

# 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

## ČASOSBĚRNÉ VIDEO

SEMESTRÁLNÍ PROJEKT  
TERM PROJECT

AUTOR PRÁCE  
AUTHOR

Bc. ROMAN JIRKA

BRNO 2011



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

## ČASOSBĚRNÉ VIDEO

TIME-LAPSE VIDEO

### SEMESTRÁLNÍ PROJEKT

TERM PROJECT

### AUTOR PRÁCE

AUTHOR

Bc. ROMAN JIRKA

### VEDOUCÍ PRÁCE

SUPERVISOR

Doc. Ing. ADAM HEROUT, Ph.D.

BRNO 2011

## Abstrakt

Tato diplomová práce se zabývá úvodem do problematiky vytváření časosběrného videa. Specializuje se na případy, při kterých není použit stativ a je tedy nutné eliminovat vzniklé nedostatky. Základními nedostatky jsou rozdílná poloha jednotlivých snímků, rozdílný jas a barevné zabarvení. Dále je popsáno, jaké zásady by se měly dodržovat při vytváření časosběrného videa. Práce uvádí a implementuje metody pro odstranění základních nedostatků při tvorbě dlouhodobých časosběrných videí snímaných z ruky. Podrobně se věnuje zarovnání snímků, korekcí jasů a barev a také se zabývá porovnáním histogramů. Výsledkem práce je aplikace, která odstraňuje výše zmíněné nedostatky.

## Abstract

This thesis deals with the introduction into the topic of time-lapse video creation. It focuses on cases where tripod is not used and therefore it is necessary to eliminate incurred shortcomings. The main shortcomings are different position of individual frames, different brightness and color adjustment. The next topic describes which principles should be followed during the creation process. Thesis describes and implements methods for elimination of main shortcomings during process long time-lapse videos, which are recorded by hand. Thesis also precisely describes image registration, correction of brightness and colors. Thesis is also considers histograms comparison. Result of this work is application, which eliminates problems described above.

## Klíčová slova

častosběrné video, zarovnání obrazu, intervalometr, sekvenční snímání, gama korekce, způsobení histogramu, porovnání histogramů

## Keywords

time-lapse video, image registration, intervalometer, sequential recording, gamma correction, histogram adjustment, histograms comparison

## Citace

Roman Jirka: Častosběrné video, semestrální projekt, Brno, FIT VUT v Brně, 2011

# Časosběrné video

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Doc. Ing. Adama Herouta, Ph.D. Uvedl jsem všechny literální prameny a publikace, ze kterých jsem čerpal.

.....

Roman Jirka  
23. května 2011

## Poděkování

Děkuji vedoucímu diplomové práce Doc. Ing. Adamu Heroutovi, Ph.D. za podmětné rady, pedagogickou a odbornou pomoc při zpracování této práce.

© Roman Jirka, 2011.

*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

<b>1 Úvod</b>	<b>2</b>
<b>2 Základní pojmy, dostupné prostředky</b>	<b>3</b>
2.1 Vznik časosběrného videa . . . . .	3
2.2 Použití stativu kontra snímání z ruky . . . . .	4
2.3 Jak snímat časosběr . . . . .	4
2.4 Dostupný hardware . . . . .	5
2.5 Dostupný software . . . . .	7
<b>3 Zarovnání snímků</b>	<b>8</b>
3.1 Faktory zhoršující zarovnání snímků . . . . .	8
3.2 Detekce významných bodů . . . . .	8
3.3 Korespondence významných bodů . . . . .	10
3.4 Vyhledání vzoru . . . . .	11
3.5 Nalezení transformace mezi snímky pomocí algoritmu RANSAC . . . . .	12
3.6 Zarovnání snímků . . . . .	15
<b>4 Korekce jasu a barev</b>	<b>17</b>
4.1 Přičtení rozdílu průměrných hodnot obrázků . . . . .	18
4.2 Aplikace zalomené lineární křivky . . . . .	19
4.3 Přizpůsobení histogramu . . . . .	20
4.4 Gama korekce . . . . .	22
4.5 Metody porovnání histogramů . . . . .	25
4.6 Korekce pomocí barevného prostoru HSV . . . . .	31
4.7 Problémy a podpůrné prostředky pro zlepšení korekce . . . . .	33
<b>5 Implementace</b>	<b>37</b>
<b>6 Vyhodnocení výsledků a návrhy na budoucí rozšíření</b>	<b>39</b>
<b>7 Závěr</b>	<b>41</b>
<b>A Manuál k programu</b>	<b>43</b>
<b>B Obsah DVD</b>	<b>47</b>

# Kapitola 1

## Úvod

V dnešní moderní době je fotografování a natáčení videosekvencí běžná záležitost. Je to způsob, jak zaznamenat událost nebo sled událostí tak, jak je vnímáme. Lidské vnímání má svá omezení a člověk je od přírody tvor zvědavý, proto se vždy snažil zdokumentovat věci okem neviděné. Nejsme například schopni bez pomocné techniky pozorovat jevy, které probíhají v jiné části spektra než je viditelné světlo, jevy příliš rychlé nebo naopak pomalé. Nyní můžete namítnout, že pozorovat pomalé jevy schopni jsme. Ano, jsme, ale jen do té doby, dokud nechceme vidět celý proces kontinuálně. Mluvím nyní o procesech v řádu dnů, týdnů nebo i delších. Jsme totiž schopni zaznamenat změnu na pozorovaném objektu, ale už nejsme schopni danou změnu popsat detailně. Vidíme rostlinku za den povyrůst o 3 cm, ale už nedokážeme postihnout naklánění rostlinky ze strany na stranu v průběhu růstu. Záznamem pomalých jevů se zabývá časosběrné video, kterému se budu v této diplomové práci věnovat.

Výhoda u časosběrného videa je, že nepotřebujete žádnou speciální techniku. Postačí váš fotoaparát nebo třeba webová kamera. Časosběrné video můžete často najít v dokumentárních filmech, reklamě a různých naučných materiálech. Zobrazuje nám věci a děje, které probíhají velmi pomalu a stávají se těžko pozorovatelnými.

Cílem této diplomové práce je mimo jiné seznámit s problematikou vytváření časosběrných videí a popsat existující nástroje pro jejich vytváření. Úskalí vytváření časosběrných videí jsou probírána u témat, kterých se tato úskalí týkají. Dále si tato práce klade za cíl popsat a vytvořit nástroje pro hlavní nedostatky, které provázejí tvorbu videí.

Práce je zaměřena na vytváření dlouhodobých časosběrných videí snímaných z ruky. Podrobně se věnuje problematice zarovnání snímků a korekcemi jasů a barev. Uplatnění se najde u dlouhotrvajících jevů, kde si nemůžeme dovolit použít stativ. Další využití může najít při záznamu hojení zranění, kde se pohybuje snímaný objekt.

Práce navazuje na semestrální projekt, jehož hlavním cílem bylo poukázat na problematiku vytváření časosběrných videí, vyhledání a popsání existujících nástrojů a probrat úskalí jejich vytváření časosběrných videí. Ze semestrálního projektu byly pro účely této práce převzaty a dále dopracovány kapitoly 2 a 3.

## Kapitola 2

# Základní pojmy, dostupné prostředky

Tato kapitola se věnuje základním pojmům, problémům a dostupné technice pro vytváření časosběrného videa. Dozvíte se, jak takové video vzniká, co jsou nejčastější problémy při jeho vytváření a jaké jsou existující nástroje.

### 2.1 Vznik časosběrného videa

Jedná se o speciální video, které zachycuje pomalý jev a zobrazuje nám jej v pro nás pozorovatelném čase. Může vznikat z existujícího videozáznamu nebo ze série snímků. Základní princip je snímat záznam v pomalejším čase, než ve kterém bude přehráán. V praxi to znamená mít sérii snímků sejmuté s konstantní prodlevou mezi jednotlivými snímky, které se spojí do videosekvence. Jednoduše řečeno nasnímáme danou scénu vícekrát - vždy v jiném čase, čímž posbíráme jednotlivé kousky videa. Výsledným efektem je zrychlený čas. Časosběrné video může být uvažováno jako protiklad zpomalených záběrů.

Při vytváření pomocí již existujícího videozáznamu jde především o urychlení původního záznamu. Nejčastěji se provádí výběrem každého  $k$ -tého snímku v závislosti na požadovaném zpomalení. Je velmi neekonomické a někdy i nemožné snímat objekt, u kterého chceme znát průběh změny v rozmezí jednoho roku. Na druhou stranu pro efekt zrychleného filmu je videozáznam velice vhodný. Videozáznam má pevně danou prodlevu mezi jednotlivými snímky a tato plynulost se projeví i ve výsledném záznamu. Vyhneme se tím některým problémům při zpracování. Nevýhodou je malý efekt zrychlení záznamu (kvůli zdrojům i ekonomické stránce).

Oproti tomu série snímků by měla být snímána s konstantní prodlevou mezi jednotlivými snímky, jinak se může projevit jako značná trhanost ve výsledném záznamu. Nejčastěji se snímání provádí pomocí fotoaparátu, který podporuje funkci sekvenčního snímání, intervalového snímání nebo má možnost připojení intervalometru. Snímání pomocí klasického digitálního fotoaparátu, který nepodporuje výše uvedené funkce, je náročné, protože uživatel musí odměřovat časovou prodlevu sám a sám také mačkat spoušť.

Tématiku, která se nevztahuje k této práci, můžete najít na [10]. Jedná se především o historii časosběrného videa, krátká expozice kontra dlouhá expozice, pohyb kamery a HDR časosběr.

## 2.2 Použití stativu kontra snímání z ruky

Většina videí je snímána ze stativu, s pevně nastavenými parametry, aby je automatika nemohla ovlivňovat a výsledný snímek neměnil barevnost nebo světelnost. Takové snímky postačí spojit do videa a máme hotovo.

Vezměme ale nyní v potaz situaci, kdy chcete vidět průběh rekonstrukce zahrady. Kvůli povětrnostním podmínkám a nebezpečí krádeže nechcete venku zanechat svůj fotoaparát upevněný ve stativu. Raději jednou nebo vícekrát denně vyjdete na zahradu a vyfotíte objekt svého zájmu z ruky. Velkou výhodou je, že se vašemu fotoaparátu nic nestane (máte ho v ruce vy), můžete tímto způsobem zaznamenat jedním aparátem více objektů a můžete fotoaparát využívat i na jiné účely (nemusí zůstat ve stativu na zahradě). Na první pohled se tento způsob zdá ideální.

Co se ovšem stane se zachycenými snímky? Každý snímek bude pořízen z jiného místa. Nikdy nejste schopni zaznamenat dva snímky s jednodenní prodlevou z dokonale totožného místa. Snímky tedy budou navzájem posunuté, jinak zvětšené a pootočené. Pokud uděláte krok do strany a budete blízko snímanému objektu, bude narušena také perspektiva. Dále intenzita světla bude v okamžik snímání rozdílná. Také barvy se v průběhu dne a vlivem povětrnostních podmínek mění. Moderní fotoaparáty a jejich elektronika dokáže jevy jako jsou změna intenzity osvětlení a různá barevnost scény znatelně eliminovat, ale elektronika není neomylná a byť malá změna působí ve výsledném videu rušivě. Video prostě více či méně "bliká" intenzitou anebo barevností.

Tyto nedostatky je zapotřebí odstranit. Nejvíce rušivý element je posuv objektu našeho zájmu vlivem snímání z ruky. Ve své práci se tedy jako první zaměřím na odstranění právě tohoto problému.

## 2.3 Jak snímat časosběr

Častými náměty, které můžete najít na internetu jsou:

- rozvíjející se květ, rostoucí květina,
- zrychlený pohyb mraků nad krajinou, východ a západ slunce,
- rozklad ovoce, průběh koroze, tající led apod.,
- průběh výstavby nebo rekonstrukce budov,
- ruch velkoměsta.

Pro většinu záběrů je velice vhodné použít stativ spolu s automatickým snímáním a vytvořit stabilní prostředí, kde nedochází ke změně osvětlení. Tím je zajištěna stabilní poloha snímaného objektu a fotoaparátu<sup>1</sup>. Pokud to fotoaparát dovolí, je vhodné kvalitně zaostřit a vypnout automatické zaostřování, prodlouží se tím životnost baterie a zamezí se případnému náhodnému přeostržení. Preference manuálního nastavení i pro ostatní volby je taktéž vhodná. Jaké zvolit osvětlení a jak volit kompozici jsou otázky, kterými se tato práce nezabývá. Potřebné odpovědi můžete nalézt v literatuře věnující se kreativní fotografii.

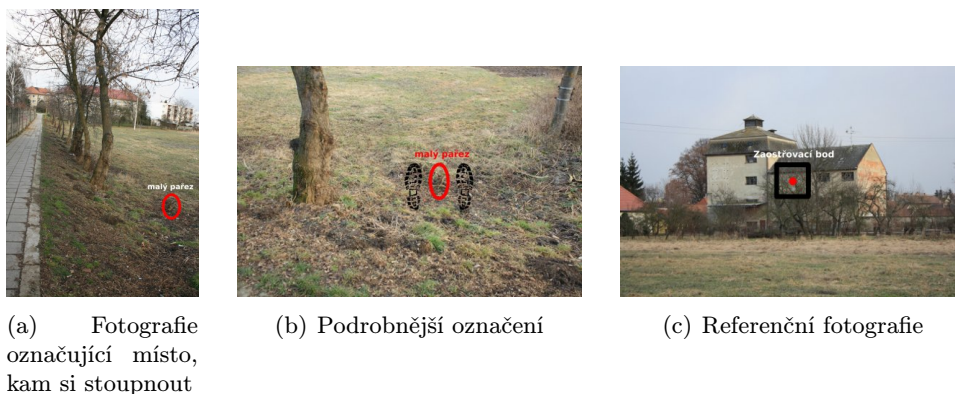
Před samotným vytvořením časosběrného videa je dobré obeznámit se s danou problematikou a možnostmi, které lze uplatnit. Vždy totiž platí, že dostatečná příprava před

<sup>1</sup>Mám na mysli prostředí, kde se dá zabránit neúmyslnému doteku fotoaparátu nebo snímaného objektu ať ze strany naší, rodinných příslušníků nebo domácích mazlíčků.



snímáním vede k urychlení zpracování a také k lepší kvalitě výsledku. Když potřebná informace není zaznamenána nebo je zaznamenána velmi špatně (špatné osvětlení, rozostření obrazu, apod.), tak ji často nejsme schopni zrekonstruovat. Příprava je rychlejší, přesnější a správná cesta. Lidé jsou ale často leniví provést potřebnou přípravu nebo nejsou s problémem dosti obeznámeni. Často také býváme omezeni samotnou technikou, která nemusí podporovat všechny potřebné funkce (absence bracketingu, nemožnost zvolit pevně dané expoziční časy, atd.) nebo je vyhodnotí špatně.

Pojďme se nyní věnovat naší problematice snímání z ruky. Před začátkem fotografování je dobré si projít okolí objektu našeho zájmu, popřemýšlet nad kompozicí a najít místo, které se dobře pamatuje, a budeme z něj moci snímat po celou dobu sběru snímků. Doporučuji vytvořit fotografii, na které je vyznačeno místo, kam si příště stoupnout a druhou fotografii s prvním pořízeným snímkem, aby bylo možné co nejlépe napodobit prvně pořízenou fotografii. Vytvoří se tak dobrý základ pro pozdější zpracování. Než se mi místo focení a umístění snímaného objektu vrylo do paměti, nosil jsem s sebou fotografie znázorněné na obrázcích 2.1.



Obrázek 2.1: Pomocné fotografie pro zajištění snímků, které se budou dát dobře stabilizovat

Dalším důležitým faktorem je zanést do záběru významné okolí, které nepodléhá povětrnostním podmínkám a předpokládá se jeho stálost a neměnnost. Nevhodné jsou stromy a všechny druhy rostlin. Naopak budovy, skály a podobné objekty jsou dobrým vodítkem. Na tyto objekty může být uplatněna maska a zarovnání bude pracovat efektivněji. Pokud nejsme schopni do záběru takové prvky zařadit, můžeme je vytvořit. Dobrým řešením je umístění skupiny kamenů nebo cedulí ve stejné rovině od objektivu jako je snímáný objekt. Při umístění příliš blízko fotoaparátu nebo příliš daleko se vystavujeme riziku nesprávného zarovnání snímaného objektu vlivem rozdílného posuvu popředí a pozadí vzhledem k sledovanému objektu. Proto i budovy a jiné stálé prvky by se měly nacházet pokud možno ve stejné rovině. Pokud je to možné, umísťujeme pomocné objekty tak, aby na výsledné fotografii zaujímaly pozici v rozích nebo na okraji. Můžeme je později jednoduše ořezat. Čím je delší vzdálenost mezi pomocnými body, tím bude zarovnání přesnější.

## 2.4 Dostupný hardware

Pro vznik časosběrného videa může být použita jakákoliv technika, která je schopná zachytit obraz. Poslouží nám mobilní telefon, digitální fotoaparát nebo třeba webová kamera. Intervalové snímání ale ve většině případech musíme provádět odpočtem a manuálním mačkáním

spouště.

Existují však také speciální fotoaparáty jako je např. GardenWatchCam od firmy Brinno, který je vyroben pro tyto účely. V podstatě je to připevněný senzor ke kolíku, který se zapíchne do země s možností intervalového snímání a ukládání snímků na paměťovou kartu. Jak takový fotoaparát vypadá můžete vidět na obrázku 2.2. Pro digitální zrcadlovky jsou často k dispozici intervalometry nebo-li spouště s časovačem. Jedná se o rozšířenou, převážně kabelovou spoušť s možností nastavení zpoždění prvního snímku, intervalu mezi snímky, počtem snímků a délkou expozice. Ukázka je na obrázku 2.3. Uvnitř digitální zrcadlovky je spousta mechanických částí, které se při snímání pohybují a jejich životnost není neomezená. Její použití je tedy na zvážení každého individuálního uživatele. Některé moderní kompaktní digitální fotoaparáty již podporují intervalové snímání přímo v menu přístroje.



