



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

NEREALISTICKÉ ZOBRAZOVÁNÍ

NON-REALISTIC RENDERING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ZUZANA SKALNÍKOVÁ

VEDOUCÍ PRÁCE

SUPERVISOR

Prof. Dr. Ing. PAVEL ZEMČÍK

BRNO 2016

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2015/2016

Zadání bakalářské práce

Řešitel: **Skalníková Zuzana**

Obor: Informační technologie

Téma: **Nerealistické zobrazování**
Non-Realistic Rendering

Kategorie: Počítačová grafika

Pokyny:

1. Prostudujte literaturu na téma "nerealistické zobrazování" - tedy zobrazování typu "cartoon", simulace uměleckých technik (malířských), případně navrhněte vlastní postupy.
2. Vyberte metodu zobrazování a vyhodnoťte její vlastnosti z pohledu způsobu zobrazování i z pohledu náročnosti implementace.
3. Navrhněte způsob implementace vybrané metody a diskutujte jeho vlastnosti, výhody i nevýhody.
4. Implementujte vybranou metodu a demonstруйте její vlastnosti na vhodném příkladě.
5. Diskutujte dosažené výsledky a možnosti pokračování práce.

Literatura:

- Dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3 zadání

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: **Zemčík Pavel, prof. Dr. Ing., UPGM FIT VUT**

Datum zadání: 1. listopadu 2015

Datum odevzdání: 18. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
602 00 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Tato bakalářská práce se zaměřuje na popis nerealistického zobrazování. V práci jsou popsány nejznámější techniky, kterými lze dodat obrazu umělecký dojem. Rovněž jsou popsány elementární algoritmy, které jsou potřebné pro tyto transformace. Součástí práce je i implementace aplikace, která by některou popsanou uměleckou techniku využila.

Abstract

This bachelor thesis is focused on describing non-realistic rendering. The thesis describes well-known techniques, which are used to add artistic feel to the display data. One part of the work describes elementary algorithms, which are required for these transformations. Part of the thesis is also implementation of application, which would use some of the described artistic technique.

Klíčová slova

počítačová grafika, Nerealistické zobrazování, zpracování videa, Qt, komiksový efekt, OpenCV

Keywords

computer graphics, Non-realistic rendering, video processing, Qt, comics effect, OpenCV

Citace

SKALNÍKOVÁ, Zuzana. *NEREALISTICKÉ ZOBRAZOVÁNÍ*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Zemčík Pavel.

NEREALISTICKÉ ZOBRAZOVÁNÍ

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením profesora Pavla Zemčíka. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

.....

Zuzana Skalníková

17. května 2016

Poděkování

Chtěla bych poděkovat svému vedoucímu bakalářské práce prof. Dr. Ing. Pavlu Zemčíkovi za odborné vedení, za pomoc a rady při zpracování této práce.

© Zuzana Skalníková, 2016.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	2
2	Realistická a nerealistická tvorba	3
2.1	Realistické zobrazování	3
2.2	Nerealistické zobrazování	4
2.3	Umělecké techniky nerealistické tvorby	4
2.4	Existující práce s nerealistickým zobrazováním	7
3	Zpracování videa a obrazových dat	9
3.1	Barevné modely a uspořádání barev	10
3.2	Konvoluce u úprav videa	12
3.3	Rozostření	13
3.4	Redukce barev	15
3.5	Detekce hran	17
4	Realizační část práce	19
4.1	Návrh řešení a motivace	19
4.2	Knihovna QT	20
4.3	Knihovna OpenCV	21
4.4	Zpracování vstupního videa	21
4.5	Zpracování jednotlivých snímků videa	22
4.6	Grafické úpravy jednotlivých snímků	22
4.7	Překrytí snímků	23
4.8	Výsledná aplikace	24
4.9	Testování	27
4.10	Optimalizace	28
5	Závěr	30
6	Literatura	31
A	Obsah CD	33

Kapitola 1

Úvod

Charakteristickým jevem pro dnešní dobu je zachytávat všechno, co nás něčím nadchne do obrazových dat, jako jsou fotografie nebo videa. Dalším takovým jevem je snaha o zobrazení představ a myšlenek do zobrazitelné podoby. Toto zobrazení se pak vytváří většinou elektronicky. Počítačová grafika se věnuje právě všem způsobům zobrazování předmětů, jevů nebo čehokoliv co zobrazit lze.

Samotné grafické úpravy obrazu a stylizace do jiných forem se stávají v dnešní populárním trendem. Různé techniky určené pro dosažení různých efektů v obrazu se dostávají jak do počítačové grafiky, tak i do videa. U grafiky vytvářené pomocí počítače byla vždy snaha o co nejpřesnější zachycení reálného světa. Nerealistické zobrazování se naopak často snaží zavést techniky používané v umění.

Označení těchto technik jako ne-fotorealistické a nerealistické zobrazování je v současné době předmětem mnoha dohadů mezi lidmi zajímajícími se o tuto tematiku. Například proto, že ti co nahlíží na grafiku z čistě profesionálního až matematického hlediska, vnímají obraz reality jinak než umělci. Dalším neshodou byla úvaha, že není příliš vhodné něco popisovat tím, čím ta věc není. Tedy nazývat tyto grafické techniky jako nerealistické zobrazování je podobné jako použít termín nematematické vzorce.

Práce s grafikou a úpravou videí má v programování velkou výhodu. Programátor při jakýkoliv změně použitého algoritmu na úpravu obrazových dat bezprostředně vidí výsledek svoji práce. To bylo i velkou motivací pro mě vybrat si práci z grafické oblasti. Motivací k tvorbě této práce je taky momentální stoupající trend zobrazování nerealistického obrazu oproti fotorealistickému přístupu.

Cílem této práce je tedy čtenáře seznámit s některými technikami, které se v současné době používají pro nerealistické zobrazování a některou z těchto technik demonstrovat aplikací a výslednou implementaci popsat. V druhé kapitole se tato práce zaměřuje na popis realistické a nerealistické tvorby. Jsou popsány i umělecké techniky charakteristické pro nerealistickou tvorbu a už existující práce s touto tvorbou. Třetí kapitola vysvětluje některé základní pojmy pro práci s videem a obrazem. Ve čtvrté kapitole jsou popsány základní úpravy obrazových dat a pojmy s nimi spojené. Páta kapitola popisuje celý návrh realizace tvorby aplikace pro tuto práci, popis použitých prostředků a popis implementace výsledné aplikace. Tato kapitola se pak věnuje i testování a popisu optimalizace aplikace.

Kapitola 2

Realistická a nerealistická tvorba

V dnešní době se realistická i nerealistická tvorba využívá v mnohých uměleckých technikách. Jejich využití zahrnuje od simulací prvků reálného světa, vytváření scén v počítačových hrách, až po doplňování scén ve filmové tvorbě.

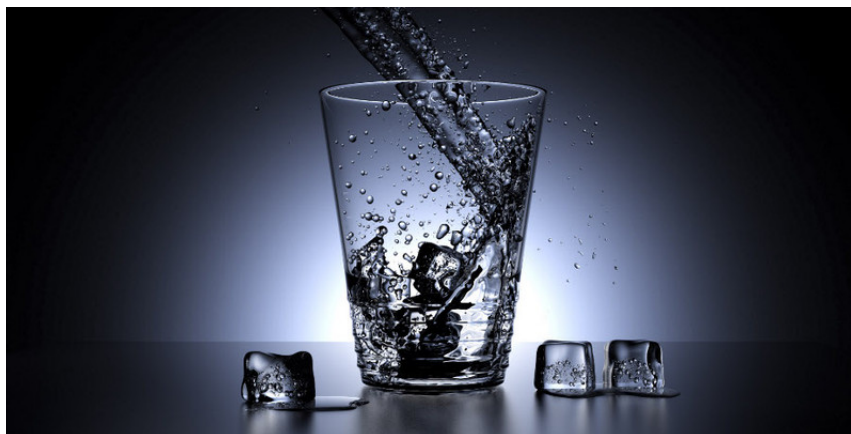
Následující část práce popisuje realistické a nerealistické zobrazování. Je důležité si uvědomit, že tato práce není encyklopedickým přehledem, ale obsahuje mnohdy jen všeobecné informace k dané oblasti. Jelikož je tato práce zaměřena na nerealistické zobrazování, bude realistické zobrazování jen stručně shrnuto. U nerealistického zobrazování, kterému se věnuje tato práce budou vysvětleny i různé způsoby provedení tohoto zobrazování. Toto ale není všechno co do realistické a nerealistické tvorby patří. Pro tuto práci však základní popis obou metod postačuje.

Výhod nerealistického zobrazování oproti realistickému je mnoho. Zjednodušení zobrazené scény umožňuje zaměřit se na jiné, méně patrné detaily. V architektuře se nerealistická tvorba využívá z důvodu lepší orientace v prostoru. Pozorovatel nezkoumá detaily, ale zaměřuje svou pozornost na tvary a všeobecný dojem ze scény. Nezáleží tedy na detailech jako je například to, zda postava v scéně má správné proporce pro člověka, nebo jaké barvy je tapeta na stěně. Využitím nerealistického zobrazování je možné docílit vytvoření požadovaných pocitů u pozorovatele.

2.1 Realistické zobrazování

Realistické zobrazování [1], anglicky označované pojmem Photorealistic Rendering je v dnešní době odvětví v počítačové grafice, které je jako téma velmi rozšířené. Snahou tohoto zobrazování je zobrazení 2D nebo 3D scény tak, aby byla nerozpoznatelná od reálného světa. Data jsou parametry popisující reálný nebo imaginární svět s jeho objekty a jejich vlastnostmi. Scény vytvořené tímto zobrazováním mají využití například ve filmové tvorbě, počítačových hrách, architektuře či různých typech simulací reálného světa.

Metoda se zaměřuje hlavně na vytvoření co nejpřesnějších detailů zobrazované scény. Kritériem je například tvar objektů, stínování, imitace vlastností materiálu nebo co nejpřesnější napodobení odrazu světla na objektech. Vytvoření výsledné scény se skládá z vytvoření objektu, na který se následně aplikuje použití mnohých textur a filtrů. Kvalita tohoto zobrazování dosáhla velmi vysoké úrovně. Mnohdy člověk, kterému není řečeno, že se jedná o počítačovou tvorbu, nepozná rozdíl s fotografií reálného světa.



