerial-view-of-a-freeway-1472012/>
- Drone.mp4: <https://www.pexels.com/video/aerial-shot-of-land-856477/>
- Horizon.mp4: <https://www.pexels.com/video/view-of-the-horizon-1570919/>
- ManNature.mp4: <https://www.pexels.com/video/man-looking-at-the-wonder-of-nature-3135808/>
- PinkSmoke.mp4: <https://www.pexels.com/video/a-woman-dispensing-a-smoke-screen-inside-an-abandon-building-3044455/>
- Shore.mp4: <https://www.pexels.com/video/waves-rushing-and-splashing-to-the-shore-1409899/>
- Silueta.mp4: <https://www.pexels.com/video/silhouette-of-a-dancer-6541889/>
- Surf.mp4: <https://www.pexels.com/video/man-holding-a-surfboard-1093664/>
- Turtle.mp4: <https://www.pexels.com/video/turtle-underwater-1456996/>