Obrázek 2.2: Speciální fotoaparát GardenWatchCam od firmy Brinno (převzato ze stránek prodejce)



Obrázek 2.3: Intervalometr TC-80N3 pro některé typy zrcadlovky Canon EOS (převzato ze stránek prodejce)

## 2.5 Dostupný software

Aplikace pro vytváření časosběrného videa nejsou běžné jak jiné aplikace. Časté jsou programky pro jejich vytváření za pomoci intervalového snímání z webové kamery nebo fotoaparátu prostřednictvím počítače. Stačí chvíli hledat na internetu a naleznete jich celou řadu. Většinou se jedná o jednoduché programy. Řada výrobců dodává pomocné programy ke svým fotoaparátům. Při trošce pozornosti ale zjistíme, že existují jiné nástroje, které dokáží vytvářet časosběrná videa z již existujících snímků. Jsou to video editory. Video editorů je celá řada a většina dokáže pracovat se skupinou obrázků na vstupu. Například známé konzolové aplikace ffmpeg a mencoder dokáží ze sekvence snímků vytvořit video. Z grafických video editorů uvedme např. Avidemux, VirtualDub a QuickTime 7 Pro.

Méně častou funkcí video editorů je stabilizace videa. Jedná se o rozsáhlou a komplikovanou úlohu. Komerční software jsem nezkoušel, ale na poli nekomerčního použití má dobré výsledky plugin Deshaker do programu VirtualDub. S filtry pro odstranění jasového kolísání a ostrých barevných přechodů se setkáváme poměrně častěji. Při testech odstranění jasového kolísání v programu VirtualDub nebyly výsledky uspokojivé. Jednalo se o automatické metody pro korekci, které si nedokázaly moc dobře poradit s testovacím videem, kde první snímek byl správně exponován, druhý byl podexponován a třetí přexponován. Snímky se ve videu stále opakovaly a výsledné video nejevilo známky výrazného zlepšení.

Zajímavou myšlenku nabízí služba Picasa 3.8. Jmenuje se Face Movie a jedná se o sestavení série snímků, na kterých je obličej určité osoby, do prezentace, kde je obličej dané osoby vždy ve středu snímku. Není to typické časosběrné video, ale ve své podstatě nám ukazuje jednoho člověka během určitého časového období ve zrychleném čase.

## Kapitola 3

# Zarovnání snímků

Největším nedostatkem časosběru snímaného z ruky je rozdílná pozice fotografovaného objektu v pořízených fotografiích. Snímky se musí zarovnat ke kotvě (zvolená referenční fotografie určující pozici výsledku). Zarovnání snímků ke kotvě prochází sérií úkonů:

1. Vyhledání korespondujících bodů v obou obrazech pomocí významných bodů anebo vyhledáním vzoru,
2. nalezení afinní anebo perspektivní transformace mezi snímky,
3. zarovnání snímků.

Při vyhledání korespondujících bodů ve snímcích můžeme volit z prací s významnými body v obraze, nebo můžeme korespondenci nalézt pomocí vyhledání vzoru. Pro zjištění zarovnání snímků můžeme volit mezi afinní a perspektivní transformací. V dalších částech této kapitoly se na jednotlivé úkony podíváme podrobněji.

### 3.1 Faktory zhoršující zarovnání snímků

Mějme na paměti, že zarovnání snímků nebude nikdy ideální. Snímáme dynamické prostředí, které se časem mění. Mění se osvětlení, zabarvení a často i celé části scény. Navíc nás může zklamat použitá technika. Nesprávné zaostření fotoaparátu se projeví v nepřesné detekci pozice významných bodů a deskriptory budou méně unikátní. O případném rozmazání snímku pohybem ani nemluvě.

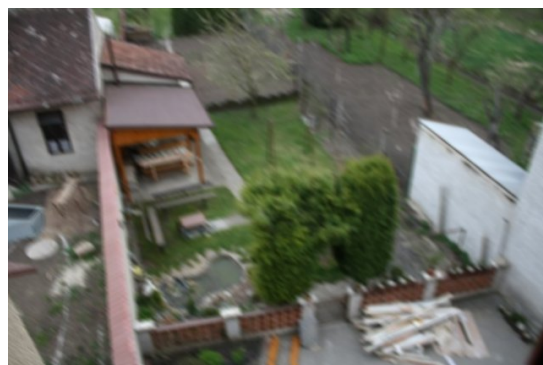
Dalším faktorem zkreslení výsledků je komprese formátů uchovávajících obrazovou informaci. Při vysoké úrovni komprese se tvoří výrazné artefakty. Objektiv fotoaparátu obsahuje vady (soudkování, chromatická aberace, atd.), které zanáší další chyby do procesu zarovnání snímků. Navíc se vady objektivu mění při zoomování. Obzvláště velkým problémem je soudkování objektivu. Přímký se deformují u okraje obrázku na křivky, které zanášejí velkou chybu do výpočtu transformace. Obrázky 3.1 zobrazují některé uvedené faktory.

### 3.2 Detekce významných bodů

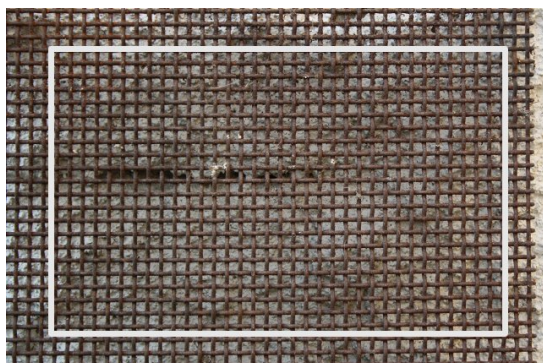
Pro detekci významných bodů jsem se rozhodl využít dostupnou metodu pro detekci významných bodů SURF (Speeded-Up Robust Features [4], [5]) z knihovny OpenCV [1], protože je rychlá a zároveň počítá deskriptory k nalezeným bodům. Při testování ostatních



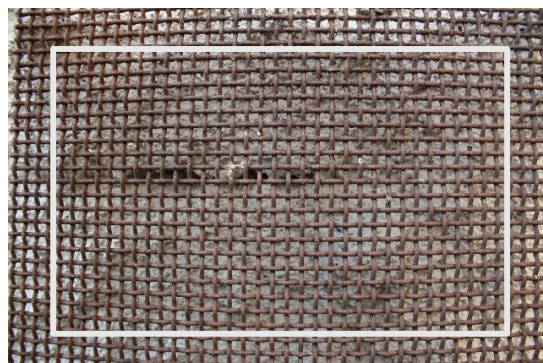
(a) Vysoká komprese JPEG



(b) Rozmazaný snímek



(c) Ohnisko 55 mm



(d) Ohnisko 18 mm - soudkování objevující se zejména u krátkých ohniskových vzdáleností

Obrázek 3.1: Ukázka problémů přispívajících k zhoršení zarovnání snímků

detektorů nebyly výsledky uspokojivé. Zkoušel jsme i moderní algoritmus MSER, ale narazil jsme na problém s extrakcí deskriptorů. Zjišťování vzájemné korespondence dle kontur nepřineslo dobrý výsledek. Zůstal jsem proto u metody SURF.

SURF je metoda pro detekci významných bodů v obraze a vytváření deskriptory pro dané body. Je částečně inspirován metodou SIFT, ale je zde kladen větší důraz na rychlost. K detekci významných bodů se používá Hessianův detektor za pomoci integrálního obrazu. Významnost nalezeného bodu může být odhadnuta z Hessianovy hodnoty. Deskriptory jsou vypočítány z odezvy 2D Haarových vlnek a jsou invariantní vůči rotaci a zvětšení. Pro podrobnější informace hledejte v [5].

Pro přesnější zpracování je možno použít masku, kterou se odfiltrují problémové oblasti jako jsou obloha či stromy. Mohou se tedy zpracovávat jen dobře zpracovatelné oblasti obsahující neopakující se různorodou strukturu. Ukázka nalezených významných bodů je vidět na obrázcích 3.2, kde na levé straně je kotevní snímek a na straně pravé snímek k zarovnání.

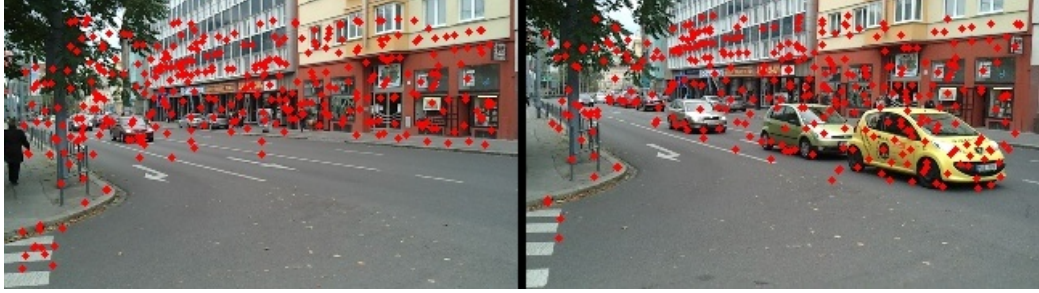
Množství detekovaných bodů lze ovlivnit nastavením prahu pro Hessianovu hodnotu přímo při volání funkce. Doporučená hodnota je mezi 300 až 500. V aplikaci je práh nastaven na hodnotu 500. Snížení prahu se mi neosvědčilo, protože nalezené body měly méně přesnou pozici a výsledné zarovnání bylo zatíženo větší chybou.

Detekce významných bodů selhává u nekvalitních obrázků a obrázků s malým rozlišením, kdy je detekováno příliš málo bodů nebo není pro body nalezena korespondence.





(a) Načtené obrázky z videosekvence



(b) Výsledek detekce významných bodů

Obrázek 3.2: Ukázka detekce významných bodů

Metodu doporučuji u snímků pořízených kvalitním fotoaparátem s vysokým rozlišením, kdy nepřesné umístění detekovaných bodů není díky velikosti snímku výrazné.

### 3.3 Korespondence významných bodů

Ke zjišťování vzájemné korespondence bodů v obraze se nejlépe osvědčilo prahování pomocí korelace [7] deskriptorů. Korelaci dvou deskriptorů vypočítáme podle vzorce 3.1, kde  $r_{xy}$  je výsledný korelační koeficient pro korelaci deskriptorů  $x$  a  $y$ ,  $N$  je velikost deskriptoru,  $\mu_x$  je střední hodnota deskriptoru  $x$  vypočítaná podle vzorce 3.2 a  $\sigma_x$  je druhá odmocnina rozptylu deskriptoru  $x$  vypočítaný vzorcem 3.3. Hodnoty  $\mu_y$  a  $\sigma_y$  jsou vypočítány obdobně.

$$r_{xy} = \frac{\sum_{i=0}^{N-1} (x(i) - \mu_x) \cdot (y(i) - \mu_y)}{(N-1) \cdot \sigma_x \cdot \sigma_y} \quad (3.1)$$

$$\mu_x = \frac{1}{N} \sum_{i=0}^{N-1} x(i) \quad (3.2)$$

$$\sigma_x = \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} (x(i) - \mu_x)^2} \quad (3.3)$$

Pomocí korelace se porovná každý deskriptor s každým a zaznamenají se dvojice s nejlepšími výsledky, které značí hledané korespondence. Aby se zamezilo velkému množství špatných korespondencí, jsou odstraněny všechny korespondence, které nesplňují práh určující míru podobnosti. Práh je zvolen na hodnotu 0,7. Ukázka nalezených korespondencí je vidět na obrázcích 3.3.



(a) Korespondence mezi obrázky



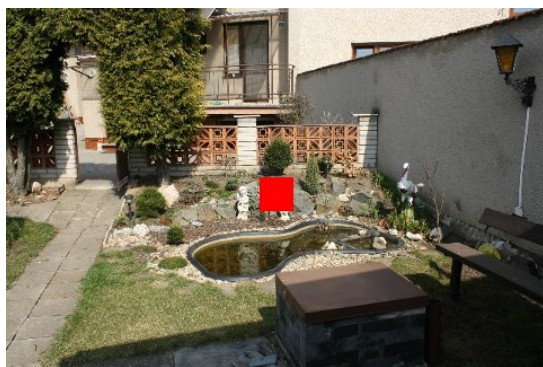
(b) Korespondence vyznačeny v upravovaném snímku

Obrázek 3.3: Úsečky znázorňující korespondence bodů. Na spodním obrázku jsou pro lepší přehlednost zobrazeny korespondence jen v upravovaném obrázku s posunem vzhledem k referenčnímu obrázku.

Na obrázku 3.8(b) jsou vidět chybně nalezené korespondence. Fotografovaná scéna je dynamická a v každém následujícím snímku se detekují body nové a jiné zase zmizí. Výsledné korespondence mohou tedy ukazovat na nově detekované body.

### 3.4 Vyhledání vzoru

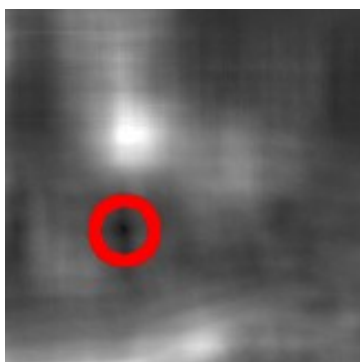
Kroky detekce významných bodů a vyhledání korespondence se mohou nahradit pomocí vyhledání vzoru. Pro vyhledání vzoru používám implementaci dostupnou v knihovně OpenCV. Funkce se jmenuje `cvMatchTemplate`. Pracuje se 3 obrázky. První obrázek je vzor (dále jen vzor), ve druhém obrázku se vyhledává daný vzor (dále jen obrázek) a třetí obrázek obsahuje výsledek vyhledávání (dále jen výsledek). Obrázek nesmí být menší než vzor. Výsledek má velikost rozdílu velikostí obrázku a vzoru  $+1$ . Funkce prochází postupně obrázek a porovnává jeho překrývající část se vzorem. Výsledek porovnání ukládá do výsledku. Jako porovnávací funkci jsem zvolil rozdíl čtverců [1]. Výslednou korespondenci určuje pozice minimální hodnoty ve výsledku. Pozice musí být přepočítána vzhledem ke vzoru i obrázku. Na obrázcích 3.4 je znázorněna funkčnost metody.



(a) Referenční snímek s vyznačeným vzorem



(b) Aktuální snímek s vyznačeným obrázkem



(c) Výsledek porovnání vyznačených oblastí s vyznačenou minimální hodnotou



(d) Nalezené korespondence

Obrázek 3.4: Ukázka funkce vyhledání vzoru

### 3.5 Nalezení transformace mezi snímky pomocí algoritmu RANSAC

RANSAC je zkratka pro RANdom SAMple Consensus [2] (shoda náhodných vzorků). Je to iterativní metoda pro odhad matematického modelu ze sady pozorovaných dat, které obsahují nevhodné data (tzv. outliers). Základním předpokladem je, že pozorovaná data obsahují vhodná data (inliers), jejichž rozložení může být popsáno nějakou sadou parametrů modelu. Nevhodná data jsou ta, která neodpovídají modelu.

Funkci algoritmu zobrazuje pseudokód v algoritmu 1. Pseudokód je převzat z [8].

Pro potřeby práce jsem implementoval RANSAC pro výpočet afinní transformace, tedy transformace skládající se z rotace, zvětšení, a posuvu na obou osách. Model pro afinní transformaci má jako parametry posuny na osách X a Y, rotaci a zvětšení. Elementy použité v algoritmu představují korespondující body v obrazech. Vytvořený algoritmus nenajde tak přesnou transformaci jako nalezení perspektivní transformace pomocí funkce `cvFindHomography` implementované v knihovně OpenCV. Zato se z něj dají získat přesnější afinní hodnoty a nevrací nesmyslné údaje. Při zjištění, že transformace nebyla nalezena, můžeme snímek vypustit z výsledného videa. Tím zajistíme kvalitnější výsledek neobsahující nezarovnané snímky.

Pro samotný výpočet afinních hodnot jsem použil vzorce známé z počítačové grafiky pro



**Data:**

data - sada pozorovaných dat  
 model - model, který může být přiřazen datům  
 n - číslo, určující minimální počet elementů pro výpočet modelu  
 k - počet iterací algoritmu  
 t - práh pro označení inlieru  
 d - minimální počet nalezených inlierů, aby mohl být model prohlášen za platný

**Result:**

best\_model - nalezený model, který nejlépe odpovídá pozorovacím datům (nebo NULL, pokud není nalezen vhodný model)  
 best\_consensus\_set - elementy, pro které byl model odhadnut  
 best\_error - chyba modelu vzhledem k datům

```

iterations := 0;
best_model := NULL;
best_consensus_set := NULL;
best_error := nekonečno;
while iterations < k do
  maybe_inliers := n náhodně zvolených elementů z data;
  maybe_model := parametry modelu vypočítané pomocí maybe_inliers;
  consensus_set := maybe_inliers;
  foreach element v pozorovacích datech, který není v maybe_inliers do
    if element odpovídá modelu maybe_model s chybou menší než t then
      | přidej element do consensus_set;
    end
  end
  end
  if počet elementů v consensus_set > d then
    (to znamená, že je nalezen vhodný model, teď se otestuje, jak je dobrý)
    better_model := parametry modelu vhodné pro všechny elementy
    v consensus_set;
    this_error := vypočti chybu modelu vzhledem k elementům
    v better_model;
    if this_error < best_error then
      (je nalezen lepší model než všechny předchozí modely, ulož ho do doby,
      než bude nalezen lepší model)
      best_model := better_model;
      best_consensus_set := consensus_set;
      best_error := this_error;
    end
  end
  end
  zvyš iterations;
end
vrať best_model, best_consensus_set, best_error;

```

**Algorithm 1:** Pseudokód pro algoritmus RANSAC

transformační matice [6]. Vzorec pro aplikaci transformační matice zobrazuje rovnice 3.4, kde  $\varphi$  je úhel,  $s$  značí zvětšení,  $S_x$  a  $S_y$  označují posuv na osách X a Y a  $C_x$ ,  $C_y$  označující

střed obrazu.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \alpha & \beta & \gamma \\ -\beta & \alpha & \delta \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.4)$$

Význam ostatních proměnných je uveden v rovnicích 3.5.

$$\begin{aligned} \alpha &= \cos(\varphi).s \\ \beta &= \sin(\varphi).s \\ \gamma &= (1 - \alpha)C_x - \beta C_y - S_x \\ \delta &= \beta C_x + (1 - \alpha)C_y - S_y \end{aligned} \quad (3.5)$$

Maticový výpočet je možné převést na soustavu 2 rovnic o 4 neznámých, zobrazenou v rovnici 3.6,

$$\begin{aligned} x' &= \alpha x + \beta y + \gamma \\ y' &= -\beta x + \alpha y + \delta \end{aligned} \quad (3.6)$$

které jsou ovšem nedostatečné a jsou zapotřebí 2 body pro zjištění potřebných parametrů.