Obrázek 2.1: Ukázka realistického zobrazování ¹

2.2 Nerealistické zobrazování

Při zobrazování scén nastává v některých případech k problému, že zobrazovaná scéna má při velkém množství detailů kterých se realistickým zobrazováním nedá dosáhnout. Nerealistické zobrazování, anglicky také známe jako Non-Photorealistic Rendering slouží k omezení detailů zobrazované scény. Pro toto zobrazování a celkově tuto tematiku v oblasti grafiky vzniklo mnoho knih a zdrojů [16] nebo [4].

Jedním z dalších možným důvodem použití tohoto zobrazování může být například zjednodušení modelu reálného světa. To znamená, že složitost scény je pro výsledné zobrazení pro dané účely nepotřebná, popřípadě nežádoucí. Tato technika zobrazování zanedbává pro nás nepotřebné detaily a zaměřuje se na zvýraznění detailů pro danou zobrazovací techniku důležitých. Tyto detaily můžou být například zvýraznění barvy, obrysů objektů nebo deformace objektu.

Velmi používanou metodou nerealistického zobrazování je animovaná tvorba. Zobrazit objekty reálného světa v dětské pohádce by dítěti nepřineslo radost, ale spíš by způsobilo zděšení. Možnost rozšíření barev v animacích je dnes pro rodiče skvělým prostředkem pro výuku malých dětí. Realistické ilustrace v naučných knihách nebo pohádkách jen málokdy povzbudí dítě v pokračování čtení nebo sledování pohádky. Stejným důvodem jako v animované tvorbě je i využívání nerealistického zobrazování například pro ilustrace knížek. Při ilustracích pro fantasy nebo sci-fi knížky často nastává situace, že daná scéna neodpovídá realitě a je proto nutnost použít nerealistického zobrazování.

Tato práce se ale věnuje převodu realistické tvorby na tu nerealistickou, přesněji převedení videa na video napodobující komiksovou kresbu. Techniky používané pro toto zobrazování jsou třeba imitace malby štětcem, komiksově tvorby, nebo perokresby. Tyto techniky budou popsány podrobně v následující kapitole.

2.3 Umělecké techniky nerealistické tvorby

I když nerealistické zobrazování patří k poměrně mladým odvětvím počítačové grafiky, existuje už poměrně velké množství uměleckých technik pro tento typ zobrazování. Příkla-

¹obrázek dostupný na internetovém zdroji: <https://www.blender.org/features/>

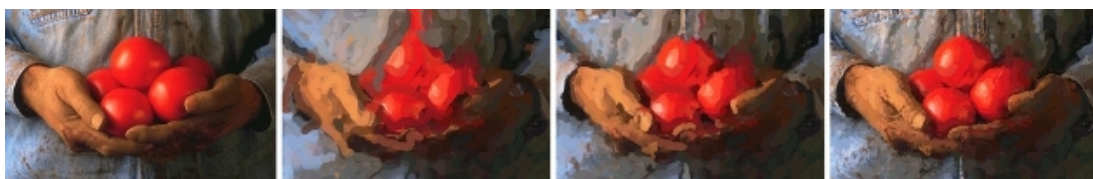
dem těchto uměleckých technik může být třeba technika malby štětcem nebo úprava scény komiksovými efekty.

Technika malby štětcem

Tato technika využívá jako předlohu při návrhu reálné tahy štětcem při malování. Základem této techniky je potenciál různých typů štětců a hlavním principem je simulování tahů štětcem. Tah štětce je reprezentován texturou a maskou. Postup při vytváření finální scény touto technikou je následující:

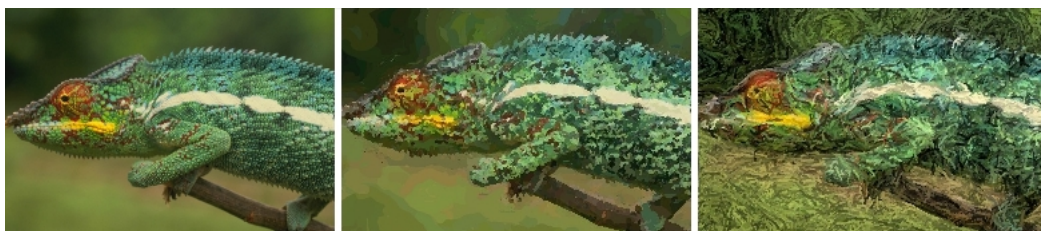
1. Tlustým štětcem se projde celá scéna. V tomto bodě jsou zanedbané veškeré detaily scény.
2. O něco užším štětcem se vykreslí jemnější detaily. Princip této činnosti spočívá v tom, že se prochází jen místa kde nastal přechod barev ve vstupní scéně nebo byla detekována hrana. Místa bez větších změn modifikována nejsou.
3. Úprava scény pokračuje opakováním alternativy bodu 2., kde tloušťka štětce závisí na detailech vstupní scény.

Na následujícím obrázku 2.2 je zobrazen postup při použití této umělecké techniky. Zleva: vstupní obrázek, znázornění použití bodu 1. z postupu uvedeného výše, úprava násobného použití bodu 2., finální úprava obrázku.



Obrázek 2.2: Postup aplikování techniky malby štětcem [6]

Změnou parametrů v algoritmu zobrazování (tzn. změnou parametrů tahu štětce) se dá docílit různých malířských technik. Na obrázku 2.3 znázorněném níže jsou zobrazeny zleva doprava tyto techniky: vstupní obrázek, impresionismus a expresionismus.



Obrázek 2.3: Zobrazení různých malířských technik [5]

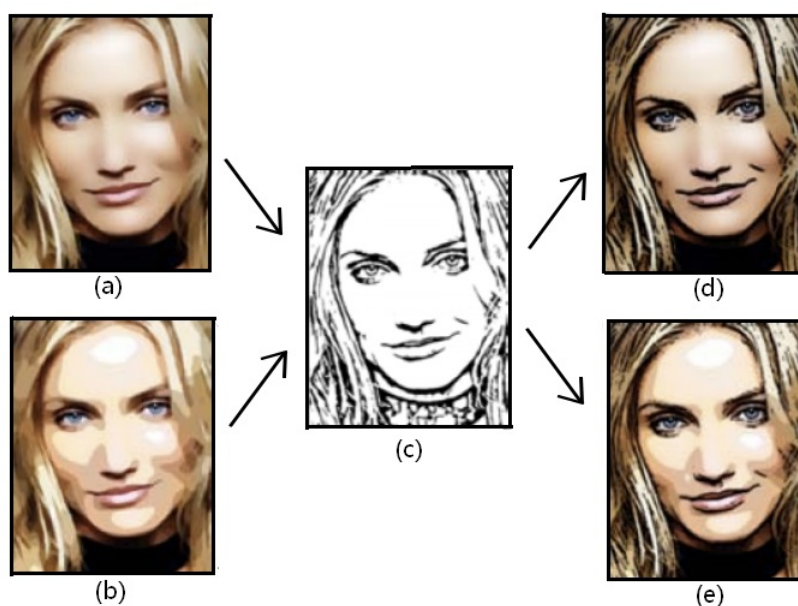
Technika komiksových efektů

Mezipaměť je uvedena na stranu klienta, aby se snížilo vytížení serveru. Při využití tohoto principu je v odpovědi od serveru uvedeno, zda je možné získaná data uložit. Takto lze omezit množství dat přenášené po síti a zároveň snížit celkové nároky na systém díky eliminaci interakce mezi klientem a serverem. Nevýhodou je však existující možnost, že data v mezipaměti se budou lišit od dat, které by klient získal při zaslání dotazu.

Animovaný film a klasický hraný film se odlišují v několika různých aspektech. Mezi ty hlavní aspekty patří omezení počtu barev a zvýraznění hran u animovaného filmu. Pro pozorovatele úbytek detailů není nežádoucí, protože se může lépe zaměřit na vnímání jiných detailů, které by v klasickém filmu nemusel zachytit. Mezi tyto detaily může patřit třeba vystihnutí nálady postavy. Toho může být docíleno úpravou barev pozadí na barvy chladnější a zesvětlením postavy. Podobné nálady u klasického filmu není možné jednoduchým způsobem dosáhnout.

Mezi techniky úpravy scén pro komiksové efekty patří například Cel Shading. Cel Shading je technika používaná převážně pro 3-D grafiku, ale lze ji použít i pro 2-D grafiku. Tato technika existuje poměrně krátkou dobu. Začala se objevovat až na začátku 21. století.

Jak může zpracování snímku vypadat je ukázáno na obrázku 2.4. V části (a) je zobrazená abstrakce vstupní snímky. Úkolem abstrakce je zjednodušení. Znamená to zanedbání malých detailů a zvýraznění nejdůležitějších prvků snímky. V části (b) je spolu s abstrakcí na snímek použita i kvantování. Kvantováním se omezí počet barev a tím se dosáhne efektu podobnému jako by byl obrázek nakreslený. Dalším krokem je detekce hran, která je zobrazena v části (c). Na detekci hran se využívají techniky jako například rozdíl gaussovských rozostření a poté zvýraznění hran prahováním, nebo Sobelův operátor opět zakončený prahováním. V části (d) je výsledný efekt úpravy snímku bez kvantování a v části (e) výsledný efekt s kvantováním.



Obrázek 2.4: Postup při vytváření komiksových efektů [15]

Pro úpravu videa je tato technika na rozdíl od techniky malby štětcem vhodnější. Při technice malby štětcem je větší problém zabezpečení koherence mezi snímky. Tento problém se projevuje tak, že jelikož je každý snímek upravován nezávisle na ostatních, tak mohou být tahy štětce pro každý snímek znázorněny jiným směrem. Ve výsledném upraveném videu to způsobí velké změny mezi jednotlivými snímky.

Na rozdíl od této techniky má technika komiksové kresby výhodu, protože rozdíl mezi vstupním realistickým snímek a snímek upraveným touto technikou je poměrně malý. Zanedbávají se detaily a snižuje se počet použitých barev. Přejechy mezi snímky jsou plynulejší a pro pozorovatele příjemnější na sledování. Problém ale mohou způsobit ostré

hrany ve snímcích, které mohou způsobit blikání a viditelné přeskoky ve videu. Možným řešením je úprava barvy méně důležitých hran. Když se použijí místo černé barvy světlejší, ideálně odstíny šedé nebo hnědé, zjemní se i přechody mezi snímky.