Vytvořená matice, reprezentující soustavu lineárních rovnic (viz rovnice 3.7), je vyřešena zavoláním funkce `cvSolve` z knihovny `OpenCV`. Tím dostaneme parametry  $\alpha$ ,  $\beta$ ,  $\gamma$  a  $\delta$ . Proměnné  $x_1$ ,  $y_1$  značí pozici bodu na ose X a Y v originálním obraze a proměnné  $x'_1$ ,  $y'_1$  značí korespondující pozici bodu v aktuálně zarovnávaném obraze. Obdobně pro proměnné s indexem 2 pro druhý bod.

$$\begin{aligned} x_1\alpha + y_1\beta + 1\gamma + 0\delta &= x'_1 \\ y_1\alpha - x_1\beta + 0\gamma + 1\delta &= y'_1 \\ x_2\alpha + y_2\beta + 1\gamma + 0\delta &= x'_2 \\ y_2\alpha - x_2\beta + 0\gamma + 1\delta &= y'_2 \end{aligned} \quad (3.7)$$

Výpočet výsledných hodnot určujících transformaci je uveden v rovnici 3.8.

$$\begin{aligned} \varphi &= \arctg\left(\frac{\beta}{\alpha}\right) \\ s &= \frac{\beta}{\sin(\varphi)} \\ S_x &= -\gamma + (1 - \alpha)C_x - \beta C_y \\ S_y &= -\delta + \beta C_x + (1 - \alpha)C_y \end{aligned} \quad (3.8)$$

Algoritmus funguje na principu náhodně zvolených elementů z pozorovaných dat. Jeho výsledky jsou proto neopakovatelné. Při opětovném spuštění jsou náhodně vybrány jiné body a výsledek se bude lišit. Pro upřesnění výsledku jsem zkoušel vypočítat průměrnou hodnotu z náhodných inlierů tvořící model i shlukovat inliery. Výsledek se nezlepšil, zůstal stejný a občas se i zhoršil. O jiné zpřesnění výsledku jsem se nepokoušel. Chceme-li přesnější výsledek, můžeme zvýšit počet iterací, čímž se zvýší pravděpodobnost na nalezení nejlepšího možného modelu. Dále můžeme snížit práh pro označení inlieru, čímž získáme méně inlierů, ale jejich pozice bude více odpovídat modelu.

Velkou výhodou je, že algoritmus nalezne model, který odpovídá největšímu množství inlierů. Vyloučí se tím chybné korespondence a pohyb části obrazu (např. pohyb větvi stromů, apod.). Ukázkou výsledných inlierů můžete vidět na obrázku 3.5.



(a) Nalezené korespondence

(b) Korespondence určené inliery

Obrázek 3.5: Výsledné inliery po nalezení transformace algoritmem RANSAC. Výsledná transformace pro pravý obrázek: úhel  $2^\circ$ , zvětšení 1.0, posun 8,3 pixelů na ose X a posun -19,1 pixelů na ose Y

### 3.6 Zarovnání snímků

Výsledné zarovnání snímku je aplikace nalezené transformace. Při zarovnání snímku vzniká oblast, kde nejsou žádná dostupná data. Tato oblast se projeví černou barvou ve výsledném videu (viz obrázek 3.6. Pro odstranění černých okrajů u zarovnaných snímků má uživatel možnost nastavit permanentní zvětšení pro výstupní snímek, čímž dojde k ořezání vznikajících černých okrajů. Výsledek můžete vidět na obrázku 3.7



Obrázek 3.6: Na levé straně je referenční obrázek a na straně pravé aktuální obrázek po aplikaci nalezené transformace

Odstranění černých okrajů za pomoci permanentního zvětšení není příliš efektivní metoda. Zkresluje tím výsledné video. Časosběrné video má tu výhodu, že objekt našeho zájmu se nachází na stejném místě a jeho okolí je často stejné. Využitím této informace můžeme nově transformovaný snímek umístit na předchozí snímek a černé okraje zprůhlednit. Místo černých okrajů je nyní předchozí obrazová informace a nedochází k degradaci obrazu permanentním zvětšením. Oblast překryvu se vyznačuje ostrými přechody, což je rušivé. Provádím tedy pozvolné prolínání na okrajích nově transformovaného obrazu. Ukázky odstranění černých okrajů můžete vidět na obrázcích 3.8.

Při afinní nebo perspektivní transformaci vzniká aliasing. Knihovna OpenCV pro transformaci implementuje dokonalejší interpolační metody, u kterých není aliasing zřetelný. Jako interpolační metodu jsem zvolil bikubickou interpolaci.



Obrázek 3.7: Stejná situace jako u obrázku 3.6, ale po aplikaci permanentního zvětšení



(a) Ukázka ostrého okraje



(b) Korespondence určené inliery

Obrázek 3.8: Výsledné inliery po nalezení transformace algoritmem RANSAC. Výsledná transformace pro pravý obrázek: úhel  $2^\circ$ , zvětšení 1.0, posun 8,3 pixelů na ose X a posun -19,1 pixelů na ose Y

## Kapitola 4

# Korekce jasu a barev

Při zpracování dlouhodobého časosběrného videa se dostáváme do situací, kdy jednotlivé snímky mají rozdílné hodnoty jasu a barevného vyvážení. U videí zachycujících krátkodobý děj (v rozmezí maximálně několika dnů) se jednotlivé změny projevují plynule a ve výsledku nám připadají přirozené. Ovšem při snímání požadovaného objektu s dlouhými časovými rozestupy se poloha slunce na obloze výrazně změní a nově vržené stíny působí problém v odhadu expozičních časů. Také barevné zabarvení snímku během dne je rozdílné. Do mnohem komplikovanější situace se dostaneme, pokud začneme pořizovat časosběrné video na podzim a během zimy napadne sníh. Barevnost snímku se téměř vytratí a korekce se stává velmi problematickou. Dalším problémem je rozdílné počasí, konkrétně barevnost oblohy, která jeden den září modrou barvou a druhý den je to pouze šedá plechová plocha na obrázku. Těmto problematikám se budu věnovat v této kapitole.

Pro úpravu jasu, kontrastu a často i barev se ve foto-editorech využívá nástroj křivky, který pracuje s histogramy. Potřebujeme znát výskyt jednotlivých jasů v obraze a histogramy nám přesně tyto informace poskytují. Proto jsou to ideální nástroje. Při zpracování pracuji s histogramy, které reprezentují jednotlivé barevné složky. S barevnými složkami pracuji samostatně, čímž se vyřeší problém korekce jasu a zároveň i barevného zabarvení snímku. V dalším textu budu zmiňovat práci s obrazem, ale ve skutečnosti se daný postup uplatní samostatně pro každou barevnou složku.

Většina obrázků zobrazených v této kapitole je složena ze 3 snímků, kde střed obrazu je referenční obrázek. Přičemž na levé a pravé straně jsou zobrazeny upravované obrázky. Zvolil jsem tento způsob zobrazení pro lepší názornost jednotlivých metod. Pod obrázky jsou navíc vyobrazeny histogramy pro jednotlivé barevné složky. Histogramy vypočítané pouze ze zobrazených částí by byly zavádějící, výpočet histogramu se proto provádí z celého obrázku. Pokud není uvedeno jinak, z důvodu jednoduššího zápisu obrázky nabývají hodnot jasu od 0 do 1.

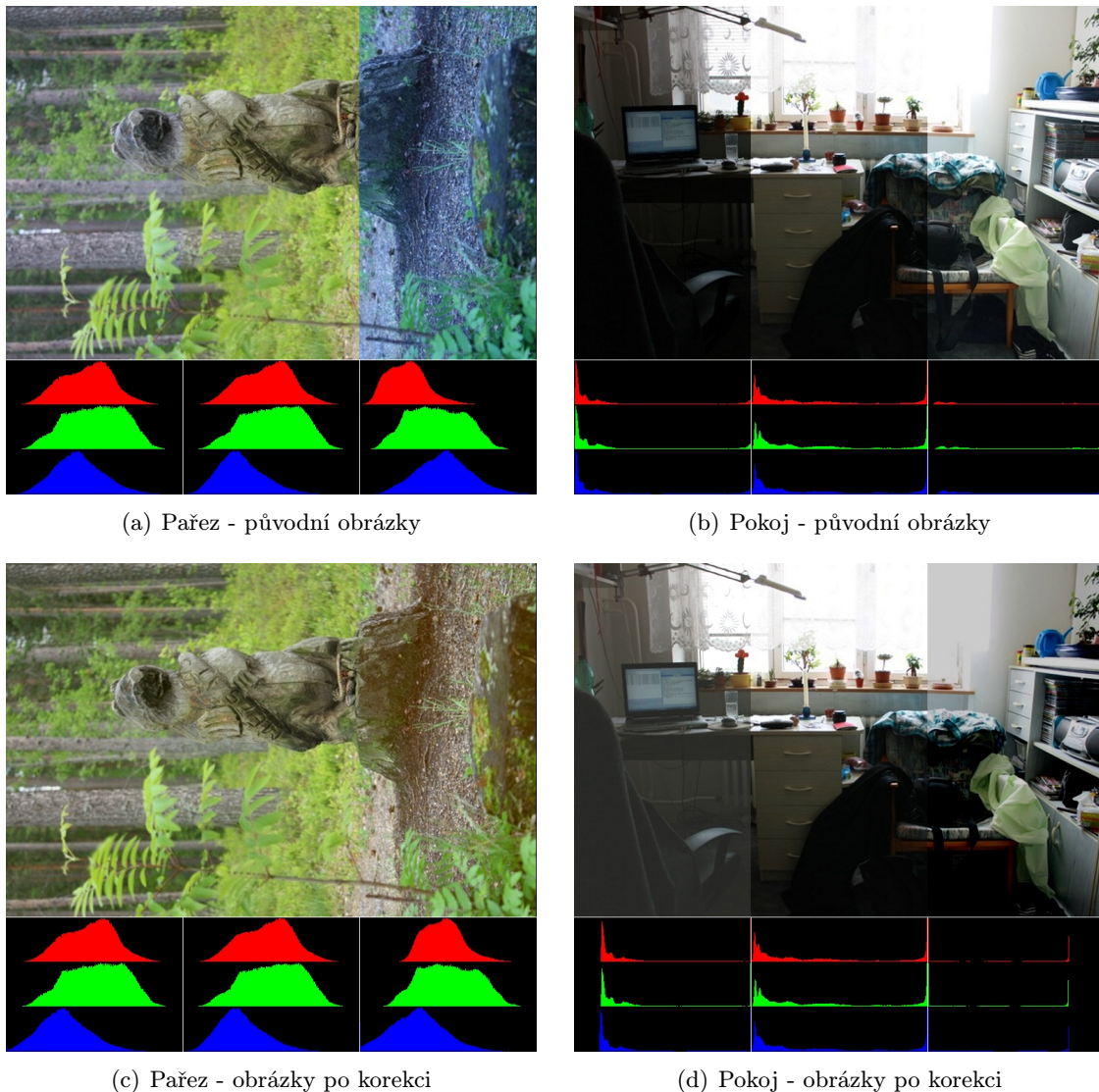
Moje první testovací snímky byly pořízeny v pokoji, který jsem porovnával se stejným snímekem, jen expoziční čas byl jednou nastaven na -2 EV a podruhé +2 EV. Další testovací snímky zobrazují obrázek vyřezaného pařezu, kde rozdíly byly ve vyvážení bíle. Můj požadavek byl, aby si korekce dokázala poradit s úpravou těchto snímků. V průběhu testování jsem narazil na situaci, kde referenční snímek byl pořízen v zimě beze sněhu a snímek určený pro korekci v zimě se sněhem. Dynamika histogramu jednotlivých snímků se výrazně liší a navíc podstatnou část snímku tvoří obloha, která v době nevlídného počasí tvoří velkou jednolitou plochu. Rozšířil jsem testování i o tento snímek, na kterém je vyobrazen starý mlýn.

Přemýšlel jsem také nad lokálními metodami, které by využívaly Haarovy transformace



nebo obrazových pyramid. Avšak lokální metody by byly náchylné na stabilizaci snímku a při malé odchylce stabilizace by mohly vznikat přeslechy. Stabilizace snímků není vždy ideální a proto jsem se rozhodl pro metody upravující obrázek globálně.

#### 4.1 Přičtení rozdílu průměrných hodnot obrázků



Obrázek 4.1: Korekce metodou přičtení rozdílu průměrné hodnoty jasu

První testovanou metodou bylo přičtení rozdílu průměrné hodnoty jasu aktuálního snímku vzhledem k referenčnímu snímku. Výpočet je uveden v rovnici 4.1, kde  $\mu_r$  je střední hodnota jasu referenčního obrázku,  $\mu_a$  je střední hodnota jasu aktuálně zpracovávaného obrázku,  $x(i)$  je hodnota jasu daného pixelu aktuálního obrázku před úpravou,  $y(i)$  je výsledná

hodnota pixelu  $x(i)$  po úpravě a  $v(i)$  je pomocná hodnota při výpočtu.

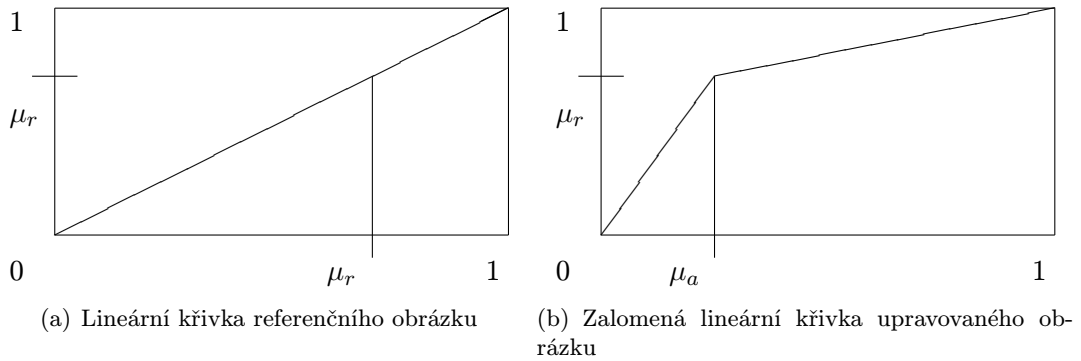
$$v(i) = x(i) + (\mu_r - \mu_a)$$

$$y(i) = \begin{cases} 0 & \text{v případě } v(i) < 0 \\ v(i) & \text{v případě } 0 \leq v(i) \leq 1 \\ 1 & \text{v případě } v(i) > 1 \end{cases} \quad (4.1)$$

Tento postup se rychle ukázal jako neúčinný, protože se posunou minimální a maximální hodnoty pixelů v obrázku a často přesáhnou rozsah obrázku. Hodnoty se musí prahovat a obrázek ztrácí kontrast a následkem toho v něm vznikají nevzhledné šedé plochy. Při pohledu na obrázek pařezu (obrázek 4.1(c)) se výsledek může zdát uspokojivý, ale pokud metodu použijeme na obrázky s rozdílnou dobou expozice (obrázek 4.1(d)), popsané problémy začnou být výrazné. Od této metody jsem tedy upustil.

## 4.2 Aplikace zalomené lineární křivky

Další metoda byla založena na předchozí myšlence, ale s tím rozdílem, že úprava nebyla provedena jednoduchým přičtením rozdílu středních hodnot, ale aplikací zalomené lineární křivky, viz 4.2.

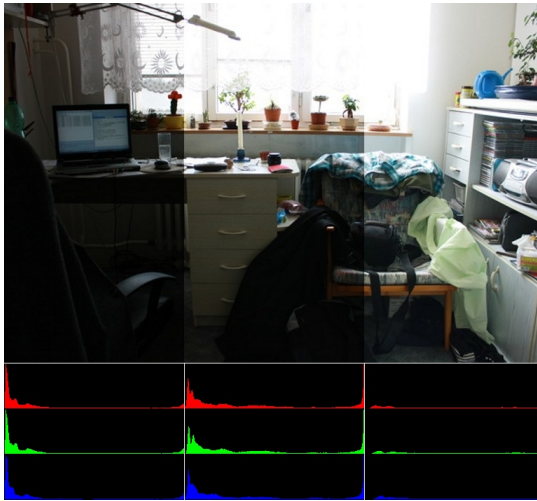


Obrázek 4.2: Vytvoření zalomené přímky křivky pro korekci obrázku. Levý obrázek znázorňuje lineární křivku s naměřenými hodnotami z referenčního snímku. Pravý obrázek je křivka použitá pro korekci aktuálního snímku. Hodnota  $\mu_r$  je střední hodnota jasu referenčního obrázku a hodnota  $\mu_a$  je střední hodnota jasu aktuálního obrázku. Lom křivky nastává v hodnotě  $\mu_a$  pro vstupní hodnotu jasu a  $\mu_r$  pro výstupní hodnotu jasu. Počátek a konec křivky je nezměněn.