2.4 Existující práce s nerealistickým zobrazováním

Úprava videa nerealistickým zobrazováním je poměrně novým odvětvím v počítačové grafice, jeho využití přesto nabývá každým dnem větší vážnost. Existuje zatím jen malé množství aplikací převádějících realistickou tvorbu na tu nerealistickou. Bylo zatím zveřejněno větší množství prací nebo výukových prostředků vysvětlujících tuto tematiku a použitelné algoritmy [9] nebo [3].

Komiksová tvorba

Tato práce je zaměřená na komiksovou kresbu, přesněji řečeno na převod realistické na nerealistickou komiksovou tvorbu. Na obrázku 2.5 je vidět kresba z komiksu The Darkness [2]. Jak už bylo popsáno výše, zanedbání detailů a využití prvků nerealistické tvorby činí komiksovou kresbu zajímavou. Umožňuje vyobrazit scény, které se v reálném světě nemohou stát. Zanedbání detailů jako je stínování určitých detailů umožňuje zvýraznění jinak nevýrazných částí kresby. Tento jev je na už zmiňovaném obrázku vidět například na kouři z doutníku nebo na ostroty obrysů budov v pozadí.



Obrázek 2.5: The Darkness [2]

Počítačová hra

Pro grafickou stránku počítačových her není nerealistická tvorba úplně typickou metodou zobrazování. Většina vývojářů se v dnešní době snaží co nejvíc přizpůsobit svojí tvorbu reálnému světu. Najde se ale mnoho případů, kdy je usouzeno, že realistická tvorba není pro konkrétní hru nejvhodnější. Takovým případem se stala i hra Borderlands. Využitou metodou pro vykreslení scén byla metoda cel shading. Touto metodou se docílilo, že vykreslená scéna byla jakoby kreslená. Jak je vidět na obrázku 2.6, zanedbání detailu a přizpůsobení scény nerealistické tvorbě pomohlo udržet hru zajímavější. Mnoho detailů, jako jsou například linie zahnutí oblečení nebo tvarů zbraní jsou tak pro hráče poutavější.



Obrázek 2.6: Borderlands ²

Filmová tvorba

Jelikož tato práce je zaměřena na převod realistického videa na video nerealistické, je potřeba vzpomenout i tento typ existující práce. Takovýmto příkladem je film *A Scanner Darkly* natočen v roce 2006. Natočen byl tradičním způsobem kamerou a dodatečně byly jednotlivé snímky počítačově upravovány tak, aby obraz působil jako komiksová kresba. Na obrázku 2.7 je ukázán snímek z tohoto filmu.



Obrázek 2.7: A Scanner Darkly

Popsané práce samozřejmě nejsou jedinými existujícími pracemi a metody nerealistického zobrazování se využívají v mnohých dalších odvětvích produkce zaměřené i na grafickou stránku. Pro tuto práci však je jen malá ukázka využití nerealistického zobrazování v praxi postačující.

²hra Borderlands dostupná na: www.borderlandsthegame.com

Kapitola 3

Zpracování videa a obrazových dat

Účelem videa jako média je promítat nějaká obrazová data. Nejlepším způsobem jak upravit video je vzít každý jeho snímek a provádět úpravy na jednotlivých snímcích zvlášť. U ne-realistického zobrazování je ale problémem u zpracování videa nekoherence snímku. Jelikož je každý snímek upravován samostatně, nastává mezi nimi ve výsledném videu k blikání nebo různým typům šumu.

Ve zpracování signálů je jako filtr označen proces odstranění nechtěných složek ze signálu. V oblasti zabývající se zpracováním obrazových dat jde především o matematické operace aplikované nad signálem, kterým je v tomto případě jeden obrázek, posloupnost obrázků anebo video. V takovém případě je obrázek uvažován jako dvoudimenzionální signál - obraz má výšku a délku. Cílem takových úprav je například rozostřit obrázek, změnit jeho jas a kontrast anebo odstranění rušivých podnětů.

Filtry jsou klíčovým prvkem při nerealistickém zobrazování, protože jejich použití představuje primární způsob úpravy obrazu. Znamená to tedy základní princip použití různých technik používaných v tomto typu zobrazování

V této kapitole budou postupně popsány barevné modely a uspořádání barev a úpravy-filtre používané na obrazové data. K demonstračním účelům budou filtry v této sekci aplikovány na následující obrázek 3.1.



Obrázek 3.1: Obrázek beze změn¹

¹Obrázek 3.1 převzat z: <https://www.pinterest.com/pin/573434965030970834/>

3.1 Barevné modely a uspořádání barev

Barva, kterou lidské oko vidí není vlastností vnímaného objektu. To jakou člověk vidí barvu závisí na několika aspektech: kvalita a intenzita světla, materiál objektu, stav adaptace zrakového aparátu (například při přechodu z tmavé místnosti na světlo). Nejzajímavějším aspektem je to, či je člověku objekt znám. Když konkrétní nebo podobný předmět zná, má vytvořenou asociaci s barvou známého předmětu.

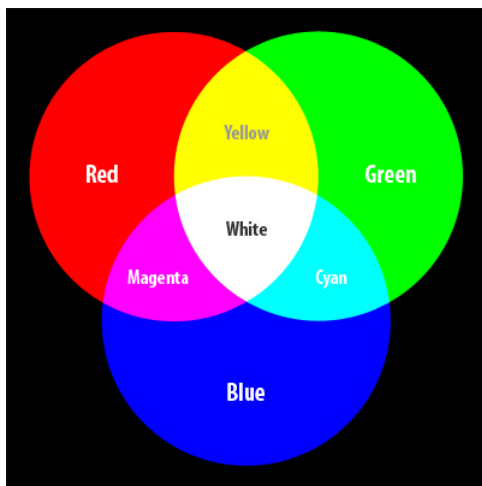
Pro popis barvy byli vytvořené barevné modely, které využívají jen určité barvy a mícháním těchto barev lze získat zbývající barvy. Možných kombinací je hodně, lidské oko však nedokáže vnímat všechny. Různé barevné modely se snaží co nejvěrněji napodobit barvy, v praxi se ale používají modely, u kterých je vhodný poměr mezi přesností barevného dojmu na jedné straně a složitostí modelu a obtížností jeho použití na druhé straně. V praxi se používá několik barevných modelů.

- RGB
- CMY (CMYK)
- HSV
- HSL

Nejpoužívanějšími jsou však modely RGB a CMYK. Tyto dva modely jsou proto podrobněji vysvětleny níže.

Barevný model RGB

Model RGB je v počítačové grafice nejrozšířenější barevný model. Jeho základní barvy jsou červená (red – R), zelená (green – G) a modrá (blue – B). Všechny ostatní viditelné barvy jsou kombinací těchto třech základních barevných složek. Barvy v tomto modelu se dají vyjádřit v desítkové soustavě jako čísla od 0 do 255. Červená barva se tak dá jednoduše zapsat jako (255, 0, 0), kde zastoupení zelené a modré barvy je nulové. Kombinace těchto troch čísel určuje tedy jak barvu, tak intenzitu světla.



Obrázek 3.2: Barevný model RGB

Pro tento model je charakteristické sčítání barevných složek. Jedná se tedy o aditivní barevný model. Jak je vidět na obrázku 3.2 součtem všech barev v jejich největších intenzitách (v desítkové soustavě hodnota 255) vzniká barva bílá. Naopak pokud není zastoupená v barvě žádná z troch základních barev, jedná se o barvu černou.

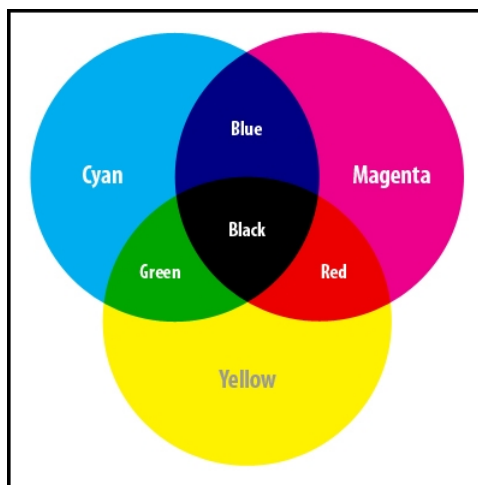
Pro tuto práci je dobré popsat i rozšíření tohoto modelu. Tímto rozšířením je doplnění modelu RGB o další barevný kanál, kterým je alfa-kanál. RGBA, kde písmeno A zastupuje alfa-kanál však už není samostatným barevným modelem. Alfa-kanál reprezentuje průhlednost daného pixelu barvy.

Tento barevný model se využívá zejména v zařízeních, kde se na tmavý podklad přidávají světlo a barvy. Takovými zařízeními jsou třeba televize, monitory. Využití se najde ale i například při vytváření fotografií.

Barevný model CMY

Barevný model CMY je založen na subtraktivním způsobu vytváření barev. To znamená, že na rozdíl od modelu RGB přidávání sytosti základních barev tohoto modelu vzniká tmavší barva. Zastoupením všech základních barevných složek v nejvyšší intenzitě tak vzniká barva černá a naopak když barevné složky zastoupené nejsou vzniká barva bílá.

Základními barvami tohoto modelu jsou tyrkysová (cyan – C), purpurová (Magenta – G) a žlutá (yellow – Y). Tyrkysová je doplňková barva k červené a proto odráží všechno světlo vyjma červeného. Podobně purpurová je doplňková k zelené a žlutá k modré. Pomocí CMY barev je tak možné řízeně snižovat intenzitu RGB světla a tak docílit barev inverzně - ubíráním bílé.

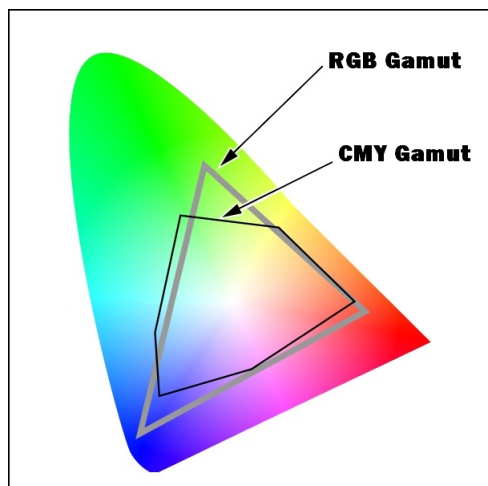


Obrázek 3.3: Barevný model CMY

Využití tohoto modelu je vhodné hlavně u reprodukčních zařízení, kde je podklad bílý a barvy nanášené na podklad se tvoří mícháním barevných složek. Složením základních barev modelu CMY ale při použití běžných tiskových barev v současnosti není možné vytvořit velmi kvalitní černou barvu. Proto je pro tyto účely tento model rozšířen o černou barvu. Model CMYK, kde K značí černou (Key black) barvu, skládá do výsledné barvy pixelu kromě základních barev i barvu černou. Jejím přidáním se docílí kvalitní černé barvy a navíc se snižují náklady na tisk, jelikož černý inkoust je levnější než barevný.

Gamut

Pojmem gamut označuje dosažitelnou oblast barev v určitém barevném prostoru. Pomocí gamutu se dají zobrazit oblasti zobrazitelné jednotlivými barevnými modely nebo schopnost zařízení zobrazit určitou barvu. Například, jak je vidět na obrázku 3.4, čistá modrá barva obsažená v barevném modelu RGB není zobrazitelná v modelu CMY. Proto se o této barvě dá říct, že je mimo gamut barevného modelu CMY.



Obrázek 3.4: Gamut

Barvy které se nenacházejí v gamutu určitého modelu nebo zařízení se dají zobrazit jen přibližně. Při reprodukci například digitální fotografie se zpravidla změní gamut v tom smysle, že některé barvy se z původního obrazu ztratí nebo jsou nahrazeny jinými.

Barevný prostor BGR v knihovně OpenCV

Aplikace vytvořena pro tuto práci je napsaná pomocí knihovny OpenCV. Využívá některých přechodů mezi barevnými prostory. Základním barevným prostorem pro tuto knihovnu je prostor BGR. Pracuje se stejnými barevnými složkami jako barevný model RGB. Rozdílem je uspořádání barevných složek v bitové reprezentaci. BGR je 24-bitová reprezentace barevného prostoru, kde nejnižších 8 bitů zaznamenává modrou barvu, dalších 8 barvu zelenou a nejvyšších 8 bitů barvu červenou.

Důvodem, proč knihovna OpenCV obsahuje barevný prostor BGR místo používání barevného modelu RGB je pouze z historického důvodu. V minulosti, kdy bylo používání barevného prostoru BGR populární u poskytovatelů softwarů a výrobců fotoaparátů, vývojáři OpenCV tento barevný prostor zvolili jako výchozí [11].

Stejně jako barevný model RGB má i barevný prostor BGR možnost rozšíření o alfa-kanál. Pomocí tohoto kanálu lze zadefinovat průhlednost pro jednotlivé pixely snímku.

3.2 Konvoluce u úprav videa

Konvoluce je matematická operace nad dvěma vstupními funkcemi, přičemž vznikne třetí funkce, na kterou lze nahlížet jako na upravenou verzi jedné z původních vstupních funkcí. Při zpracování obrazu je použita diskrétní konvoluce. Zde se vyskytují dvě pole čísel, často různých velikostí, ale stejně se stejnými dimenzemi. Pro modifikaci obrazu je poté použito

konvoluční jádro s hodnotami, které mají dosáhnout žádaného efektu a samotná obrazová data. Výsledkem je lineární kombinace určitých vstupních pixelů. Použitím konvoluce lze dosáhnout efektů jako například rozostření obrazu, detekce hran nebo zaostření.

$$b[i, j] = \sum_{y=i-r}^{i+r} \sum_{x=j-r}^{j+r} f[x, y] * w[y, x] \quad (3.1)$$

Rovnice 3.5: Všeobecný vzorec pro diskrétní konvoluci

3.3 Rozostření

Digitální rozostření obrazu se typicky využívá pro odstranění šumu nebo nežádoucích artefaktů, lze jej však využít i pro implementaci některých dalších metod zpracování obrazu, jako například u detekcí hran. Jednoduchá rozostření se typicky implementují pomocí jednoduché konvoluce s obrazovými daty.

Gaussovské rozostření

Gaussovské rozostření vznikne použitím Gaussovské funkce nad obrazovými daty. Používá se k odstranění šumu a odstranění detailu. Výsledkem je jemné rozmazání, které je parametrizováno poloměrem zahrnutých obrazových bodů. Tím je dosaženo efektu, kdy obraz vypadá jako by se pozorovatel díval skrze průsvitné sklo.

Gaussovské rozostření se implementačně nejsnáze dosáhne konvolucí obrazu s konvolučním jádrem tvořeným Gaussovými hodnotami. V praxi se používá dvouprůchodová technika, kdy v prvním průchodu se obrazová data rozostří pouze v jednom směru (v horizontálním nebo vertikálním) zatímco v dalším průchodu se rozostří ve vynechaném směru. Takto se ušetří výpočetní čas, než kdyby se použila konvoluce s dvojrozměrným jádrem.

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \quad (3.2)$$

Rovnice 3.6: Jednodimenzionální výpočet Gaussovského rozostření



Obrázek 3.7: Gaussovo rozostření s poloměrem pixelů sedm

Box Blur

Box Blur je jednoduché rozostření, kde se využívá lineárního filtru. Pomocí konvoluce se každý pixel ve výsledném obrazu stane průměrnou hodnotou okolních pixelů. Pomocí Box Blur lze aproximovat i gaussové rozostření.

Typickou maticí používanou v Box Blur je ukázan na následující matici:

$$M = \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$$

Matice 3.8: Matice používaná v Box Blur



Obrázek 3.9: Box Blur

3.4 Redukce barev

Redukce barev je skupina úprav, která určitým způsobem omezuje paletu barev ve výstupním obrazu. Důvodem může být například úspora místa, ale jsou i další oblasti počítačové grafiky, kde se dají tyto změny využít. Nejpoužívanějšími úpravami jsou kvantování, převod obrazu do stupňů šedi a nebo převod obrazu na černobílý.

Kvantování

V oblasti zpracování obrazu se kvantováním označuje proces, kdy je v obrázku snížen počet unikátních barev. Typicky je snaha o to, aby modifikovaný obraz byl co nejvíce podobný původnímu obrazu. Používá se pro komprimaci obrazových dat anebo v uměleckých technikách pro dosažení různých efektů, které se poté dále kombinují s jinými algoritmy a filtry.



Obrázek 3.10: Obrázek po aplikaci kvantování

Převod do stupňů šedi

Obraz, který je reprezentován ve stupních šedi má pro každý pixel nastavenou hodnotu jasu. Pixel s nejsilnější intenzitou jasu je představen bílou barvou, zatímco ten s nejslabší intenzitou černou barvou. Všechny úrovně intenzity mezi bílou a černou barvou jsou popisovány jako stupně šedi.

Převod do stupňů šedi je úprava obrazu, která mění barevné spektrum obrazu do příslušného odstínu šedi. Existuje je několik způsobů jak této úpravy dosáhnout. Následující postupy popisují převod obrazu z barevného modelu RGB na obraz reprezentující stupně šedi. První možnost je vybudována na jednoduchém způsobu, který zprůměruje nejvíce zastoupenou barvu a nejméně zastoupenou barvu podle vzorce níže.

$$Y = \frac{(\max(R, G, B) + \min(R, G, B))}{2} \quad (3.3)$$

Rovnice 3.11: Zprůměrování nejvíce a nejméně zastoupené barvy

Další možností je vypočítat prostý průměr zastoupení barev:

$$Y = \frac{(R + G + B)}{3} \quad (3.4)$$

Rovnice 3.12: Prosté zprůměrování barevných složek

Nejvhodnější způsob je však použití vzorce respektujícího citlivost lidského oka. Lidské oko je nejvíce citlivé na zelenou složku barevného spektra a proto je koeficient násobící tuto složku nejvyšší. Na modrou složku je naopak citlivé nejméně, proto je koeficient nejnižší.

$$Y = 0.2126R + 0.7152G + 0.0722B \quad (3.5)$$

Rovnice 3.13: Výpočet pomocí koeficientů



Obrázek 3.14: Převod do stupňů šedi

Prahování

Prahování (anglicky thresholding) je metoda, která slouží na převod obrazu reprezentovaného v stupních šedi na binární obraz. Binárním obrazem se rozumí obraz složený jen z dvou barev, černé a bílé. Algoritmus prahování spočívá ve volbě prahovací hodnoty, jinak řečeno prahu. Hodnoty pixelů nad hodnotou prahu jsou následně považovány za bílou barvu a naopak hodnoty pod hodnotou prahu jsou změněny na barvu černou.



Obrázek 3.15: Výsledek prahování

3.5 Detekce hran

Detekce hran se v zpracování obrazu označuje za algoritmus určený k vyhledání skupin pixelů, u kterých dochází k výrazné změně jasu. Teoretickou hranu lze hledat na hranicích objektů a nebo při přechodu světla do stínu. Dokonalé detekce však nelze úplně dosáhnout, protože reálná hrana může být v obrazu příliš obklopena šumem. K vyhledání hran jsou často používány různé filtry a nebo jejich kombinace, které se typicky konvolují s obrazovými daty. Po aplikaci jedné z těchto metod se většinou pro zbavení šumu používá prahování.

Sobelův operátor

Sobelův operátor nebo též Sobelův-Feldmanův filtr je v oblasti zpracování obrazu používán k detekci hran. K této operaci jsou použity dvě matice o rozměrech 3x3. Princip je založen na výpočtu aproximací parciálních derivací pro obrazová data. Filtr je tedy směrově závislý. Existuje několik různých variant matic, kde každá z nich detekuje hrany v jiném směru. Aplikace filtru je nenáročná a je realizována jako jednoduchá diskrétní konvoluce s obrazovými daty.

Matice pro detekci hran ve směru k horní hraně zobrazena následovně na matici:

$$M = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Matice 3.16: Matice používána pro detekci hran



Obrázek 3.17: Sobelův filtr se zachováním barev

Rozdíl Gaussovských rozostření

Rozdíl gaussovských rozostření je založen na principu odečtení dvou obrázků, na které byly předtím aplikovány gaussovske rozostření s různou směrodatnou odchylkou. Většinou je vhodné převést obrázek do odstínů šedi a poté konvolucí získat dvě verze obrázku s různými gaussovskými rozostřeními. Tímto způsobem lze předběžně zvýraznit hrany obrázku. Lepšího efektu se dosáhne, pokud je poté obrázek ještě podroben prahování, který ještě lépe zvýrazní hrany.



Obrázek 3.18: Gaussovských rozostření s inverzí barev a úpravou jasu

Kapitola 4

Realizační část práce

Tato kapitola je zaměřená na stručnější popis při tvorbě výsledné aplikace, která vstupní video převede na video podobající se komiksové kresbě. Budou postupně popsány použité nástroje, vlastní implementace a podrobněji budou vysvětleny některé zajímavější části aplikace.