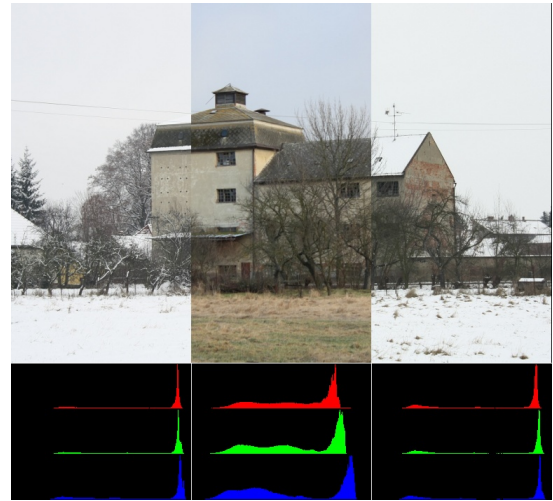
Výpočet křivky je uveden v rovnici 4.2, kde jednotlivé proměnné mají stejný význam jako v části **Přičtení rozdílu průměrných hodnot obrázků**.

$$y(i) = \begin{cases} \frac{x(i)\mu_r}{\mu_a} & \text{v případě } x(i) \leq \mu_a \\ \frac{\mu_a}{(x(i)-\mu_a)(1.0-\mu_r)} + \mu_r & \text{jinak} \end{cases} \quad (4.2)$$

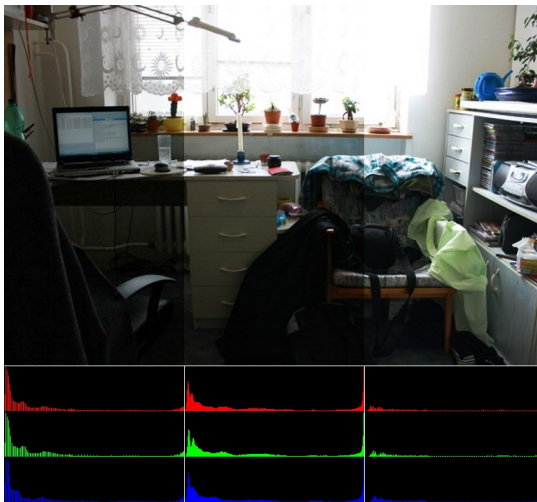
Tato metoda pracovala spolehlivěji, ale dosažené výsledky stále nebyly uspokojivé. Kontrast snímku zůstal zachován, ale výsledná korekce nebyla dostatečná. Výsledek můžete vidět na obrázku 4.3.



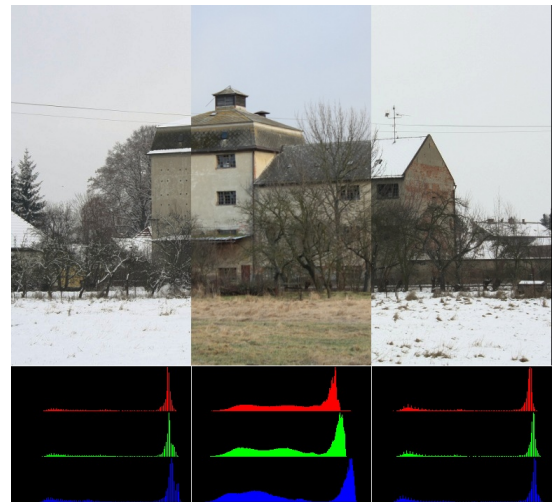
(a) Pokoj - původní obrázky



(b) Mlýn - původní obrázky



(c) Pokoj - obrázky po korekci



(d) Mlýn - obrázky po korekci

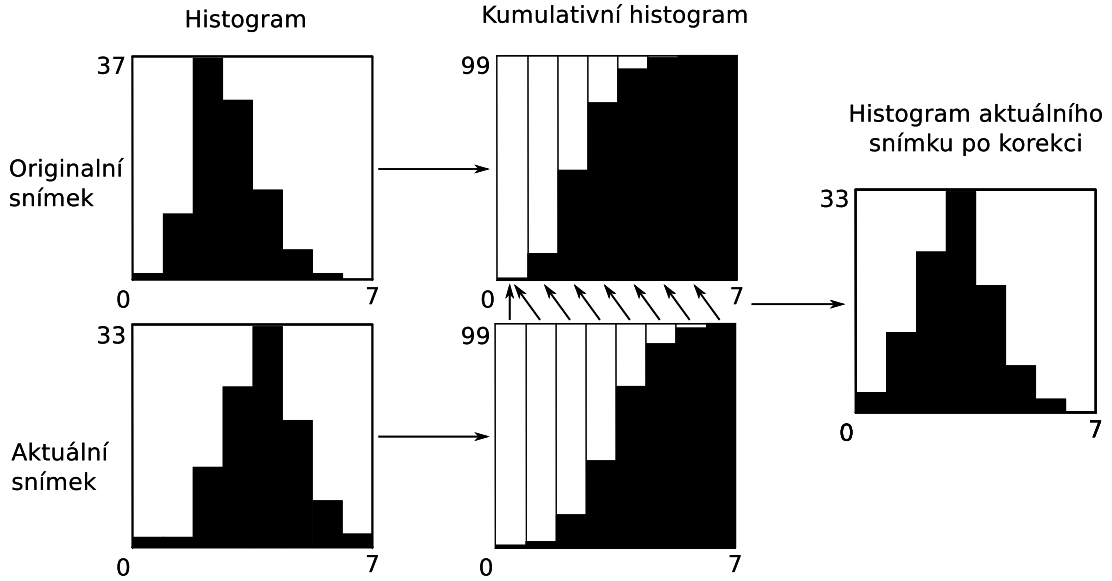
Obrázek 4.3: Korekce metodou aplikace zalomené lineární křivky

### 4.3 Přizpůsobení histogramu

Když jsem přemýšlel, jakou jinou metodu uplatnit pro korekci, napadlo mě, že výsledkem by měly být co možná nejpodobnější histogramy. Pro napodobení referenčního histogramu využívám kumulativní histogram. Metoda spočívá ve vytvoření vyhledávací tabulky (lookup table) na základě hledání nejbližších hodnot kumulativních histogramů. Jak metoda pracuje, můžete vidět na obrázku 4.4, který nabývá celočíselných hodnot jasu od 0 do 7. Příslušná vyhledávací tabulka je uvedena v tabulce 4.1. Nejdříve jsou vypočítány kumulativní histogramy, pomocí vzorce 4.3, kde  $C$  je kumulativní histogram a  $H$  je normální histogram.

$$C(i) = \sum_{j=0}^i H(j) \quad (4.3)$$





Obrázek 4.4: Ukázka korekce metodou přizpůsobení histogramu

Index	Hodnota
0	0
1	0
2	1
3	2
4	3
5	4
6	5
7	6

Tabulka 4.1: Vyhledávací tabulka pro obrázek 4.4

Pomocí kumulativních histogramů je vytvořena vyhledávací tabulka dle vzorce 4.4, kde  $L$  je vyhledávací tabulka,  $C_a$  je kumulativní histogram aktuálního snímku,  $C_r$  je kumulativní histogram referenčního snímku a  $n$  je šířka histogramu (v našem případě 8).

$$L(i) = \underset{j=\langle 0, n-1 \rangle}{\operatorname{argmin}} (|C_a(i) - C_r(j)|) \quad (4.4)$$

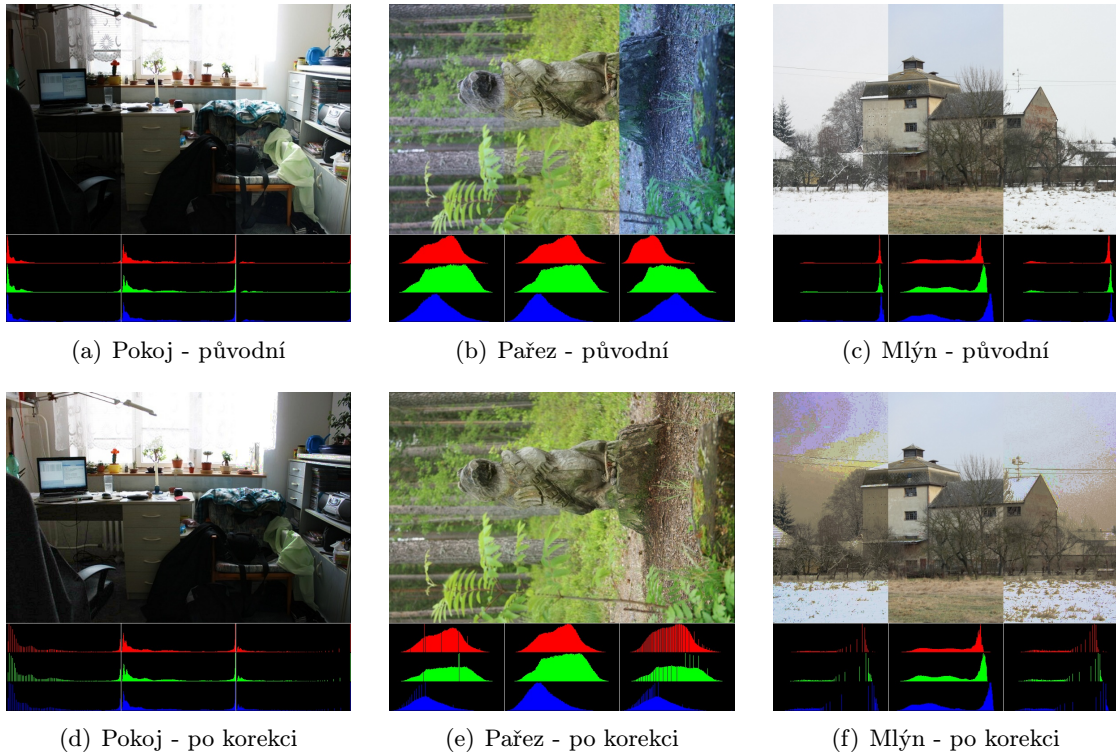
Korekce je provedena pomocí vyhledávací tabulky dle vzorce 4.5, kde  $x$  je aktuální snímek před úpravou a  $y$  je aktuální snímek po úpravě.

$$y(i) = L(x(i)) \quad (4.5)$$

Tato metoda dosahuje nejlepších výsledků při korekci snímků pokoje a pařezu. Dalším testováním jsem zjistil, že metoda je nevhodná pro obrázky, kde se vyskytují velké plochy, které mají malou dynamiku (například zatažená obloha, zasněžená krajina nebo zeď budovy). Při korekci vzniká efekt posterizace<sup>1</sup>, který velmi ruší výsledný obraz. Nejvíce je

<sup>1</sup>výrazný jasový přechod vzniklý snížením počtem jasů v důsledku manipulace s histogramem; efekt se na histogramu projevuje velkými mezerami v místech, kde původně byl plynulý přechod

vidět na velkých jednolitých plochách, kde vznikají velké artefakty. Výsledky jsou dobře viditelné na obrázcích 4.5.



Obrázek 4.5: Korekce pomocí napodobení RGB histogramu

Myslím, že i přes nežádoucí efekt posterizace u náročnějších korekcí je metoda dobře použitelná pro časosběrné video, u kterého je zajištěna nedrastická změna osvětlení nebo barevnosti. Jako příklad vezměme situaci, kdy fotíme v místnosti daný objekt s povoleným bleskem. Automatika fotoaparátu zajistí podobnou expozici pro každý snímek a vzhledem k použití blesku je i barevnost snímku téměř shodná. Také je dobré zabránit vniknutí přímého slunečního světla, např. pomocí zatažení závěsu nebo umístění bílé plachty blokuující sluneční paprsky. Výsledná odchylka jasu a barev mezi snímky je malá, ale stále rušivá. V tomto případě je použití korekce přizpůsobení histogramu dobrou volbou. Během testování metody bylo zcela odstraněno slabé blikání způsobené bleskem ve videích, kde nebyl viditelný odraz blesku. Metoda neodstranila blikání způsobené přímým odrazem blesku v kovových a jiných objektech, ale potlačila jej.

## 4.4 Gama korekce

Gama korekce je nelineární operace používaná pro kódování a dekódování jasu v obrazových a video systémech [9], [3]. Její základní křivka se vypočítá podle vzorce 4.6, kde  $\gamma$  je požadovaná hodnota gama,  $x(i)$  je hodnota jasu pixelu před korekcí a  $y(i)$  je hodnota jasu pixelu po korekci.

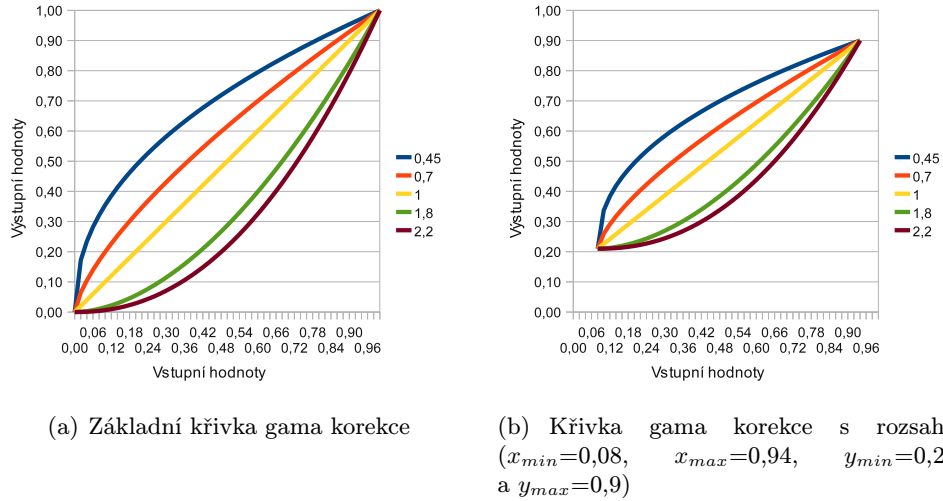
$$y(i) = x(i)^\gamma \quad (4.6)$$

Rozsah hodnot jasu je od 0 do 1. Vzorec základní křivky gama korekce je možné upravit do podoby, která přizpůsobí rozsah hodnot aktuálního snímku do rozsahu hodnot referenčního

snímku. Dále tuto korekci budu nazývat gama korekce s rozsahy. Výpočet křivky reprezentuje vzorec 4.7, kde  $x_{min}$  a  $x_{max}$  jsou minimální a maximální hodnoty jasu aktuálního snímku a  $y_{min}$ ,  $y_{max}$  jsou minimální a maximální hodnoty referenčního snímku.

$$y(i) = y_{min} + (y_{max} - y_{min}) \cdot \left( \frac{x(i) - x_{min}}{x_{max} - x_{min}} \right)^\gamma \quad (4.7)$$

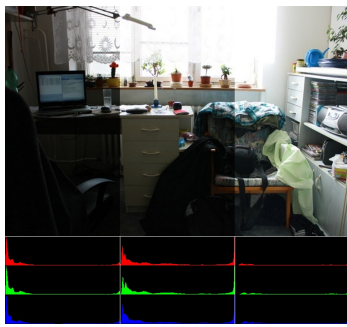
Oproti základní křivce, přizpůsobuje kontrast aktuálního snímku referenčnímu a výsledné video vypadá plynulejší. Ukázka základních křivek gama korekce a křivek gama korekce s rozsahy je vyobrazena na obrázku 4.6.



Obrázek 4.6: Křivky gama korekce a gama korekce s rozsahy pro ukázkové hodnoty gama

Gama křivka má oproti zalomené lineární křivce výhodu v její plynulosti. Hodnotu gama jsem v programu omezil na rozsah od 0,4 do 2,5. Zvolený rozsah omezuje sílu korekce, na druhou stranu zmírňuje chyby vzniklé velmi rozdílnými histogramy. Výsledný snímek vypadá přirozeně - bez efektu posterizace a korekce byla dostatečná pro testovací obrázky. Výsledky jsou zobrazeny na obrázcích 4.7. Z předchozích metod se mi gama korekce jeví jako nejuniverzálnější metoda s dobrými výsledky. U obrázku 4.7(i) je vidět špatně upravený zelený kanál. Je to způsobeno tím, že u referenčního obrázku je pár velmi světlých pixelů v zeleném kanálu. Je jich opravdu málo a proto nejsou viditelné na histogramu vlivem zaokrouhlovací chyby. Přesto tam jsou a úprava gama korekcí s rozsahem upraví výstupní hodnoty vzhledem k těmto málo pixelům. Při zvolení libovolné hodnoty gama není možnost aby se histogram posunul do tmavších odstínů, kde by jsme jej očekávali a korekce by byla správná. Možnost ignorování a prahování 5% okrajových hodnot výsledek zlepšila, ale při testech na videích byly výsledky stále horší než základní gama korekce. Pro některé scény ovšem byly výsledky lepší, protože se ve snímcích neobjevovaly malé jasově rozdílné oblasti (např. náhodné odlesky). Implementoval jsem obě metody gama korekce a nechávám na individuálním uživateli, kterou zvolí.

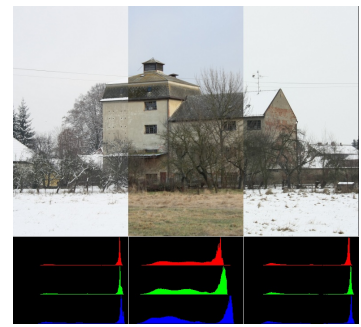
Nalezení nejvhodnější hodnoty gama provádím hrubou silou. Aplikuji postupně gama korekci na aktuální snímek v celém povoleném rozsahu po určitém kroku. V každém kroku porovnám histogramy referenčního snímku se snímkem, který by vznikl aplikací dané gama korekce. Nejlepší výsledek porovnání určuje gama hodnotu pro korekci. Porovnávací metoda byla zvolena cumulative<sup>2</sup>, protože při testech vykazovala nejlepší výsledky. Více o porovnání histogramů se věnuji v podkapitole **Metody porovnání histogramů**.



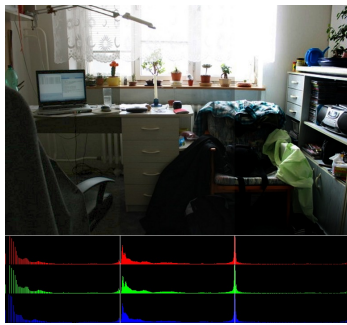
(a) Pokoj - původní



(b) Pařez - původní



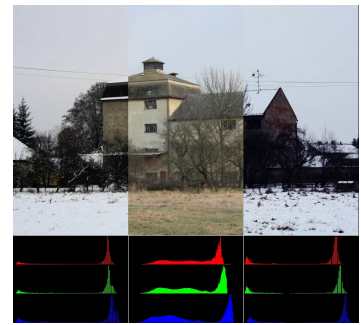
(c) Mlýn - původní



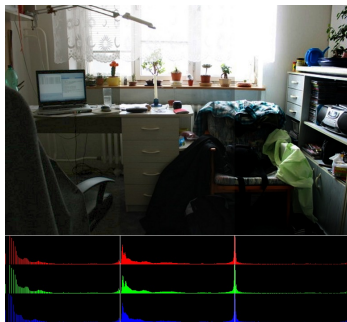
(d) Pokoj - základní gama  
(0,6; 0,6; 0,64), (2,5; 2,5; 2,5)



(e) Pařez - základní gama  
(0,94; 0,98; 1,26),  
(0,54; 0,8; 1,54)



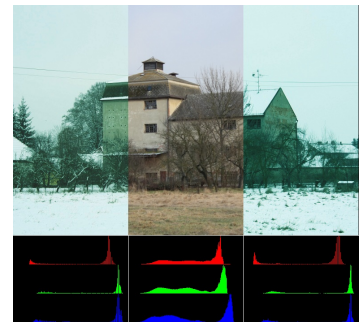
(f) Mlýn - základní gama  
(2,5; 2,5; 2,44), (2,5; 2,5; 1,88)



(g) Pokoj - gama s rozsahy  
(0,6; 0,6; 0,64), (2,5; 2,5; 2,5)



(h) Pařez - gama s rozsahy  
(0,96; 0,98; 1,24),  
(0,64; 0,84; 1,48)



(i) Mlýn - gama s rozsahy  
(2,36; 1,18; 1,44),  
(2,48; 0,98; 1,08)

Obrázek 4.7: Úprava pomocí gama korekce. Pod obrázky po korekci jsou uvedeny nalezené hodnoty gama pro jednotlivé RGB vrstvy. Nejprve pro levý obrázek, potom pro pravý obrázek.

Gama korekce je nejuniverzálnější z uvedených metod. Neposkytuje tak dobré výsledky u obrázků s malými odchylkami jako přizpůsobení histogramu, na druhou stranu si poradí i s velmi rozdílnými histogramy a nedochází k efektu posterizace.

## 4.5 Metody porovnání histogramů

U metody přizpůsobení histogramu je zapotřebí detekce nežádoucího efektu posterizace pro zlepšení výsledného dojmu. Pokud efekt posterizace detekujeme, máme na výběr korekci neprovést a vyhnout se tak zhoršení celkového dojmu, nebo můžeme pro tyto snímky aplikovat gama korekci. Dále je zapotřebí zjistit nejvhodnější hodnotu gama pro gama korekci. Oba problémy můžeme vyřešit porovnáním histogramů.

Knihovna OpenCV nabízí 4 metody pro porovnání histogramů. Jedná se o tyto metody:

- Correlation,
- Chi-square,
- Intersection,
- Bhattacharyya distance.

Pro úplnost uvádím rozsah hodnot výsledku a příslušné vzorce, kde  $H_k(i)$  je  $i$ -tý vzorek  $k$ -tého histogramu,  $d_{metoda}(H_1, H_2)$  je výsledek dané metody pro histogramy  $H_1$  a  $H_2$  a  $N$  je šířka histogramu. Údaje jsou převzaty z [1].

Hodnota *Correlation* je vypočítána podle vzorce 4.8. Větší hodnota reprezentuje větší shodu. Dokonalá shoda vrací 1 a totální neshoda vrací -1. Hodnota 0 indikuje žádnou statisticky zjititelnou lineární závislost.

$$H'_k(i) = H_k(i) - \frac{1}{N} \left( \sum_{j=0}^{N-1} H_k(j) \right)$$
$$d_{correlation}(H_1, H_2) = \frac{\sum_{i=0}^{N-1} H'_1(i) \cdot H'_2(i)}{\sqrt{\sum_{i=0}^{N-1} H'_1(i)^2 \cdot H'_2(i)^2}} \quad (4.8)$$

U *chi-square* (vzorec 4.9) nižší hodnota reprezentuje větší shodu. Dokonalá shoda vrací 0 a čím je číslo vyšší, tím jsou histogramy rozdílnější (horní hranice není stanovena, závisí na velikosti histogramu).

$$d_{chi-square}(H_1, H_2) = \sum_{i=0}^{N-1} \frac{(H_1(i) - H_2(i))^2}{H_1(i) + H_2(i)} \quad (4.9)$$

Vzorec 4.10 slouží pro výpočet *Intersection*. Vysoký výsledek určuje velkou shodu a nízký výsledek určuje malou shodu. Pokud jsou histogramy normalizovány na hodnotu 1, potom dokonalá shoda vrací 1 a totální neshoda vrací 0.

$$d_{intersection}(H_1, H_2) = \sum_{i=0}^{N-1} \min(H_1(i), H_2(i)) \quad (4.10)$$

U *Bhattacharyya distance* (vzorec 4.11) nízká hodnota indikuje větší shodu a vysoká



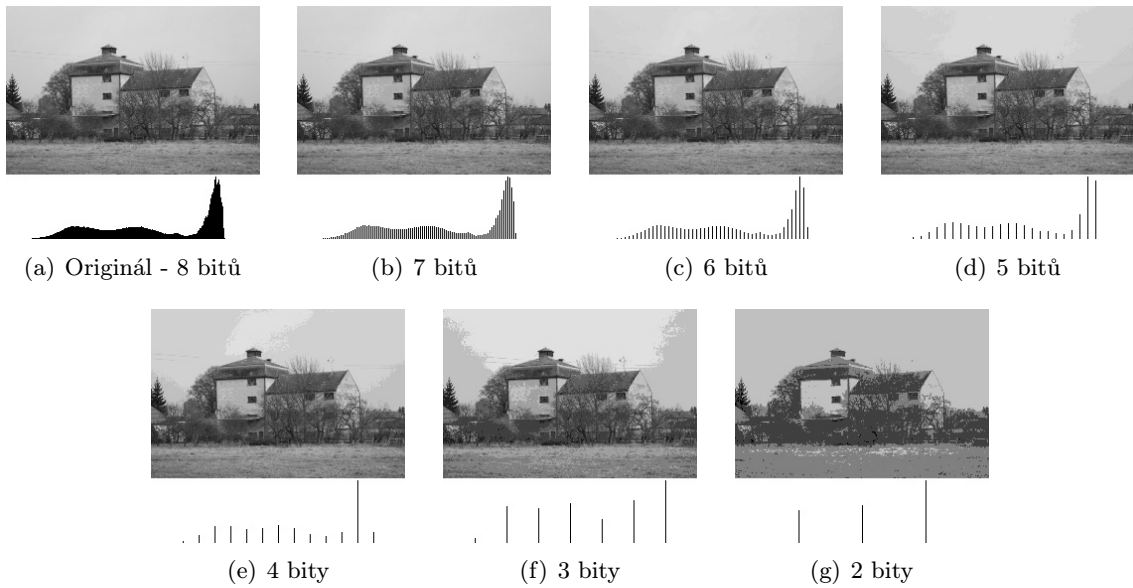
hodnota indikuje malou shodu. Dokonalá shoda vrací 0 a totální neshoda vrací 1.

$$d_{Bhattacharyya}(H_1, H_2) = \sqrt{1 - \frac{\sum_{i=0}^{N-1} \sqrt{H_1(i) \cdot H_2(i)}}{\sqrt{\sum_{i=0}^{N-1} H_1(i) \cdot \sum_{i=0}^{N-1} H_2(i)}}} \quad (4.11)$$

Při porovnávání jednotlivých funkcí je jejich nesoulad ve vrácených hodnotách matoucí. Rozhodl jsem se, že výsledky sjednotím. Všechny metody budou při dokonalé shodě vracet hodnotu 0 a při neshodě kladné číslo, které bude narůstat s narůstající mírou neshody. Metody chi-square a Bhattacharyya distance tuto podmínku splňují. V programu pracuji s histogramy normalizovanými na hodnotu 1, takže metody correlation a intersection mohou být přepočítány stejným vzorcem 4.12, kde  $d'$  určuje výsledek přepočtu původního výsledku  $d$ .

$$d' = 1 - d \quad (4.12)$$

Pro porovnání jednotlivých metod jsem zvolil stejné obrázky jako pro testování korekce. Jsou to obrázky pokoje s rozdílnou expozicí (obrázek 4.5(a)), pařezu s odlišným nastavením vyvážení bíle (obrázek 4.5(b)), dále mlýnu foceného v jiné dny (obrázek 4.5(c)) a mlýnu, u kterého je vybrána jen zelená vrstva a rozdíl spočívá v simulaci nižší bitové hloubky (obrázky 4.8).



Obrázek 4.8: Zelená složka obrázku mlýna a zobrazení efektu posterizace pro nižší bitovou hloubku

Metoda	Rozdílnost obrázku	Pokoj						Pařez - VB=zataženo					
		+2 EV			-2 EV			VB=Automatika			VB=žárovka		
		R	G	B	R	G	B	R	G	B	R	G	B
Metoda	Vrstva												
	Intersection	0,30	0,32	0,29	0,30	0,32	0,32	0,06	0,02	0,26	0,58	0,17	0,42
	Correlation	0,48	0,55	0,51	0,30	0,31	0,33	0,01	0,00	0,21	0,88	0,15	0,55
	Chi-Square	0,30	0,34	0,32	0,31	0,34	0,38	0,01	0,00	0,18	0,84	0,11	0,46
	Bhattacharyya	0,31	0,33	0,32	0,32	0,34	0,37	0,05	0,02	0,22	0,53	0,17	0,36
	cumulative	0,15	0,15	0,14	0,20	0,20	0,20	0,02	0,01	0,08	0,21	0,07	0,15
	cumulative Sqrt	0,38	0,38	0,37	0,44	0,44	0,44	0,11	0,06	0,23	0,37	0,22	0,32
	10.cumulative <sup>2</sup>	0,27	0,27	0,24	0,43	0,45	0,42	0,01	0,00	0,15	0,90	0,10	0,46
Metoda	Rozdílnost obrázku	Mlýn - foceno 21.1.						Mlýn - 8 bitů					
	Vrstva	12.4.			25.4.			7 bitů			4 bitů		
	Intersection	0,61	0,62	0,60	0,63	0,61	0,56	0,50	0,76	0,87	0,94	0,98	0,99
	Correlation	1,11	1,12	1,10	1,12	1,11	1,06	0,38	0,62	0,73	0,82	1,03	1,00
	Chi-Square	1,01	1,03	0,98	1,02	1,00	0,86	0,67	1,22	1,55	1,77	1,92	1,95
	Bhattacharyya	0,64	0,67	0,64	0,63	0,65	0,60	0,54	0,71	0,80	0,87	0,94	0,94
	cumulative	0,17	0,17	0,19	0,13	0,13	0,14	0,00	0,00	0,01	0,02	0,04	0,06
	cumulative Sqrt	0,33	0,35	0,39	0,27	0,28	0,31	0,03	0,04	0,07	0,10	0,14	0,19
	10.cumulative <sup>2</sup>	0,64	0,59	0,56	0,47	0,45	0,35	0,00	0,00	0,00	0,01	0,04	0,11

Tabulka 4.2: Metody porovnání histogramu - metody correlation a intersection přepočítány podle vzorce 4.12 a metoda cumulative<sup>2</sup> násobena číslem 10

Rozdílnost obrázku	Pokoř										Pařez - VB=zatařeno					
	+2 EV					-2 EV					VB=Automatika					
	R	G	B	R	G	R	G	B	R	G	R	G	B	R	G	B
Metoda	Vřstva															
	Intersection	0,30	0,32	0,30	0,31	0,32	0,32	0,32	0,06	0,02	0,06	0,02	0,27	0,59	0,17	0,43
	Correlation	0,43	0,49	0,45	0,27	0,27	0,27	0,29	0,01	0,00	0,01	0,00	0,18	0,79	0,13	0,49
	Chi-Square	0,15	0,17	0,16	0,16	0,18	0,18	0,19	0,01	0,00	0,01	0,00	0,09	0,43	0,06	0,24
	Bhattacharyya	0,32	0,35	0,34	0,34	0,36	0,36	0,39	0,06	0,02	0,06	0,02	0,23	0,56	0,19	0,38
	cumulative	0,71	0,72	0,68	0,95	0,97	0,97	0,93	0,08	0,03	0,08	0,03	0,39	1,00	0,35	0,72
	cumulative Sqrt	0,85	0,85	0,83	0,99	1,00	1,00	0,98	0,24	0,14	0,24	0,14	0,52	0,82	0,50	0,72
	cumulative <sup>2</sup>	0,30	0,30	0,27	0,48	0,50	0,50	0,47	0,01	0,00	0,01	0,00	0,17	1,00	0,11	0,51

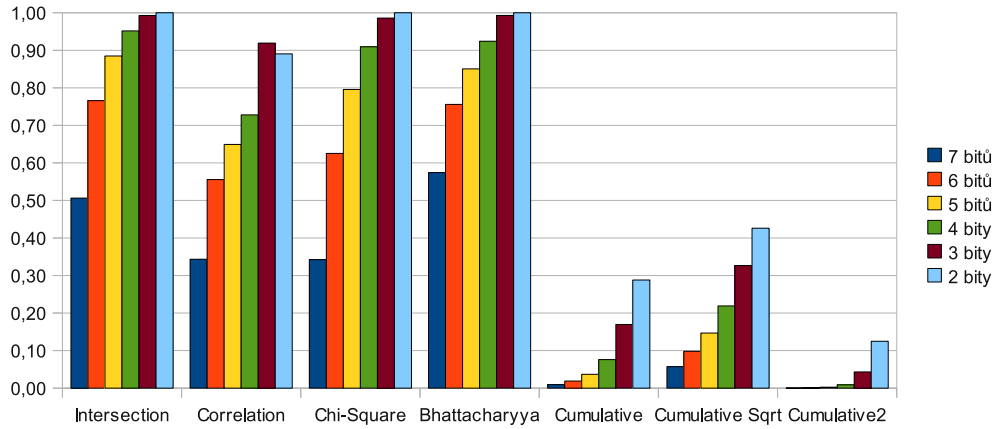
Rozdílnost obrázku	Mlŷn - foceno 21.1.										Mlŷn - 8 bitů					
	12.4.					25.4.					7 bitů					
	R	G	B	R	G	R	G	B	R	G	R	G	B	R	G	B
Metoda	Vřstva															
	Intersection	0,62	0,62	0,61	0,64	0,64	0,62	0,57	0,51	0,77	0,51	0,77	0,88	0,95	0,99	1,00
	Correlation	0,99	1,00	0,98	1,00	1,00	0,99	0,94	0,34	0,56	0,34	0,56	0,65	0,73	0,92	0,89
	Chi-Square	0,52	0,53	0,50	0,52	0,52	0,51	0,44	0,34	0,63	0,34	0,63	0,80	0,91	0,99	1,00
	Bhattacharyya	0,68	0,71	0,68	0,67	0,67	0,69	0,63	0,57	0,76	0,57	0,76	0,85	0,92	0,99	1,00
	cumulative	0,80	0,83	0,92	0,60	0,60	0,63	0,66	0,01	0,02	0,01	0,02	0,04	0,08	0,17	0,29
	cumulative Sqrt	0,75	0,80	0,89	0,61	0,61	0,64	0,70	0,06	0,10	0,06	0,10	0,15	0,22	0,33	0,43
	cumulative <sup>2</sup>	0,71	0,65	0,62	0,52	0,52	0,50	0,39	0,00	0,00	0,00	0,00	0,00	0,01	0,04	0,12

Tabulka 4.3: Metody porovnání histogramu - poměřové výsledky vřhledem k maximální hodnotě dané metody v tabulce 4.2



Výsledky porovnávacích metod pro testovací obrázky jsou uvedeny v tabulce 4.2. Výsledky jsou ale zavádějící, protože rozmezí možného výsledku není pro všechny metody stejné. V tabulce 4.3 jsou zapsány poměrové výsledky vzhledem k maximální hodnotě dané metody v tabulce 4.2. Poměrové výsledky jsou názornější, protože dané hodnoty jsou ve stejném rozsahu a zajímá nás, jak si daná metoda stojí vzhledem k testovacím obrázkům namísto k teoretické totální neshodě. Dále v textu budu pracovat s poměrovými hodnotami, pro které jsou grafy lépe čitelnější.