Uživatel pomocí vytvořené aplikace bude moci načíst libovolné video z disku, následně vybrat video, které bude přepsáno novým upraveným videem nebo zadat název pro nové video vytvořené aplikací. Po načtení vstupního a výstupního videa může zvolit automatický převod nebo případně upravit parametry některých změn, které pak budou aplikovány na nové video. V případě úpravy vstupních parametrů bude součástí okna i ukázka takto upraveného vstupního snímku. Po převodu videa na video s efekty komiksové kresby je k dispozici možnost přehrání obou - jak vstupního tak výstupního videa. Po ukončení procesu převodu si uživatel může nanovo upravit vstupní parametry na už načteném videu nebo nahrát nové vstupní a výstupní videa a převod opakovat.

4.1 Návrh řešení a motivace

Pracovat na této aplikaci, konkrétně komiksové úpravě videa přináší mnoho faktorů, které motivují k jejímu dokončení. Jedním z nich je například to, že oblast převodu realistického videa na video nerealistické tvorby je poměrně nová a velmi vyhledávaná.

První věcí na kterou se bylo při hledání řešení této práce zaměřit, bylo najít nejvhodnější programovací jazyk, kterého využití by pomohlo při práci s videem a jeho snímky. Nejvhodnějším kandidátem se stalo C++. Tento jazyk byl zvolen z několika důvodů. Prvním byla snaha rozšířit si znalosti v tomto jazyku. Druhým to, že jsou k němu vytvořené knihovny pro přímou práci a editaci videí a obrazových dat. V neposlední řadě se jedná o jazyk, který není interpretovaný, proto se nachází v jiné výkonnostní kategorii než jazyky skriptovací. Pro zpracování obrazových dat je poté zde dostupná knihovna OpenCV, která poskytuje funkce přímo využitelné pro tuto práci. Použití některých funkcí bude popsáno později.

Požadavkem pro tuto práci bylo i jednoduché GUI (grafické rozhraní) aplikace. Využití programovacího jazyka C++ tak poskytlo další výhodu. Knihovna QT implementovaná pro tento jazyk obsahuje funkce pro jednoduché vytváření GUI. Pro práci s touto knihovnou jsou vytvořeny vývojové prostředí, které jsou přizpůsobené na práci na aplikaci obsahující grafické uživatelské rozhraní. Aplikace bude implementovaná v prostředí QT Creator, které obsahuje možnost editace WYSIWYG (What you see is what you get), tedy programátor

už při návrhu rozhraní přesně vidí, jak bude výsledná aplikace vypadat.

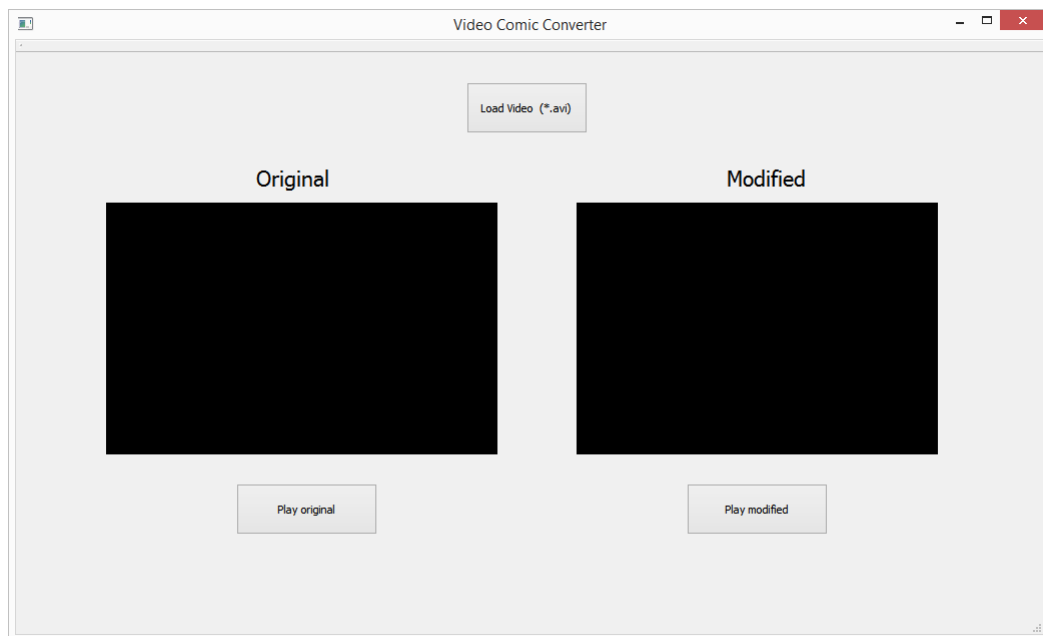
Cílem práce je upravit video. To znamená pracovat s jeho jednotlivými snímky. Prvním bodem úprav je kvantování, která omezí počet barev snímků. Dalším je detekce hran z původního snímku a následně překrytí těchto dvou úprav. Problém, který při implementaci může nastat je nekoherence mezi snímky, tedy viditelné nevyžádané změny mezi jednotlivými snímky, které budou pozorovatele rušit. Zda tento problém nastane už záleží na použitých algoritmech. Výběr úpravy na komiksovou kresbu by tento problém měl z větší části obejít.

4.2 Knihovna QT

Knihovna QT je multiplatformní open-source softwarová knihovna, implementovaná v programovacím jazyku C++, která umožňuje a hlavně zjednodušuje práci na grafickém rozhraní aplikace. Pomocí QT je jednoduché vytvářet objekty a pracovat s nimi při implementaci funkcionality aplikace. Objekty jsou prvky, které se uživateli zobrazí, nebo to jsou prvky sloužící pro správný běh aplikace. Může jít třeba o tlačítko, nebo různé informační okna které vytváření výsledné grafické rozhraní aplikace. Práce s objekty probíhá pomocí signálů a slotů, kterými se dá přesně specifikovat co aplikace má udělat při interakci uživatele s objektem.

Knihovna QT umožňuje jednoduchý vstupní a výstupní mechanismus pro uživatele pro načítání a uložení nově vytvořeného videa. Práce s touto knihovnou byla velmi zjednodušena kvalitní dokumentací [17]. Na obrázku 4.1 je vidět aplikace v průběhu její implementace. Objekty vytvořenými pomocí knihovny QT jsou hlavní okno aplikace, tlačítka různých funkcí, popisky a nejdůležitější jsou zobrazovací plochy pro videa.

Pro možnost přehrání videa jak vstupního, tak vytvořeného aplikací, bylo potřeba naimplementovat přehrávač s jednoduchým ovládáním. Základem je výběr vstupního videa, následuje volba, popřípadě možnost vytvoření výstupního videa. Po skončení části aplikace, která zabezpečuje úpravu videa, je umožněno uživateli spustit obě videa. Je tady i možnost pozastavení obou videí. Problémem při implementaci bylo spojení přehrávání do jednoho vlákna, díky čemu by se obě videa ovládaly jedním tlačítkem a přehrávaly se najednou. Z tohoto důvodu si musí uživatel spustit videa jednotlivě. Jedno je vždy přehráno se zpožděním od toho druhého. To představuje nevýhodu při porovnávání snímků. Implementace přehrávače je převzatá ze stránky The Coding Exodus [14], která poskytuje několik algoritmů a rad z oblasti využití knihoven QT a OpenCV.



Obrázek 4.1: Uživatelské rozhraní aplikace

4.3 Knihovna OpenCV

OpenCV (Open Source Computer Vision) je multiplatformní knihovna funkcí pro práci s obrazem. Je zaměřena na počítačové vidění a zpracování obrazu v reálném čase. Většina práce na úpravě videa a snímků videa v této práci je prováděná právě funkcemi z této knihovny. Nevýhodou této knihovny je, že neumožňuje práci na videu se zvukem. Vytvořené video aplikace je bez zvukové stopy. Pro účel této práce zvuk nehraje roli a proto je tato část videa při jeho vytváření zanedbaná. Pro tuto knihovnu je kvalitně vytvořená dokumentace, která obsahuje i ukázky využití jednotlivých funkcí [7]. To práci výrazně zjednodušilo.

4.4 Zpracování vstupního videa

Základním krokem pro možnost úpravy videa je video načíst a získat z něj různé potřebné informace, jako například počet snímků za sekundu. Pro tento úkol je vhodným nástrojem třída *VideoCapture* obsažená v OpenCV knihovně. Získávání jednotlivých snímků z videa pro následné úpravy je možné vícero způsoby. Například přetížením operátoru » nebo využitím metody *grab*.

Pro uložení upraveného videa je použita třída *VideoWriter*, taky z knihovny OpenCV. Pro postupný zápis snímku je podobně jako pro jejich získání přetížen operátor «.

Vzhledem k tomu, že jazyk C++ je založen na programovacím idiomu RAII (Resource Acquisition Is Initialization), který je spjatý s délkou života objektu, to znamená naalokování zdroje (acquisition) je provedení během jeho vytvoření, přesněji inicialiaci pomocí konstruktoru, zatímco jeho dealokace-uvolnění je provedeno během destrukce, neboli dealokace pomocí destrukturu [18]. Z tohoto důvodu není potřeba uvolňovat po práci paměť od použitých funkcí/objektů, jako tomu je například v programovacím jazyku C, pro který je taky knihovna OpenCV naimplementována.

4.5 Zpracování jednotlivých snímků videa

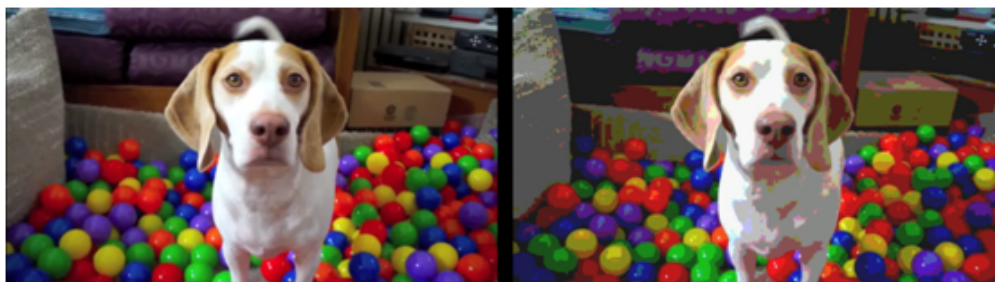
Načtený snímek je pro účel zpracování potřebné uložit. třída *Mat*. Objekt třídy *Mat* je n-dimenzionální pole, které dokáže ukládat obrazové data. Tato třída je taky získaná z knihovny OpenCV a poskytuje mnoho metod užitečných pro práci s obrazovými daty. Jedná se například o klonování vstupního obrazu, nebo získávání šířky a výšky obrazu. Právě tyto rozměry jsou dále v implementaci použity při procházení všech pixelů snímku pro jejich úpravu algoritmem kvantování.

Principem úpravy jednotlivých snímků je vytvoření nových snímků s jednotlivými úpravami a jejich následné překrytí. Z tohoto důvodu jsou vytvořeny samostatné snímky s odlišnostmi od vstupního snímku. Jednotlivými úpravami jsou například kvantování, rozostření a zvýraznění hran snímku. Tyto úpravy jsou popsány v další kapitole i s grafickou ukázkou jednotlivých upravených snímků.

Pro překrytí snímků jsou důležité úpravy obrazových modelů jednotlivých snímků. Detekce hran vytvoří černobílý snímek, kde černá barva označuje hrany objektů snímku. Je tedy potřebné bílou barvu, která je pro snímek nadbytečná zprůhlednit. Probíhá tedy změna obrazových kanálů, tak aby tato úprava byla možná. Vstupní snímek vytvořený pomocí třídy *VideoCapture* má obrazový model BGR(modra, zelená a červená). Pro vytvoření transparentnosti je konkrétnímu snímku změněn obrazový model na BGRA(modra, zelená, červená a alpha). Tento model je rozšířen o tzv alpha kanál A s informací o průhlednosti konkrétního pixelu.

4.6 Grafické úpravy jednotlivých snímků

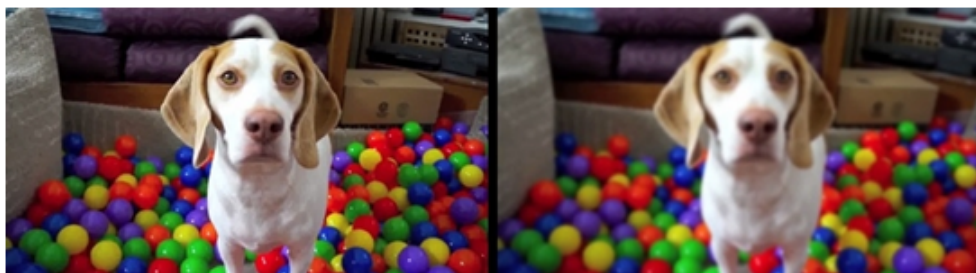
První úpravou snímku videa bylo omezení jejich detailů. Pro tento účel byl využit algoritmus kvantování, jehož popis už byl vysvětlen v kapitole 3.4. Tento algoritmus byl převzatý z knihy [10] pro samovýuku práce s knihovnou OpenCV. Pro shrnutí, účelem tohoto algoritmu je v každém snímku projít všechny pixely a omezit počet unikátních barev celého snímku. Výsledek této části úpravy videa je zobrazen na obrázku 4.2. Video na kterém jsou testování jednotlivé úpravy pochází od youtube uživatele Maymo [12].



Obrázek 4.2: Kvantování

Další úpravou snímku je rozostření obrazu. Rozostření se většinou používá pro odstranění šumu a nežádoucích prvků ze snímku, v tomto případě je použito jako samostatná úprava snímku. V návrhu řešení se s touto úpravou nepočítalo, byla ale přidána z důvodu, že v některých případech bez této úpravy měly přechody mezi snímky dost ostré změny a to bylo pro sledování rušivé. S touto úpravou snímků byl celkový výsledek změny videa podobnější komiksové tvorbě, co bylo dalším důvodem pro její použití. Pro rozostření byl

použít algoritmus gaussovského rozostření. Implementačně se jedná o použití funkce *GaussianBlur* z knihovny OpenCV. Popis tohoto algoritmu byl popsán v kapitole 3.3. Výsledek tohoto kroku je zobrazen na obrázku 4.3.



Obrázek 4.3: Gaussovské rozostření

Detekce hran je další úpravou vstupního snímku. Výsledkem této úpravy je černobílý snímek, kde černá barva zobrazuje hrany objektů snímku a bílá barva zbytek snímku. Původní snímek je v převeden prahováním do černobílého spektra barev. Tato úprava byla popsána v kapitole 3.4. Barevný snímek by zapříčinil, že nalezené hrany snímku by zůstaly také barevné. Aby bylo dosaženo efektu změny videa tak, aby se co nejvíce podobalo komiksové kresbě, je ale potřeba zvýraznit hrany tak, aby byly pouze černé barvy. Po převodu barevného spektra snímku je aplikován algoritmus nazvaný rozdíl gaussovských rozostření. Tento algoritmus už byl popsán v kapitole 3.5. Principiálně se jedná se o použití funkce *GaussianBlur*. Nad černobílým snímek jsou použity dvě verze, kde nalezení hran je získáno rozdílem výsledků těchto dvou rozostření. Konkrétní hodnoty konvolučního jádra použitého v jednotlivých rozostřeních už záviselo na subjektivním odhadu nejlepšího konečného výstupu aplikace.

Po vytvoření snímku reprezentujícího hrany objektů, je snímek ještě jednou jemně rozostřen, aby se odstranily případné nedostatky. Tímto způsobem se vyhladí například ostrost hran. Na obrázku 4.4 je zobrazen snímek upraven popsáním algoritmem.



Obrázek 4.4: Detekce hran

4.7 Překrytí snímků

Vytvořením nových snímků upravených popsánými algoritmy v kapitole 4.6 je připraveno vše pro poslední krok úpravy vstupního videa. Aby výstupní video mělo efekt komiksové kresby je potřeba nově vytvořené snímky překrýt. Pro překrytí snímků je použita funkce *addWeighted* z knihovny OpenCV. Tato funkce umožňuje nastavit procenta průhlednosti jednotlivých snímků.

Jako první probíhá překrytí kvantizovaného snímku se snímkem rozostřeným. Snímek vytvořený kvantováním obsahuje velmi ostré přechody mezi barvami snímku. Tento jev se omezil překrytím tohoto snímku se snímkem upraveným rozostřením. Rozostření kvantovaného snímku je potřeba jen pro jemné uhlazení. Proto má snímek s rozostřením nastaven viditelnost jen na 20%.

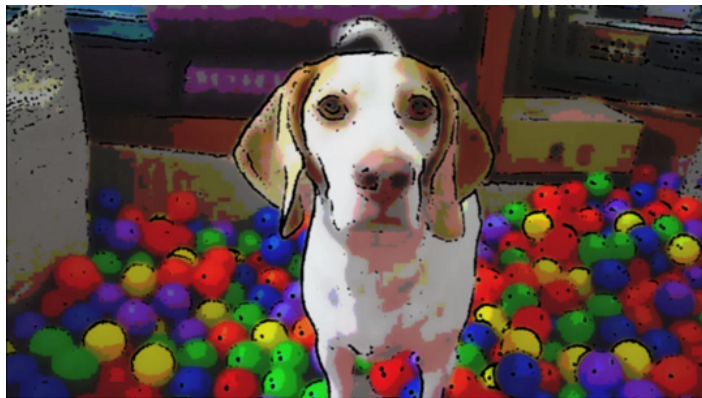
Dříve než se může nově vytvořený snímek překrýt se snímkem zobrazujícím hrany je potřebná poslední úprava snímku s hranami. Touto úpravou je zprůhlednění. Snímek vytvořen algoritmem pro detekci hran je černobílý. Bílá barva je však pro výsledné video nadbytečná a je potřeba tuto část snímku zprůhlednit. Snímek je tedy převeden do barevného modelu BGRA, který jak už bylo v kapitole 3.1 popsáno, umožňuje nastavení alpha kanálu. Přechází se každý pixel a při nalezení pixelu s bílou barvou je alpha kanál nastaven na hodnotu 0. Takto je nastavena průhlednost jen na potřebných pixelech bílé barvy.

Převodem snímku na barevný model BGRA je překrytí snímku s nově vytvořeným kvantovaným snímkem náročnější. Snímky vytvořené při zpracování vstupního videa jsou v modelu BGR. Funkce pro překrytí dvou snímků *addWeighted* a její jednodušší protějšek *add* nepodporují barevné modely s alpha kanálem a proto je jejich použití pro tento účel není možné. Byl proto využit algoritmus převzán z [8].

Výstupem po překrytí všech upravených snímků je snímek s efekty komiksové kresby.

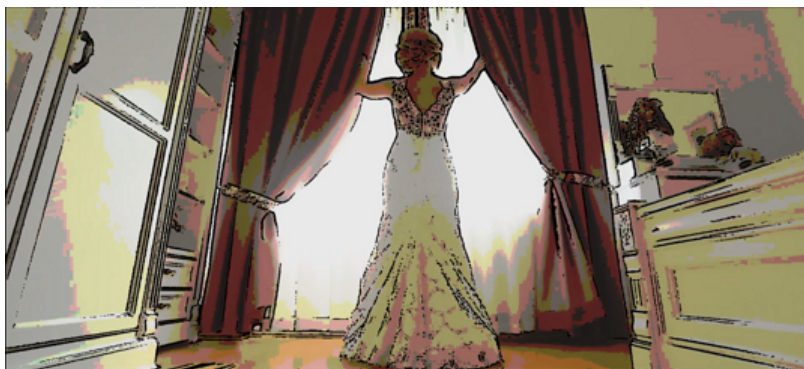
4.8 Výsledná aplikace

Konkrétní vzhled výstupního snímku lze pozměnit jednoduchou změnou hodnot parametrů úprav. Pro uživatele se jedná o citlivost detekce hran, míru rozostření nebo počet barev pro kvantování. Po vytvoření upraveného snímku jsou snímky postupně zapisovány do výstupního videa. Výstupem po subjektivním výběru nejvhodnějších hodnot parametrů je zobrazen na obrázku 4.5.