Gama korekce potřebuje ke své funkci porovnávací metodu, která je schopna vyhodnotit rozdíl mezi referenčním histogramem a histogramem vzniklým po korekci. Korekci vzniká slabý efekt posterizace, z čehož vyplývá, že zvolená metoda musí umět porovnávat histogramy, které obsahují mezery. Vhodnou metodu zvolíme za pomoci testovacích obrázků simulujících nižší bitovou hloubku. Obrázky s nižší bitovou hloubkou se vyznačují posterizačním efektem a přitom se jedná o stejné obrázky jako je referenční obrázek. Grafické znázornění výsledků můžeme vidět na obrázku 4.9. Z uvedeného grafu vyplývá,



Obrázek 4.9: Grafické znázornění poměrových výsledků jednotlivých metod pro obrázky s nižší bitovou hloubkou (viz obrázky 4.8)

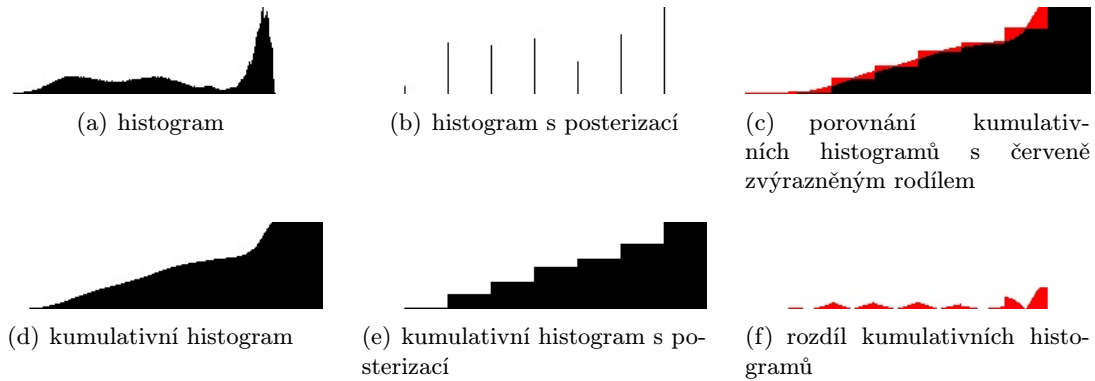
že porovnávací metody implementované v knihovně OpenCV vraceli největší rozdíl právě u obrázků zasažených posterizací. Tyto metody jsou tedy pro účely gama korekce nevhodné. Pro jistotu jsem provedl test na obrázku pokoje s rozdílnými expozicemi a ani jedna metoda si nedokázala poradit s podexponovaným obrázkem (výsledek výpočtu gama hodnoty byl vždy 1.0 - beze změny).

Navrhl jsem tedy porovnávací metodu využívající kumulativního histogramu, která je odolná vůči posterizaci. Metodu jsem nazval *cumulative* a spočívá v součtu rozdílů jednotlivých vzorků kumulativního histogramu, viz vzorec 4.13.

$$d_{cumulative}(H_1, H_2) = \frac{\sum_{i=0}^{N-1} |(H_1(i) - H_2(i))|}{N - 1} \quad (4.13)$$

Nízká hodnota indikuje větší shodu a vysoká hodnota indikuje malou shodu. Dokonalá shoda vrací 0 a totální neshoda vrací 1. Kumulativní histogram má tu vlastnost, že při výskytu posterizace neobsahuje mezery a tudíž chyba způsobená posterizací není tak výrazná a snižuje se do výskytu následujícího použitého jasu. Obrázek 4.10 zobrazuje rozdíly

mezi kumulativním histogramem originálního obrázku a kumulativním histogramem stejného obrázku s efektem posterizace.



Obrázek 4.10: Ukázka výpočtu metody cumulative pro histogram obrázku a histogram stejného obrázku s posterizací - výsledek je součet rozdílů

Další variantou metody cumulative je metoda nazvaná *cumulative Sqrt*, která při výpočtu rozdílu navíc vypočítá druhou odmocninu, viz vzorec 4.14. Nízká hodnota indikuje větší shodu a vysoká hodnota indikuje malou shodu. Dokonalá shoda vrací 0 a totální neshoda vrací 1. Oproti metodě cumulative je tato metoda více citlivá na malé změny v histogramu a větší změny se tolik neprojeví. Je méně odolná na efekt posterizace, ale oproti metodám v OpenCV si s posterizací u gama korekce dokáže poradit. Metoda je tedy vhodná pro případy, kdy má být kladen větší důraz na malé lokální změny v obraze.

$$d_{cumulative\ sqrt}(H_1, H_2) = \frac{\sum_{i=0}^{N-1} \sqrt{|(H_1(i) - H_2(i))|}}{N - 1} \quad (4.14)$$

Poslední variantou je *cumulative<sup>2</sup>*, která při výpočtu navíc vypočítá druhou mocninu, viz vzorec 4.15. Rozsah hodnot a použití je stejné jako u metody cumulative. Rozdíl spočívá v menší citlivosti na malé změny histogramu. Větší důraz je kladen na globální změny.

$$d_{cumulative^2}(H_1, H_2) = \frac{\sum_{i=0}^{N-1} (|H_1(i) - H_2(i)|)^2}{N - 1} \quad (4.15)$$

V průběhu testování gama korekce měly metody cumulative a odvozené metody velmi podobné výsledky pro většinu snímků. Pro obrázky s obtížnou korekcí jsem zjistil, že každá metoda selhávala na jiný typ scén. Při testování na různá časosběrná videa se mi nejvíce líbily výsledky korekce u metody cumulative<sup>2</sup>, proto jsem ji použil pro výpočet hodnoty gama.

Porovnávací metody mohou být také použity jako detekce vzniku posterizace u metody přizpůsobení histogramu. Nyní je náš cíl opačný než v případě gama korekce. Hledáme funkci, která by při vzniku posterizace vracela hodnotu reprezentující neshodu. Požadavkem je, aby zvolená metoda vracela největší neshodu právě pro efekt posterizace. Metody cumulative a odvozené metody můžeme z dalšího porovnávání vyloučit, protože jsou navrženy, aby byly odolné vůči posterizaci. Při pohledu na obrázek 4.9 vidíme, že metody Intersection, Chi-Square a Bhattacharyya tento požadavek splňují. Correlation nemá největší neshodu pro efekt posterizace a navíc pro 2-bitovou posterizaci je neshoda menší

než pro 3-bitovou, je tedy pro její detekci nevhodná. Slabou posterizací nevnímáme rušivě a proto nás zajímá až silnější efekt posterizace. V grafu by měl být zřetelný rozdíl mezi slabou a silnou posterizací, který nejvíce splňuje metoda Chi-Square a z tohoto důvodu se mi jeví jako nejvhodnější pro detekci již vzniklé posterizace. Jako práh pro detekci jsem nejdříve zvolil hodnotu 0,7 z poměrových výsledků, ale detekce nebyla úspěšná (označila vhodně upravený obrázek, i když posterizace byla vizuálně velmi výrazná). Snížení prahu nepomáhalo a metoda Chi-Square se ukázala pro detekci nevhodná. Při testování zbylých metod výsledky také nebyly uspokojivé.

Rozhodl jsem se tedy pro řešení zahrnující prahování podle rozdílnosti histogramů před korekcí. Pro lepší vizuální posouzení jednotlivých metod jsou na obrázcích 4.11(a), 4.11(b) a 4.11(c) zobrazeny grafy poměrových výsledků z tabulky 4.3 pro testovací obrázky. Víme, že korekce napodobením histogramu uspěla u obrázku pokoje a pařezu, proto by u těchto obrázků měly být výsledkem nízké hodnoty, kdežto na obrázku mlýna byla viditelná silná posterizace a výsledky by měly být vysoké. Nejzřetelnější poměr rozdílů mezi výsledky je u metody Chi-Square a jako práh volím hodnotu odpovídající průměru výsledků pro obrázek pokoje, výsledný práh po přepočtu je 0,3. Při testování jsem zkoušel i ostatní metody a jiné prahy, ale uvedená metoda a zvolený práh se mi zdáli opticky nejvhodnější. Testy jsem vždy prováděl na videích, které jsou přiložené k této diplomové práci.

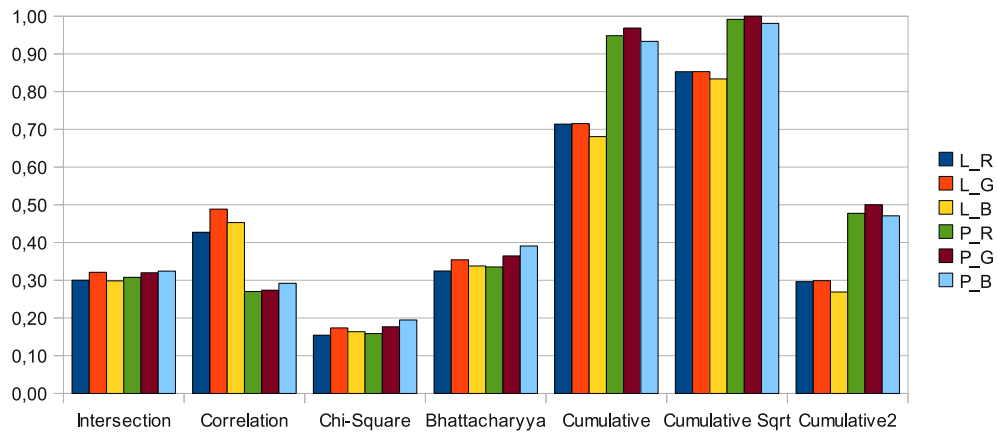
## 4.6 Korekce pomocí barevného prostoru HSV

Dále jsem se rozhodl otestovat korekci pomocí barevného modelu HSV, který nabízí oddělenou práci s barevným odstínem (H), jasnem (V) a saturací (S). Jasovou složku můžeme zpracovat metodou přizpůsobení histogramu nebo pomocí gama korekce. Složka odstínu je upravena dle rovnice 4.16, kde  $\mu_r$  je střední hodnota odstínu referenčního obrázku,  $\mu_a$  je střední hodnota odstínu aktuálně zpracovávaného obrázku,  $x(i)$  je hodnota odstínu daného pixelu aktuálního obrázku před úpravou,  $y(i)$  je výsledná hodnota pixelu  $x(i)$  po úpravě a  $v(i)$  je pomocná hodnota při výpočtu. Maximální hodnota odstínu je 180 a minimální je 0. Složka saturace zůstává beze změny.

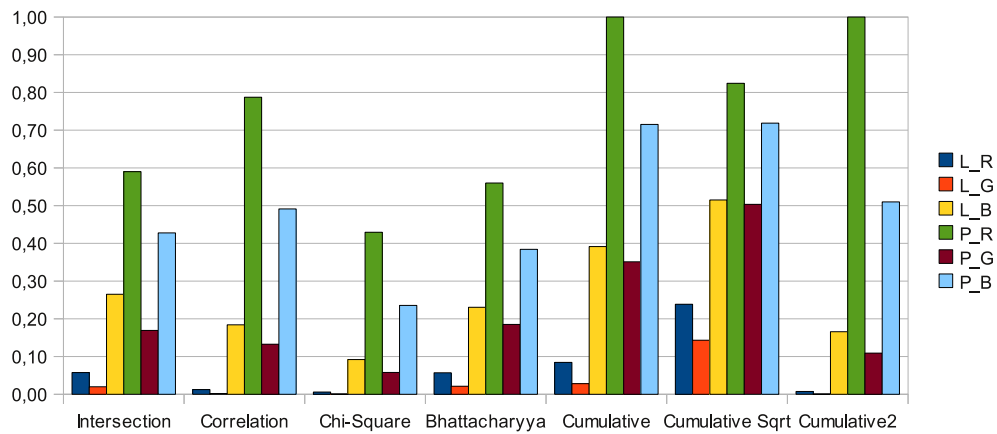
$$v(i) = x(i) + (\mu_r - \mu_a)$$

$$y(i) = \begin{cases} v(i) + 180 & \text{v případě } v(i) < 0 \\ v(i) & \text{v případě } 0 \leq v(i) \leq 180 \\ v(i) - 180 & \text{v případě } v(i) > 180 \end{cases} \quad (4.16)$$

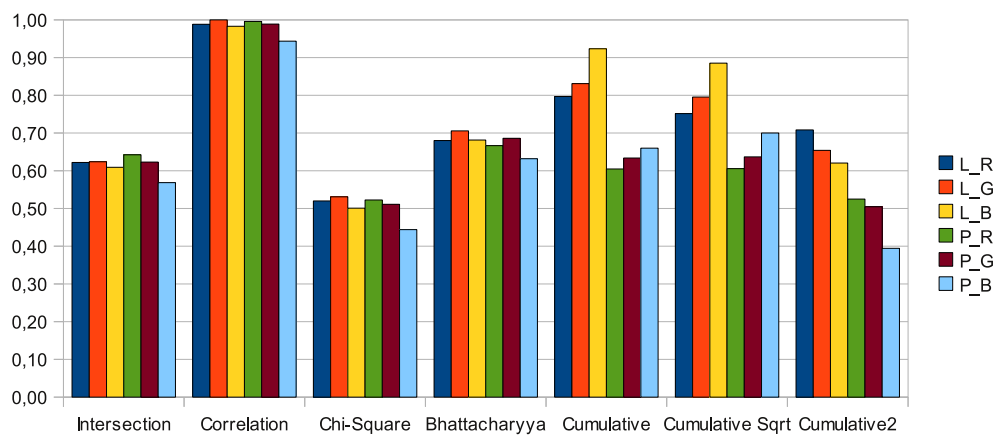
Při testování se vyskytoval problém s posunem barevného odstínu do nežádoucích odstínů. Je to pravděpodobně způsobeno rovnoměrným rozložením základních barev RGB ve složce odstínu. Tento problém by možná mohla vyřešit volba jiného barevného modelu pracujícího odděleně s jasovou složkou a barevným oddílem. Mě se tento barevný model nepodařilo nalézt a proto jsem tuto metodu opustil. Výsledky můžete vidět na obrázku 4.12. Na obrázku pokoje jsou výsledky uspokojivé, protože mezi referenčním i aktuálním snímkem je stejný barevný odstín. Při pohledu na obrázek pařezu je vidět chybný barevný posuv. Chyba je obzvlášť výrazná na listech rostlin. Výsledky byly kvůli chybnému posunu barev nepoužitelné a do programu jsem tuto metodu neimplementoval.



(a) Pokoj (viz obrázky 4.7(a))

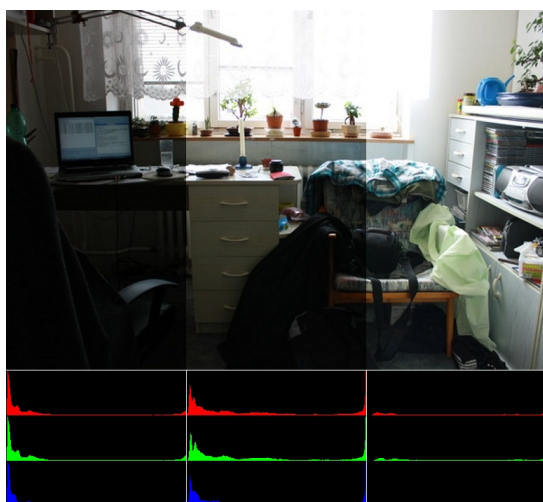


(b) Pařez (viz obrázky 4.7(b))

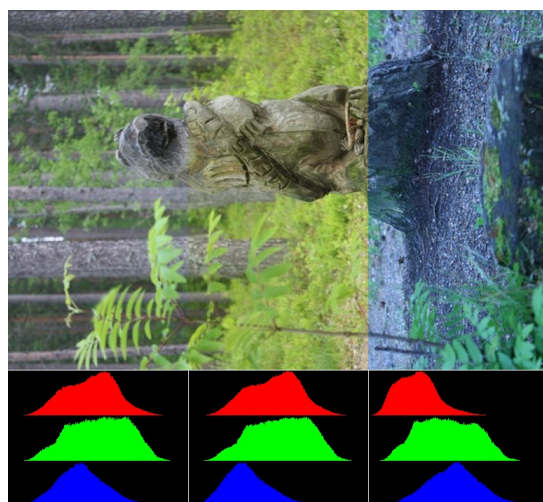


(c) Mlýn (viz obrázky 4.7(c))

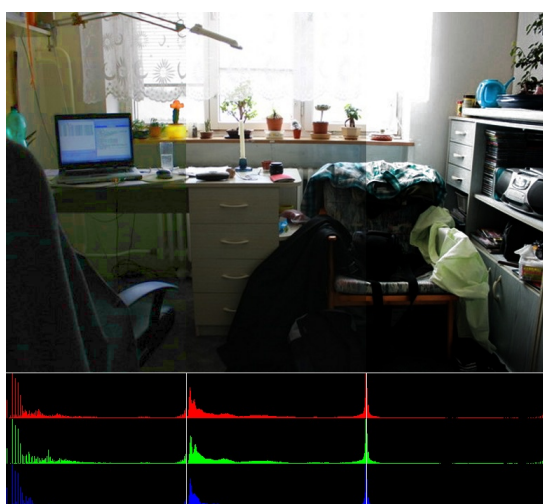
Obrázek 4.11: Grafické znázornění poměrových výsledků jednotlivých metod pro testovací obrázky — L\_X značí levý obrázek a barevnou vrstvu X, P\_X značí pravý obrázek a barevnou vrstvu X



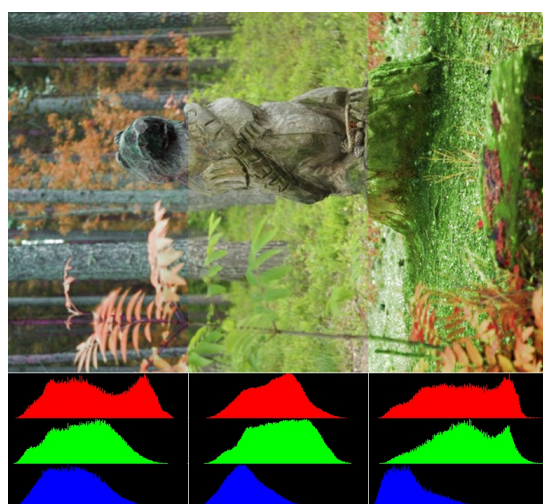
(a) Pokoj - původní obrázky



(b) Pařez - původní obrázky



(c) Pokoj - obrázky po korekci



(d) Pařez - obrázky po korekci

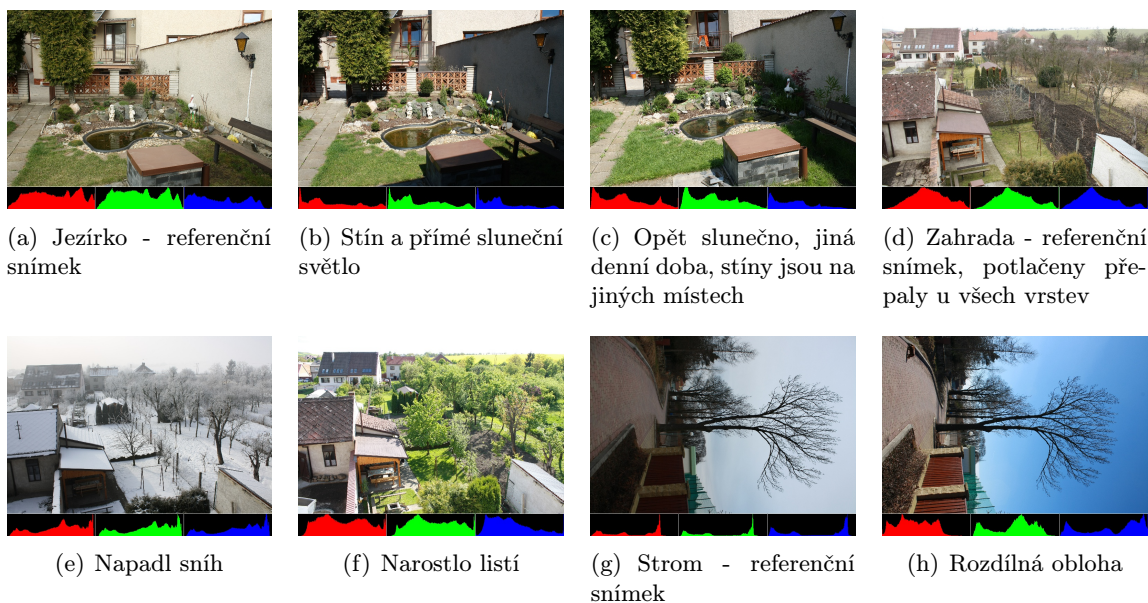
Obrázek 4.12: Korekce pomocí barevného prostoru HSV

## 4.7 Problémy a podpůrné prostředky pro zlepšení korekce

Během sběru testovacích snímků se postupně začaly objevovat problémy, které způsobovaly nesprávnou korekci. Největší problémy jsem zaznamenal u slunečního dne, kdy focení v různou denní dobu vrhalo odlišné stíny a tvořilo velký kontrast scény. Když je polovina pozorovaného objektu zahalena stínem a na druhou polovinu objektu dopadá přímé sluneční světlo, musíme se spokojit s kompromisem, kdy korekce upraví snímek tak, aby jeho celkový jas co nejvíce napodobil referenční snímek. Stále ale bude viditelný nepříjemný kontrast stínu a osvětlené části, protože metody upravují obraz globálně. Dále rozdílná obloha velmi komplikuje korekci. Snímky mají rozdílné barevné zabarvení a přepálená obloha dokáže velmi ztmavit výsledný snímek v případě gama korekce a způsobí silnou posterizaci u metody přizpůsobení histogramu. Ukázky problémových scén můžete vidět na obrázcích 4.13.

Vlivem ročního období dochází ve venkovních scénách k velkým změnám, jako je na-



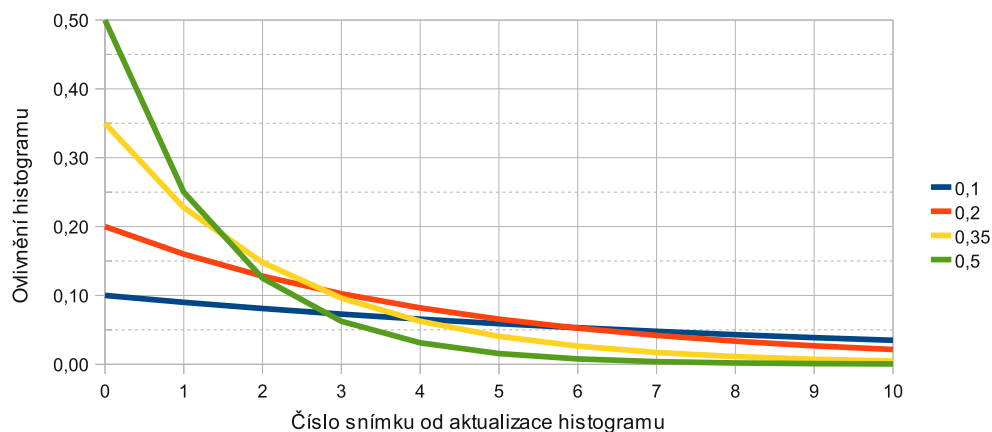


Obrázek 4.13: Ukázka různých vlivů, které ztěžují korekci. U některých snímků bylo zapotřebí potlačit maximální hodnotu jasu v histogramu kvůli přepálené obloze. U upravených snímků je uveden údaj *potlačeny přepaly* spolu s informací, kterých vrstev se úprava týká.

padení sněhu, růst rostlin, opadání listí, apod. Tyto rozdíly jsou přirozené a nechceme je odstraňovat, nýbrž zachovat. Pro vypořádání se s vlivy různého ročního období můžeme postupně aktualizovat referenční histogram pomocí vzorce 4.17, kde  $w$  je váha adaptace,  $H_r$  je referenční histogram a  $H_a$  je histogram aktuálního snímku.

$$H_r(i) = w \cdot H_a(i) + (1 - w) \cdot H_r(i) \quad (4.17)$$

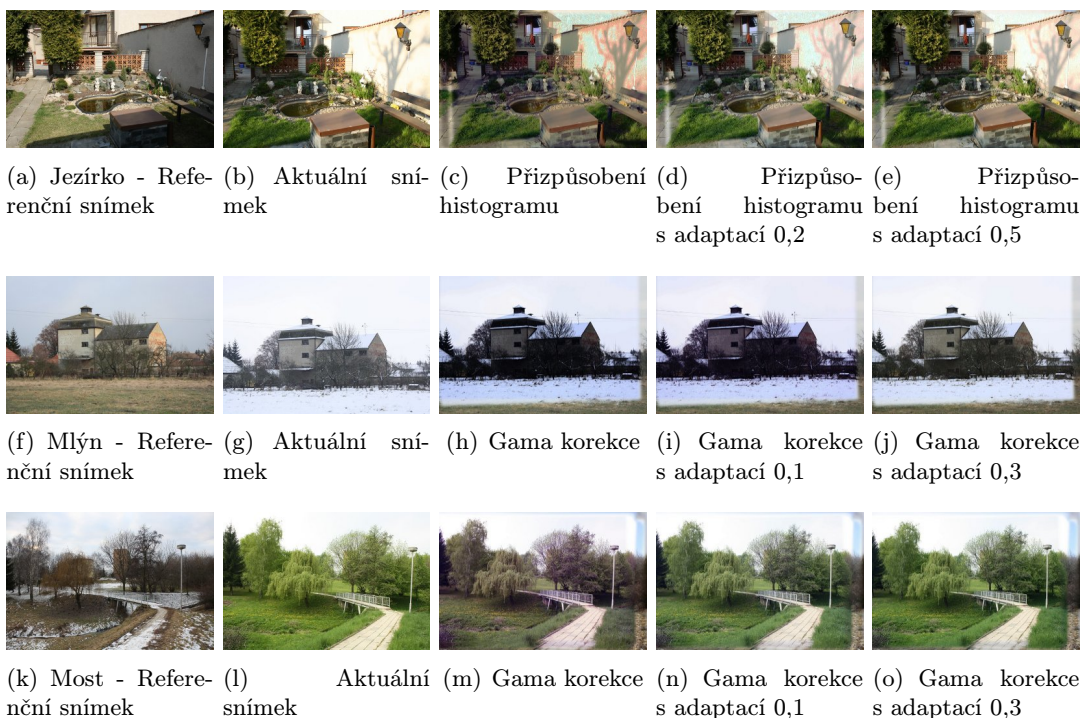
Ovlivnění referenčního histogramu aktuálním snímkem v čase je znázorněno na obrázku 4.14. Korekce je utlumena o velikost váhy adaptace, na druhou stranu se referenční histogram



Obrázek 4.14: Ovlivnění referenčního histogramu aktuálním snímkem v závislosti na váze adaptace v čase. V grafu jsou vyneseny hodnoty pro ukázkové váhy.

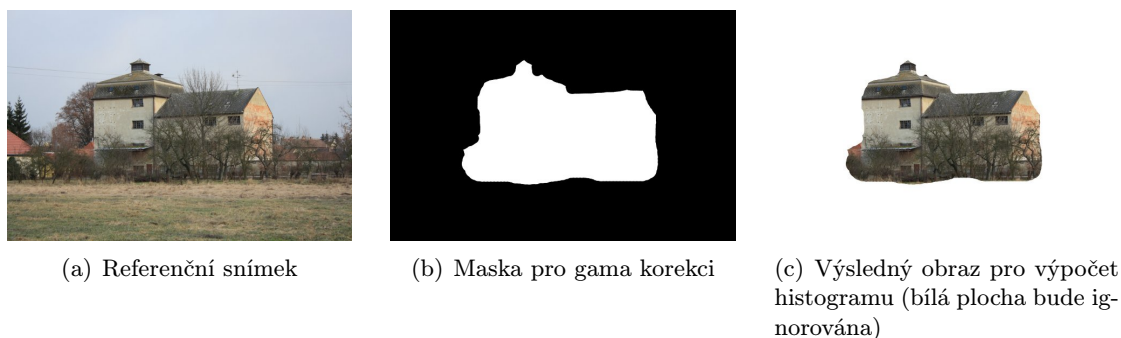
postupně přizpůsobuje působícím vlivům a starší hodnoty se pozvolna vytrácí. Při testo-

vání metody gama korekce aktualizace histogramu odstranila hlavně chybný barevný odstín (obrázek 4.15(j)) a zlepšila podání barev při přechodu z pochmurné zimy do barevného jara (obrázek 4.15(o)). U metody přizpůsobení histogramu se potlačil vznik posterizace (obrázek 4.15(e)). Každé video je rozdílné a s ideální hodnotou váhy je zapotřebí trochu experimentovat. Jako začáteční hodnoty váhy adaptace doporučuji pro dlouhodobé časosběry volit kolem 0,25 a pro krátkodobé časosběry 0,1 nebo 0,05.



Obrázek 4.15: Rozdíl korekce s aktualizací histogramu. Snímky jsou stabilizovány a až potom je uplatněna korekce.

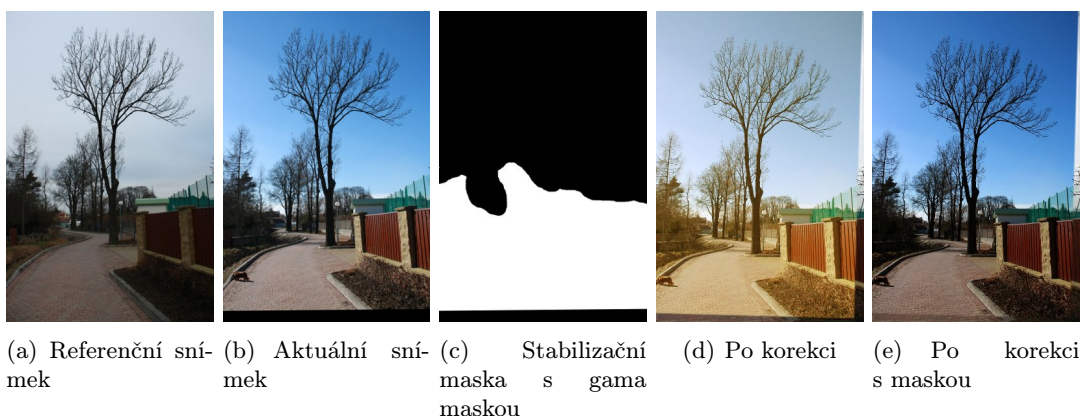
Pro eliminaci vlivu rozdílné oblohy můžeme použít masku pro výpočet histogramu, viz obrázky 4.16. Mějme ale na paměti, že maska vyloučí část obrazu a výsledný histogram



Obrázek 4.16: Ukázka masky pro gama korekci

bude postrádat jasy, které se budou korekcí upravovat. Proto je maska vhodná především pro úpravu pomocí základní gama křivky. Gama křivka s rozsahy může provést nesprávné

upravení výstupního rozsahu a metoda přizpůsobení histogramu upraví oblast masky a ignorované okolí bude ve většině případů postiženo silnou posterizací. Dále vyvstává otázka v jaké části zpracování časosběrného videa je vhodné provádět korekci. Pro možnost použití masky při korekci jasu a barev je nezbytné, aby byly snímky před úpravou zarovnány. Dále je nutné, aby byla známa poloha upraveného obrázku a mohly být vyloučeny oblasti, kde již obrázek není. Pro polohu obrázku vytvářím další masku. Výsledná maska je tvořena minimem jednotlivých pixelů z obou předchozích masek. Výslednou masku následně použiji pro výpočet histogramu aktuálního snímku. Ukázku korekce pomocí masky můžete vidět na obrázku 4.17, kde je pomocí masky ignorován vliv oblohy, takže je naším cílem zajistit nerušivé přechody ve spodní části obrazu. Maska v tomto ohledu velmi pomohla. Samozřejmě se můžete používat společně masku pro výpočet histogramu a aktualizaci histogramu.



Obrázek 4.17: Ukázka gama korekce s maskou nastavenou na část snímku bez oblohy



## Kapitola 5

# Implementace

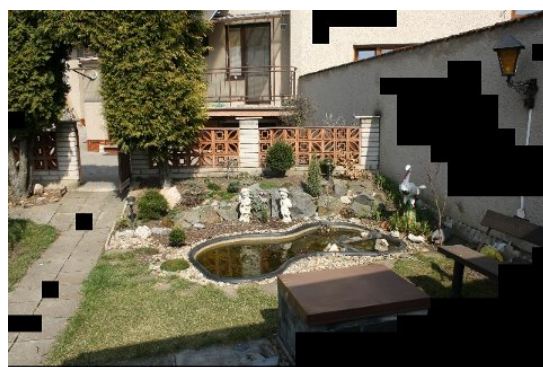
Pro implementaci jsem zvolil jazyk C++ s využitím multiplatformního toolkitu Qt a multiplatformní knihovnu pro zpracování obrazu OpenCV. Toolkit Qt je použit pro práci s adresáři a pro možnost zjištění všech dostupných obrázků v adresáři.

Aplikace je spouštěna z konzole a umí na vstupu pracovat s obrázky nacházejícími se v určené složce nebo již existujícím videozáznamem. Výstup může být uložen opět do videozáznamu nebo jako série obrázků. Snažil jsem se aplikaci vytvořit pomocí tříd, které se dají použít i pro případné grafické rozhraní. Veškeré nastavení aplikace obstarává jedna třída, takže není problém do budoucna aplikaci rozšířit o projektové soubory.

Zarovnání snímků pomocí významných bodů můžeme značně urychlit redukcí detekovaných bodů. Redukce se provádí řadícím algoritmem insert sort. Řadí se významné body a jako porovnání funkce pro řazení slouží porovnání hodnot Hessianů, které určují významnost daných bodů. Společně s řazením významných bodů musíme symetricky řadit deskriptory k významným bodům. Až seřadíme požadované množství bodů, uvolníme nepotřebné body a deskriptory. U velkých obrázků může dojít k detekci několika desítek tisíc bodů a redukce počtu detekovaných bodů značně urychlí proces vyhledání korespondencí.



(a) aktuální obrázek



(b) Aktuální obrázek s odstraněnými částmi obrazu s nízkým rozptylem

Obrázek 5.1: Ukázka vynechání částí obrázku s nízkým rozptylem pro metodu vyhledání vzoru

Nejvíce časově náročná operace v programu je vyhledání korespondencí částí obrazu metodou vyhledání vzoru. Vyhledání vzoru se provádí postupně po čtvercích. Pro čtverce, které obsahují jednolitý obrázek, není zapotřebí vyhledávat jejich protějšek v aktuálním

snímku. Jako příklad vezměme oblohu. Obloha je problematická a její vyhledávání bude vést k chybným výsledkům. Před samotným vyhledáním vzoru nejdříve zjistím rozptyl hodnot vzoru. Pokud rozptyl nesplňuje určený práh, jeho obsah má nízkou dynamiku a můžeme jej z vyhledávání vyloučit. Prah jsem zvolil experimentálně tak, aby oblaka, stěny a podobné části obrazu byli z vyhledávání korespondencí vyloučeny. Obrázky 5.1 znázorňují vynechané části obrazu pro vyhledání vzoru.

Pro lepší představu, jak rychlý děj je na videozáznamu zachycen, jsem implementoval vkládání informací o čísle aktuálního snímku. Dále, pokud jsou na vstupu obrázky, můžeme do výstupního snímku zapsat čas poslední úpravy souboru s obrázkem.

## Kapitola 6

# Vyhodnocení výsledků a návrhy na budoucí rozšíření

Vytvořená aplikace odstraňuje základní nedostatky vzniklé dlouhými prodlevami mezi jednotlivými snímky a focením z ruky. Volbou parametrů aplikace jsou k dispozici nástroje zajišťující zarovnání snímků a korekce jasu a barev. U obou nástrojů jsou implementovány 2 různé metody pro jejich realizaci. Pro přesnější zpracování má uživatel možnost specifikovat oblast výpočtu pomocí masky.