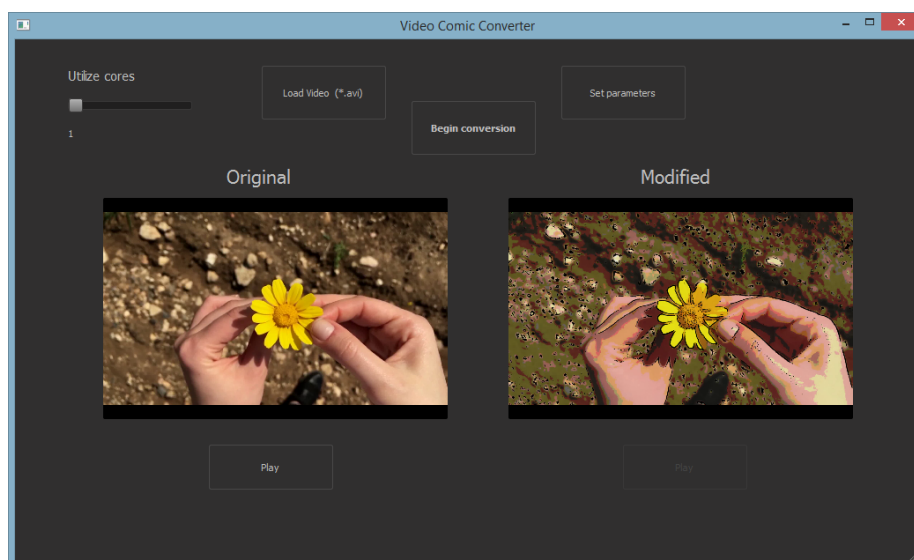
Obrázek 4.5: Výstupní snímek 1

Další zkouškou správné funkcionality převodu bylo otestování na jiných videích. Pro každé video bylo vhodné použít jiné vstupní hodnoty úprav videa. Pro některé videa bylo vhodné zesílit nebo zeslabit citlivost detekce hran, pro jiné byla vhodnější jen úprava rozostření snímků. Další ukázka převodu videa je zobrazena na obrázku 4.6.



Obrázek 4.6: Výstupní snímek 2¹

Výsledné grafické rozhraní aplikace je zobrazeno na obrázku 4.7. Uživatel při jejím spuštění má možnost jen výběrů vstupního a výstupního videa. Po této akci dostane možnost úpravy vstupních dat, nebo možnost upravit video se základním nastavením.

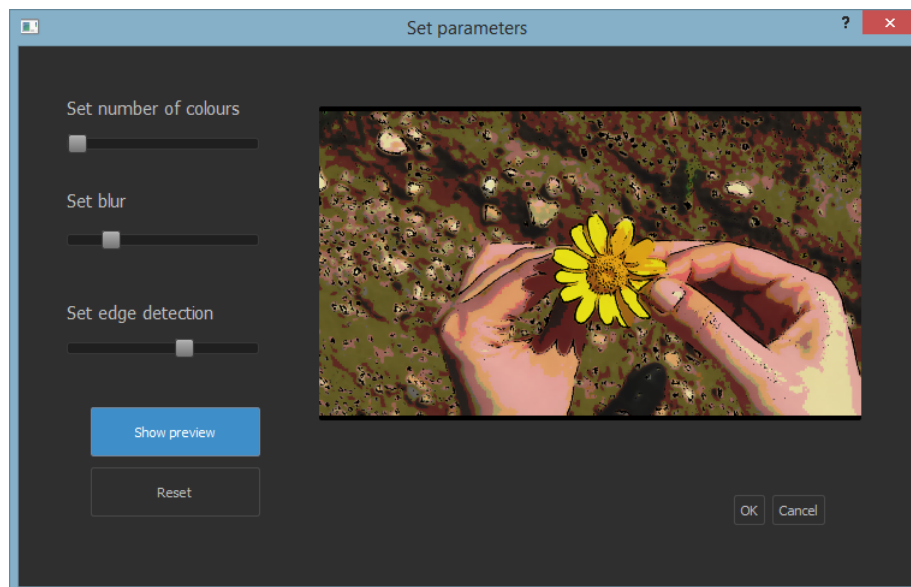


Obrázek 4.7: Finální uživatelské rozhraní aplikace ²

Při volbě změny vstupních parametrů, si uživatel jak je vidět a obrázku 4.8 může v dialogovém okně pozměnit úpravu videa podle vlastního uvážení. Tato možnost zahrnuje i ukázkou prvního snímku videa, na který jsou postupně aplikovány úpravy parametrů zadané uživatelem. Jestli volbu potvrdí tlačítkem OK, následný převod videa proběhne s těmito vstupními parametry. V jiném případě je video převedeno se základním nastavením.

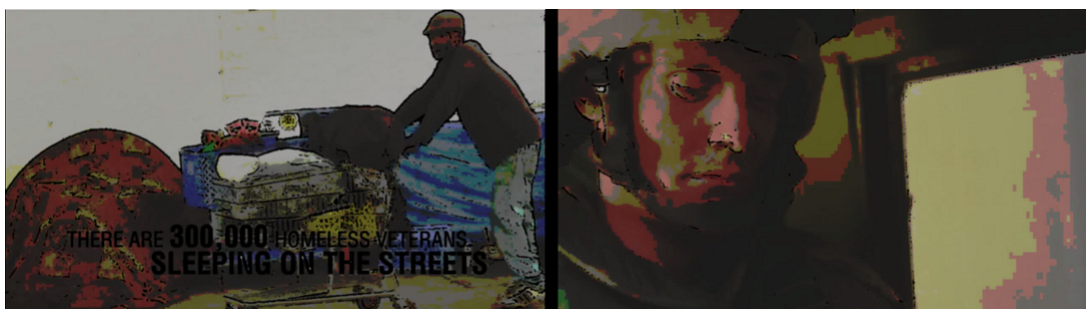
¹Obrázek 4.6 je upraven z převzatého videa z: <https://pixabay.com>

²Obrázek květiny na obrázku 4.7 je upraven z převzatého videa z: <https://pixabay.com>



Obrázek 4.8: Finální uživatelské rozhraní aplikace - změna parametrů

Zjištěným nedostatkem převodu je nekoherentnost upravených snímků při delším videu. Aplikace byla vytvářena a zkoušena na kratších videích. Průměrná délka testovaných videí se pohybovala od troch po deset sekund. Při delších videích kdy je zobrazováno vícero různých scén je potřebná pro každou scénu jiná úprava. Řešením by bylo dát uživateli možnost rozdělit video a úseky a pro jednotlivé úseky možnost zadat jiné vstupní parametry. Ukázka tohoto nedostatku je zobrazena na obrázku 4.9, na kterém jsou zobrazeny 2 různé snímky ze stejného videa. Pro část obrázku vlevo byli nastavené vstupní parametry, přesněji citlivost hran, která by však pro pravou část obrázku měla být větší. Část obrázku vpravo se tak převedla nevhodně a pro uživatele je nesledovatelná a nepříjemná.



Obrázek 4.9: Zobrazení špatného převodu³

Při snaze o co nejlepší výsledek bylo důležitým faktorem i rozlišení videa. Při větších rozlišeních, kde byli snímky videa kvalitnější byl i výsledný snímek s efektem komiksové kresby lepší. Ostřejší hrany nebo plynulé hrany bylo důkazem kvalitního převodu videa.

³Obrázek 4.9 je upraven z převzatého videa z: <https://pixabay.com>

4.9 Testování

Testování je důležitým krokem při tvorbě jakékoliv aplikace. Tímto procesem lze získat zpětnou vazbu o intuitivnosti grafického rozhraní a správné funkcionalitě aplikace. Nabízí individuální hodnocení aplikace od lidí, kterým byla předložena. Hodnocení se odvíjí od mnoha faktorů, na které se uživatelé při testování zaměřují. Přesnost aplikace - tedy zda aplikace při různých vstupních videích dokáže poskytnout vhodně upravené výstupní videa s efektem komiksové kresby. Dalším aspektem je, zda je uživatel schopen bez obtíží aplikaci používat a chápe její chování. K otestování této aplikace je potřeba navrhnout postup, kterým se ověří tyto aspekty aplikace. Pomocí testování aplikace se pak dá vytvořit seznam věcí, které jsou potřeba pro výslednou podobu aplikace upravit.

Návrh

Pro dosažení co nejpresnějších výsledků je dobré se při testování zaměřit na větší počet lidí. Od testujících uživatelů je potřeba získat co největší počet informací, proto je pro testování navržen testovací plán. Pro jednotnou zpětnou vazbu je ideální uživatelům poskytnout alespoň jedno stejné vstupní video na kterém by zhodnotili výsledky aplikace.

Navržený testovací plán zahrnuje všechny činnosti, který by uživatel měl s aplikací vyzkoušet.

1. Spustit aplikaci
2. Zvolit vstupní a výstupní video a odzkoušet přepsání existujícího videa novým, nebo vytvořit zcela nové video.
3. Zvolit hodnoty úprav. Úroveň rozostření, citlivost detekce hran a počet barev pro kvantování.
4. Spustit proces úprav.
5. Přehrát vstupní a výstupní video. Zhodnotit subjektivní úspěšnost převodu videa.
6. Nahrát nové vstupní video, spustit úpravy a přehrát nové výstupy.

Tento postup by měl být pro všechny uživatele stejný. Možností testování je nechat uživatele odzkoušet aplikaci bez testovacího plánu a zhodnotit intuitivnost grafického rozhraní aplikace. Následně poskytnout uživateli testovací plán a porovnat postup při jednotlivých testováních.

Zjištěné informace

Výstupem testování jsou informace, které pomohou aplikaci vylepšit, popřípadě opravit nedostatky funkcionality nebo grafického rozhraní. Zároveň však zpětná vazba testujících uživatelů poskytuje indikátor o úplnosti aplikace.

Pozitivním hodnocením, které aplikace obdržela byla intuitivnost přístupu ke všem jejím funkcím. Grafické rozhraní bylo navrženo co nejjednodušeji, aby uživatel bez problému viděl, co mu aplikace nabízí. Tento aspekt aplikace během testování splnila.

Nejdůležitějším faktorem pro testování byla správná funkcionalita procesu úprav videa. To znamená, zda aplikace poskytla vhodný převod vstupního videa a video s efektem komiksové kresby. Uživatelé zhodnotili možnost individuálního výběru vstupních hodnot pro úpravu videa, počtu barev pro kvantování, úrovně rozostření a citlivosti detekce hran,

jako pozitivní funkci aplikace. Avšak problémem převodu se stala situace, kdy uživatel zadal delší video, kterého části nebyli obsahově podobné. Tento stav touto aplikací není řešitelný.

Prvním hlavním nedostatkem, na který upozornili všichni uživatelé byl časový interval běhu procesu úprav videa. Tento proces zabíral při úpravě delšího videa velké množství času. Proto byla navržena a následně implementována optimalizace na využívání uživatelem zadaného množství jader procesoru. Tato optimalizace je dál popsána níže.

Návrhem pro zlepšení přístupu uživatele k úpravě vstupních hodnot, bylo vytvoření zobrazovací plochy pro jeden snímek videa, na kterém by se při změně hodnot zobrazila adekvátní úprava. Uživatel si tak může prohlédnout jak změna vstupních parametrů ovlivní celkovou úpravu videa bez toho, aby musel čekat na ukončení procesu úpravy.

4.10 Optimalizace

Optimalizace je proces který vede k vylepšení výpočetního procesu aplikace a snížení nároku celé aplikace. Testující uživatelé často vznášeli stížnost, že zpracování videa trvá příliš dlouhou dobu. Řešením tohoto problému bylo umožnit uživateli zadat počet jader procesoru, které by se podílely na výpočetním procese paralelně a tak urychlily úpravu vstupního videa.

Nejprve je video načteno a zjištěna velikost snímku. Podle toho je poté určen počet snímků na jeden cyklus. To znamená, že je předběžně načteno do dynamického pole tolik snímků, aby se ještě vešly do paměti. Části načteného pole jsou poté rozděleny rovnoměrně každému vláknu ke zpracování.

V následující tabulce 4.1 je vidět přehled rozdílu rychlosti převodu různých videí v závislosti od uživatelem zadaného počtu jader, které mají video zpracovávat. Mimo časové délky videa je důležitou informací i rozlišení. Všechny testy proběhly na procesoru Intel i7-4500U.

Délka videa/min	Rozlišení videa	Počet jader / čas			
		1	2	3	4
3:00	640x360	00:01:46	00:01:22	00:01:11	00:01:00
0:29	1280x720	00:01:17	00:00:59	00:00:57	00:00:53
7:12	1280x720	00:18:06	00:14:58	00:12:51	00:11:55
0:22	1920x1080	00:02:39	00:01:49	00:01:48	00:01:31

Tabulka 4.1: Porovnání rychlostí převodu videa po optimalizaci 1

Pro větší přehlednost jsou v následující tabulce 4.2 zaznamenány převody na dvou videích. Každé z videí mělo dvě varianty na základě rozlišení. První testované video bylo dlouhé 54 sekund a druhé 30 sekund. Tyto údaje je nutné zmínit pro lepší přehled o naměřených hodnotách.

Video	Rozlišení videa	Počet jader / čas			
		1	2	3	4
Video1	960x540	00:01:35	00:01:09	00:00:59	00:00:48
Video1	1920x1080	00:06:48	00:05:50	00:05:02	00:04:10
Video2	640x360	00:00:49	00:00:45	00:00:42	00:00:37
Video2	1920x1080	00:03:57	00:03:25	00:02:59	00:02:28

Tabulka 4.2: Porovnání rychlostí převodu videa po optimalizaci 2

Jak je patrné z předchozích zaznamenaných hodnot, tato optimalizace pomohla urychlit převod videa. Rozdíl v rychlosti převodu při použití dvou jader a třech jader v některých případech není až tak velký. Důvodem může být, že procesor není složen fyzicky z čtyř jader ale pouze ze dvou a každé z nich se dělí na dvě logická podjádra. Hodnoty získané testováním převodů nemusí být ideální. Záleží na tom, jestli byl procesor v době převodu vytěžován i jinou činností.

Kapitola 5

Závěr

Cílem této práce bylo popsat techniky použité při nerealistickém zobrazování a jeden z algoritmů zvolit a implementovat jej nad video daty. Zvolený algoritmus z citované práce byl implementován a výstupem je video s aplikovanými prvky nerealistické tvorby, tudíž cíl práce byl splněn.

Pro výslednou aplikaci byl zvolen jeden z méně komplikovaných stylů vizualizace a to hlavně z důvodů, že výstupy se zdály vizuálně poutavější. Výhody a nevýhody daného algoritmu aplikovaného nad videem poté byly zjištěny během implementace a popsány v příslušné kapitole včetně výstupů, kterých bylo pomocí tohoto algoritmu dosaženo.

V rámci práce byly shrnuty nejpoužívanější a nejznámější úpravy realistické tvorby na tvorbu nerealistickou a byly předvedeny existující práce, které existují na podobné bázi. Rovněž bylo nahlédnuto do úprav a zpracování obrazu a byly popsány elementární operace, které se nedílně pojí s úpravami obrazových dat. Implementace pak odhalila technickou stránku této problematiky a případná úskalí.

Výsledkem práce je plně funkční aplikace schopná převádět realistické video na video s efektem komiksové tvorby. Poskytnutím možnosti v aplikaci přizpůsobit parametry úprav na vstupním snímku je možné vytvořit co nejvhodnější převod pro konkrétní videa. Z důvodu, že proces úprav je aplikován na každý snímek samostatně, je optimalizace pro zrychlení převodu velkým bonusem pro výslednou aplikaci.

Navzdory plné funkčnosti aplikace bylo získáno několik poznatků, o které by práce mohla být v budoucnosti rozšířena. Jedním z nich je, že bez ručních zásahů nelze vždy dosáhnout ideálních výsledků. Každé video už obsahuje vykreslenou scénu a proto spousta matematických detailů není k dispozici. Pro největší podobnost s komiksovou kresbou by bylo proto potřeba mnoho dodatečných úprav výstupního videa. Při zadání delšího videa totiž může nastat situace, že různé části videa potřebují jiné parametry úprav podle zobrazené scény.

Pokračování práce na této aplikaci by mohla být možnost rozdělit video na uživatelem zadané úseky a na různé úseky pak aplikovat vhodné parametry úprav. Dalším možným rozšířením práce by byla kombinace vícero technik pro převod realistického do nerealistického videa nebo případně nahrazení kvantování obrazu jinou úpravou, které by se také podobala komiksové tvorbě - ovšem v jiné stylizaci.

Kapitola 6

Literatura

- [1] AKENINE-MOLLER, Tomas, E. HAINES a N. HOFFMAN: *Real-Time Rendering*. Natick (Massachusetts): A.K. Peters Ltd., třetí vydání, 2008, ISBN 978-1-56881-424-7.
- [2] ENNIS, Garth, D. WOHL, M. SILVESTRI a J. BENITEZ: *Heart of darkness*. Los Angeles: Top Cow Productions, 2001, ISBN 1582402051.
- [3] FINKELSTEIN, Adam: *Non-Photorealistic Rendering*. Princeton: Princeton University, 2005.
URL <http://gfx.cs.princeton.edu/proj/sg05lines/course7-4-npr.pdf>
- [4] GOOCH, Bruce a A. GOOCH: *Non-Photorealistic Rendering*. United States: A K Peters and CRC Press, 2001, ISBN 1568811330.
- [5] HERTZMANN, Aaron: *Four Painterly Rendering Styles*. New York, USA: New York University, 1998, [cit. 25.12.2015].
URL <https://mrl.nyu.edu/projects/npr/painterly/styles.html/>
- [6] HERTZMANN Aaron a K. PERLIN: *NYU Media Research Lab*. New York, USA: New York University, 2001, [cit. 18.1.2016].
URL <https://mrl.nyu.edu/projects/npr/painterly>
- [7] Intel: *OpenCV 3.1.0*. [online], 2015-12-18, [cit. 3.2.2015].
URL <http://docs.opencv.org/3.1.0/index.html>
- [8] JEPSON, Michael: *Overlay transparent image in OpenCV*. Mountain View (California): Google Inc., 2012-10-26, [cit. 2.4.2016].
URL <http://jepsonsblog.blogspot.cz/2012/10/overlay-transparent-image-in-opencv.html>
- [9] KANG, Henry, S. LEE a C. K. CHUI: Flow-Based Image Abstraction. *IEEE Transactions on Visualization and Computer Graphics*, ročník 15, č. 1, 2009: s. 62–77, ISSN 1077-2626.
- [10] LAGANIERE, Robert: *OpenCV 2 Computer Vision Application Programming Cookbook*. Birmingham, Mumbai: Packt Publishing, 2011, ISBN 978-1849513241.

- [11] MALLICK, Satya: *Why does OpenCV use BGR color format ?* Platte Woods (USA): Big Vision LLC, 2015-09-27, [cit. 2.5.2016].
URL <http://www.learnopencv.com/why-does-opencv-use-bgr-color-format/>
- [12] Maymo: *Best Dog Birthday Surprise: DIY Ball Pit for Maymo*. Mountain View (California): Google Inc., 2015-06-28, [cit. 26.3.2016].
URL <https://www.youtube.com/watch?v=1kgmfPClsZc>
- [13] MCKESSON, Jason L.: *Learning Modern 3D Graphics Programming*. Linköping: Linköping University.
URL <http://webstaff.itn.liu.se/~stegu/TNM046-2015/LearningModern3DGraphicsProgramming.pdf>
- [14] NNEJI, Stephen: *The Coding Exodus*. Mountain View (California): Google Inc., 2012-12-10, [Online; navštíveno 2.4.2016].
URL <http://codingexodus.blogspot.cz/2012/12/working-with-video-using-opencv-and-qt.html>
- [15] RICHARDT, Christian: *Non-Photorealistic Rendering (NPR)*. Cambridge: University of Cambridge, 2010-02-18, [cit. 28.11.2015].
URL <http://www.cl.cam.ac.uk/teaching/1011/RSL/>
- [16] STROTHOTTE, Tomas a S. SCHLECHTWEG: *Non Photorealistic Computer Graphics*. San Francisco: Morgan Kaufmann, 2002, ISBN 978-1558607873.
- [17] The Qt Company: *Qt Documentation*. Helsinki: Digia, 2016-04-16, [cit. 1.5.2016].
URL <http://doc.qt.io/>
- [18] Wikimedia Foundation Inc.: *Resource Acquisition Is Initialization*. San Francisco: Wikimedia Foundation Inc., 2016-04-15, [cit. 1.5.2016].
URL https://en.wikipedia.org/wiki/Resource_Acquisition_Is_Initialization
- [19] ŽÁRA, Jiří, B. BENEŠ, J. SOCHOR a P. FELKEL: *Moderní počítačová grafika*. Brno: Computer Press, druhé vydání, 2005, ISBN 80-251-0454-0.

Příloha A

Obsah CD

Příložené CD má následující obsah:

- app - složka se spustitelnou aplikací
- projekt_src - složka se zdrojovými kódy k technické zprávě
- src - složka se zdrojovými kódy aplikace
- videos - složka s originálními videi a ukázkami videí vytvořených aplikací
- projekt.pdf - Technická zpráva
- readme.txt - Textový soubor obsahující postup pro instalaci