Testování probíhalo na 6 různých testovacích sadách, které byli snímány s prodlevou jednoho dne a více dnů. Snažil jsem se zachytit běžné situace, které mohou nastat. Některé testovací sady jsem vytvářel podle dříve uvedených doporučení. Výsledky na těchto sadách byly dobré. Zarovnání snímků probíhalo bez problémů pro většinu snímků. Vzhledem k použití algoritmu RANSAC pro zarovnání neuvádím výsledky. Výsledky jsou totiž pokaždé jiné. Ve většině případů zarovnání selhalo u každého 10 až 20 snímku. To považuji za dobrý výsledek, když uvážím, jaké změny se v některých snímcích odehrávaly.

Zvolil jsem si jednu testovací sadu, ve které jsem záměrně vytvářel nedostatky. Nastavoval jsem chybné vyvážení bílé a měnil expozici. Předpokládal jsem znatelné selhávání implementovaných metod. K výraznému selhávání zarovnání snímků nedocházelo, pokud byli změněné hodnoty v určitých mezích. S vyvážením bílé jsem nezaznamenal problém. Problém byl s rozdílnou expozicí. Změna expozice o +1 EV se ukázala problematickou při korekci jasu a barev. Problémy nastávaly kvůli přepalům oblohy. U snímků, kde nebyla obloha, nebyl problém tak výrazný. S podexponovaným snímkem si korekce poradila mnohem lépe. Detekce významných bodů také nezaznamenala výraznější problémy. Problémy se začaly objevovat až u expozic lišících se o více než 1 EV.

Při testování detekce významných bodů, jsem narazil na situaci, kdy tato metoda selhala. Jednalo se o časosběrné video, které bylo snímáno z okénka letadla a jako stabilní oblast pro aplikaci masky jsem zvolil křídlo letadla. Křídlo ale nemělo žádné výrazné vzory a významné body se objevovali spíše na okraji křídla. Vlivem jednolité barvy dále selhávala i korespondence významných bodů. Navíc křídlo zabíralo malou plochu snímku. Nastavení masky také nepomohlo.

Pro zarovnání snímků nabízím následující doporučení. Pro snímky, které jsou dostatečně velké (rozlišení PAL a větší) doporučuji použít detekci významných bodů. Rychlost a kvalita zarovnání pomocí významných bodů je závislá na množství detekovaných bodů. Při testování se mi osvědčilo pracovat s plným rozlišením vstupních obrázků a omezit množství detekovaných bodů na hodnotu od 2000 do 5000. Výsledné video můžeme uložit ve finál-

ním rozlišení. Pro malé nebo nekvalitní vstupní snímky doporučuji použít vyhledání vzoru. Metoda je pomalejší, ale pracuje přesněji. Doporučuji ji na případy, kdy významné body selhaly nebo pro dodatečnou jemnou stabilizaci již výsledného videa.

Při korekci jasu a barev jsem nejlepších vizuálních výsledků dosáhl při zatažené obloze. Pokud bylo daný den slunečno, šel jsem fotit až těsně po západu slunce, kdy do fotografované scény nedopadaly sluneční paprsky a nevytvářely ostré stíny. Pomocí aktualizace histogramu jsem odstranil problémy vlivu ročního období a nastavením masky jsem odstranil chybu způsobenou rozdílností oblohy.

Metodu přizpůsobení histogramu doporučuji na vstupní data, která nemají velkou dynamiku. Metoda poskytuje přesné výsledky, ale selhává u snímků, kde se dynamika scény hodně mění. Jako velmi úspěšnou ji hodnotím u situací, kdy je zapotřebí upravit slabé blikání, ať už v jasu anebo barvě. Osvědčila se mi při úpravě scény fotografované v noci s bleskem. Výsledné video nejevilo známky sebemenšího blikání.

Gama korekce je univerzálnější a je schopna upravit většinu nedostatků. Její výsledky jsou horší než výsledky metody přizpůsobení histogramu, ale výsledek není postižen posterizací. Při testování se vyskytoval problém se špatně upraveným barevným odstínem, ale tento problém odstranila aktivace akumulace (adaptace) histogramu.

Jako budoucí rozšíření by bylo dobré vytvořit grafické uživatelské rozhraní, kde se dají specifikovat základní úkony jako je jednoduché vytvoření masky významných bodů a zvolení kotvy pro stabilizaci snímků. Dále by bylo dobré přidat funkci změny kotvy pro lepší funkčnost u skokových změn a také možnost výřezu na více pozicích a plynulá změna pozice ve výsledném videu pro definované výřezy. Implementace grafického rozhraní by nemělo být problémem, protože jsem vytvoření grafického rozhraní předpokládal od prvopočátku.

Dále by bylo vhodné implementovat systém pro zprávu zarovnání snímků, která by ukládala transformace již zarovnaných snímků. Došlo by k podstatnému zrychlení opětovného zpracování. Ve spojení s grafickým uživatelským rozhraním by uživatel mohl změnit předvolby pro nezarovnané snímky a dorovnat je zvlášť. Pokud chceme experimentovat s různými korekcemi jasu a barev, musí být známy transformace mezi snímky. Export do videozáznamu zhoršuje kvalitu výsledku, protože jsou uloženy předchozí snímky na okrajích aktuálního snímku.

V neposlední řadě je možnost implementovat detekci a odstranění náhodných objektů ve výsledné videosekvenci. Jako příklad poslouží ruka stínící přímému slunečnímu světlu. Dále je možnost implementovat ořezávací a jiné filtry při ukládání videozáznamu.

Pro medicínské účely může být implementováno předzpracování dat aby byly lépe detekovány významné body pro defekty kůže a různých zraněních. Pozadí je v tomto případě jednolitě a různé defekty kůže mohou být nedostatečně významné pro detekci významných bodů. V průběhu hojení se také sledovaný objekt bude výrazně měnit, proto by bylo dobré zvážit jiné zarovnání než k prvnímu snímku.

Další možnost rozšíření se může věnovat zlepšení časového zrychlení a zpomalení. Momentálně je implementováno jednoduché vynechávání nepotřebných snímků nebo jejich duplicita. Jako další metody by mohli být implementovány průměr snímků odpovídajících jednomu výslednému snímku a volitelný poměr průměru snímků kontra vynechání snímků v rámci jednoho výsledného snímku.

## Kapitola 7

### Závěr

Tato diplomová práce seznámila s problematikou a úskalími vytváření časosběrného videa. Popsala také existující nástroje pro vytváření časosběrného videa. Práce se dále zabývá časosběrným videem snímaným bez použití stativu. Byl podrobně popsán a implementován princip zarovnání snímků pomocí významných bodů v obraze. Dále se práce podrobně věnuje metodám pro korekci rozdílného jasu a barevného zabarvení mezi snímky.

Výsledkem práce je konzolová aplikace, která jako vstup přijímá sérii obrázků nebo již existující videozáznam. Jako výstup je možnost volby opět série obrázků nebo videozáznam. Volbou parametrů aplikace jsou k dispozici nástroje zajišťující zarovnání snímků a korekce jasu a barev. U obou nástrojů jsem implementoval 2 metody pro jejich realizaci. V práci jsou popsány situace, ve kterých metody podávají dobré výsledky a ve kterých selhávají. Uživatel má také možnost zadat masky pro jednotlivé nástroje, které specifikují oblast zpracování a přispívají k přesnějším výsledkům.

Během vývoje jsem narazil na problémy, které byly potřeba vyřešit. Problémy se také vyskytovaly během postupného sběru testovacích dat a testování již implementovaných metod. Nasbíraná testovací data upozornila na předem nepředvídané problémy, které bylo potřeba vyřešit. Požadavek na rychlé zpracování u časosběrného videa není vyžadován, přesto jsem základní optimalizace prováděl již v průběhu implementace.

Testování jsem prováděl na minimálně 6 různých testovacích sadách, které byly snímány s prodlevou jednoho dne a více. Některé testovací sady jsem zkoušel vytvářet kvalitně, jiné jsem záměrně vytvářel s nedostatky, abych lépe mohl posoudit účinnost implementovaných nástrojů. Sběru dat a jejich následnému testování bylo věnováno hodně času a značné úsilí, aby byly zjištěny případné nedostatky. Nedostatky byly odstraněny a metody se staly použitelnější.

Jako první možnost dalšího vývoje se nabízí vytvoření grafického uživatelského rozhraní pro snazší ovládání. Grafické uživatelské rozhraní nabízí lepší interakci aplikace s uživatelem, které by se mohlo využít při doplňujícím nastavení. Dále by bylo vhodné implementovat paměť pro zarovnání snímků. Zarovnání by potom nemuselo být prováděno stále znovu, pokud chceme experimentovat s korekcemi jasu a barev. V neposlední řadě je zde možnost implementovat detekci a odstranění náhodných objektů ve výsledné videosekvenci, přidání ořezávacích a jiných filtrů při ukládání videozáznamu.



# Literatura

- [1] Bradski, G.; Kaehler, A.: *Learning OpenCV: computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008, ISBN 978-0-596-51613-0, 555 s.
- [2] Fischler, M. A.; Bolles, R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, June 1986: s. 381–395, ISSN 0001-0782, doi:10.1145/358669.358692.
- [3] Gonzalez, R. C.; Woods, R. E.: *Digital Image Processing*. Tom Robbins, 2001, ISBN 0-201-18075-8.
- [4] Herbert Bay, T. T.; Gool, L. V.: SURF: Speeded Up Robust Features. *Proceedings of the ninth European Conference on Computer Vision*, May 2006.
- [5] Herbert Bay, T. T., Andreas Ess; Gool, L. V.: Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 2008: s. 346–359, ISSN 1077-3142, doi:10.1016/j.cviu.2007.09.014.
- [6] Kršek Přemysl, P., doc. Ing.: Základy počítačové grafiky - Geometrické transformace ve 2D/3D. slajdy, 2004.
- [7] Wikipedia: Korelace. <http://cs.wikipedia.org/wiki/Korelace>, 2010.
- [8] Wikipedia: RANSAC. <http://en.wikipedia.org/wiki/RANSAC>, 2010.
- [9] Wikipedia: Gamma correction. [http://en.wikipedia.org/wiki/Gamma\\_correction](http://en.wikipedia.org/wiki/Gamma_correction), 2011.
- [10] Wikipedia: Time-lapse. [http://en.wikipedia.org/wiki/Time\\_lapse](http://en.wikipedia.org/wiki/Time_lapse), 2010.

## Příloha A

# Manuál k programu

Syntaxe: `./timelapse [param]`

Parametry:

- `-i jmeno_souboru`
- `—input jmeno_souboru`
  - vstup, muze byt:
    - 1 – video soubor
    - 2 – slozka (vstupem jsou vsechny obrazky v danne slozce)
    - 3 – soubor txt s nazvy obrazku (jeden obrazek na radek, absolutni i relativni cesta)
- `-o jmeno_souboru`
- `—output jmeno_souboru`
  - vystup (vychozi je `output.avi`), muze byt:
    - 1 – video soubor (pripona `.avi`)
    - 2 – skupina obrazku, format zapisu: `jmeno%dd.pripona`, kde `%dd` specifikuje, kolik mista ma byt vyhrazeno pro cislici souboru, kde `dd` znaci cislici. `%` musi byt nasledovano 2 cislicemi.
- `-r cislo`
- `—fps cislo`
  - pocet snimku za sekundu u vystupniho videa
- `-s rozliseni`
- `—size rozliseni`
  - nastavi rozliseni vystupu. Format je `SxV`
  - Mozno take zadat `Sx` a `xV` => zbytek se dopocita automaticky
  - Nacteny obrazek se prevzorkuje na pozadovany rozmer
  - a potom se provadi pripadne zarovnani a korekce.
- `-hq`

`—high-quality`  
 pri práci se pracuje s originalním obrazem a převzorkování proběhne až při ukládání snímku.

`—c`  
`—compare`  
 vytvoří na výstupu video s dvojnásobnou sítí, kde levou stranu tvoří vstupní nezpracovaný snímek a pravou stranu výsledný snímek

`—c2`  
`—compare2`  
 stejně jako `—c`, ale levou stranu tvoří vstupní nezpracovaný snímek s vyznačenými korespondencemi bodů nalezených při stabilizaci

`—c3`  
`—compare3`  
 stejně jako `—c`, ale levou stranu tvoří vstupní stabilizovaný snímek

`—ng`  
`—no-gui`  
 při běhu programu se nebude zobrazovat grafický výstup

`—speed rychlost`  
 rychlost video souboru. Prozatím implementováno jednoduše, uložením jednoho snímku vícekrát nebo preskakováním snímku

`—info`  
 vloží do každého frame informací o čísle frame, pokud se jedná o sérii obrázků, vloží navíc název souboru

`—date`  
 vloží do každého frame datum vytvoření souboru funguje jen pro obrázky

`—stab <parametr=hodnota[:parametr=hodnota:parametr:...]>`  
 stabilizuje jednotlivé obrázky vzhledem ke zvolenému obrázku (kotva). Pokud kotva nebyla zvolena, kotvou je první obrázek. Jako hodnota je zde posloupnost `<parametr=hodnota>` vzájemně oddělený známkou `<:>`. V případě, že parametr nepotřebuje specifikovat hodnotu, zadá se jen parametr — `<parametr>`. Vždy musí být specifikován minimálně jeden parametr.  
 Příklady: `—stab method=surf:zoom=1.2`  
`—stab method=correlation:mask=IMG_12.JPG:zoom=1.2:affine`  
 Možné parametry s hodnotami:

`method=metoda`  
 metoda, kterou se bude stabilizace snímku provádět. Na výběr je jedna z možností:  
     `surf`      zarovnání pomocí významných bodů generovaných algoritmem SURF  
     `corr`      zarovnání pomocí korelace části obrazu kotvy s aktuálním obrázkem. Metoda je přesnější, ale pomalejší.  
     `corr_good_pictures`  
         stejně jako `corr`, ale posun v porizovaných obrázcích je minimální  
`mask=jmeno_souboru`  
 maska pro vytváření významných bodů. Jedná se o obrázek, kde body, pro které maska nabývá hodnot větší než 127 budou ponechány, ostatní budou odstraněny  
`transform=transformace`  
 určuje, která transformace na být použita, možné hodnoty:  
     `affine`  
         afinní transformace (výchozí hodnota)  
     `perspective`  
         perspektivní transformace  
     `affine_perspective`  
         afinní transformace, pokud bylo nalezeno dostatek korespondencí, bude pomocí korespondencí použita perspektivní transformace  
`zoom=hodnota`  
 zvětšení výsledného videa, používá se pro odstranění černých okrajů. Hodnota 1.0 znamená 100% velikost. Pripustné hodnoty jsou od 0.5 do 2.0.  
`borders=cele_cislo`  
 Používá se pro odstranění černých okrajů. Jako základ se vezme předchozí snímek a do něj se vloží aktuální snímek. Černé okraje budou průhledné a zadaná hodnota udává, jak dlouhé má být povolené prodloužení okraje. Hodnota 0 znamená ostrý okraj a záporná hodnota vypne odstranění černých okrajů. Výchozí hodnota je 0. Při nekorektní zadání se použije hodnota 0.  
`max_points=cele_cislo`  
 Určuje, s kolika významnými body se má maximálně pracovat. Malá hodnota urychluje běh programu, ale zhoršuje výsledek. Použitelné u metody SURF. Rozumná hodnota je mezi 2000 až 5000.  
`no_skip`  
 snímek bude použit i přes to, že není možné snímek zarovnat. Výchozí nastavení snímek přeskočí.

`-deflick <parametr=hodnota[:parametr=hodnota:parametr:...]>`  
 upravuje svetelnost a barevnost snimku vzhledem ke zvolenému snimku (kotva). Pokud kotva nebyla zvolena, kotvou je první obrázek. Jako hodnota je zde posloupnost `<parametr=hodnota>` vzájemně oddělený známkem `<:>`. V případě, že parametr nepotřebuje specifikovat hodnotu, zadá se jen parametr – `<parametr>`. Vždy musí být specifikován minimálně jeden parametr.  
 Příklady: `-deflick method=hist_rgb`  
 Možné parametry s hodnotami:

`method=metoda`  
 metoda, kterou se bude provádět korekce. Na výběr je jedna z možností:

- `hist_rgb`  
 korekce se provádí v barevném modelu RGB pomocí přizpůsobení histogramu vzhledem ke kotvě
- `hist_rgb_similar`  
 stejně jako `hist_rgb`, ale změna se aplikuje jen na podobné obrázky, při velkém rozdílu histogramu se korekce neprovede
- `gamma`  
 provede gamma korekci vedoucí k přizpůsobení střední složky jednotlivých RGB histogramů.
- `gamma2`  
 provede gamma korekci vedoucí k přizpůsobení střední složky jednotlivých RGB histogramů s přizpůsobením rozsahu do mezí minimální a maximální hodnoty referenčního snímku
- `rgb_gamma`  
 pokud jsou histogramy obrázku podobné, provede korekci metodou `hist_rgb`, jinak použije metodu `gamma`.

`accumulate`  
 místo pevného histogramu z kotevního snímku se kotevní histogram vypočítá následovně:  

$$0.75 * \text{puvodni\_histogram} + 0.25 * \text{aktualni\_histogram}$$

`accumulate=hodnota`  
 výpočet kotevního histogramu se vypočítá následovně:  

$$(1.0 - \text{hodnota}) * \text{puvodni\_histogram} + \text{hodnota} * \text{aktualni\_histogram}$$
 rozsah parametru: 0.0 až 1.0

`mask=jmeno_souboru`  
 maska pro vymezení oblasti, se kterou se bude pracovat při korekci. Oblast, pro kterou maska nabyvá hodnot větší než 127 budou ponechány, ostatní budou zanedbány



## **Příloha B**

### **Obsah DVD**

DVD obsahující elektronickou verzi technické zprávy, spustitelný program včetně zdrojových kódů, testovacích sad obrázků a ukázkových videí.