

**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ**

Katedra radioelektroniky



**Metody rekonstrukce archivních
videozáznamů**

Červen 2014

Autor práce: Bc. Marek Štych
Vedoucí práce: Ing. Karel Fliegel

Čestné prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne:

.....

podpis studenta

Poděkování

Děkuji panu Ing. Karlu Fliegelovi za obětavé vedení mé diplomové práce, za podnětné připomínky a rady ke zpracovávané problematice.

Zadání:

Seznamte s problematikou digitalizace archivních videozáznamů. Podejte přehled typických obrazových artefaktů, které se vyskytují v digitalizovaném obraze. Popište metody pro detekci a potlačení těchto artefaktů, vybrané metody implementujte a ověřte jejich účinnost. Zaměřte se také na problematiku objektivního a subjektivního vyhodnocení diferencí mezi projekcí filmového a digitálního obrazu.

Anotace

Cílem diplomové práce je nalezení vhodných metod pro detekci a potlačení obrazových artefaktů, které se vyskytují v digitalizovaném obraze archivních filmů. V textu jsou uvedeny typické artefakty, které se mohou ve filmu vyskytovat. Současně jsou popisovány metody pro jejich detekci a potlačení. Vybrané metody jsou poté implementovány a prakticky ověřeny.

Stěžejní část této práce tvoří porovnání účinnosti vybraných metod a jejich optimalizace. Po zvolení nejefektivnějších metod pro odstranění rušivých artefaktů přichází na řadu jejich společná aplikace na poškozené filmové sekvence a subjektivní testy kvality.

Summary

The thesis is focused on finding a suitable methods for detection and correction of disturbing artifacts in digitalized archive movies. The work includes descriptions of typical disturbing artifacts and several methods of degradation correction for each type. Some of these methods are then implemented and tested on corrupted movie sequences.

Principal part of thesis is comparison of efficiency of implemented methods and their optimization. The most effective methods are implemented on corrupted movie sequence and their result is evaluated by subjective quality tests.

Obsah

1. Úvod.....	1
2. Typy vad a jejich rozdělení.....	2
2.1 Skvrnová chyba.....	2
2.2 Vada poškrábáním.....	2
2.3 Šum v obraze.....	3
2.4 Blikavý šum.....	3
2.5 Degradace barev.....	4
2.6 Nestabilita obrazu.....	5
2.7 Rozdělení chyb podle příčiny vzniku a média, na kterém vznikly.....	5
2.8 Rozdělení chyb podle postupu při odstraňování.....	7
3. Detekce a odstranění skvrnové chyby.....	8
3.1 Úvod.....	8
3.2 Detekce skvrn.....	8
3.2.1 Detektory skvrnové chyby.....	9
3.2.2 Porovnání výsledků testů.....	11
3.3 Odstranění skvrn.....	12
3.4 Vyhodnocení.....	13
4. Vada blikavého šumu.....	14
4.1 Automatická metoda na odstranění blikavého šumu.....	14
4.2 Odstranění blikavého šumu pomocí ekvalizace histogramu.....	17
5. Degradace barev.....	19
6. Detekce a odstranění vady poškrábáním.....	22
6.1 Úvod.....	22
6.2 Detekce vady poškrábáním.....	23
6.2.1 Detekce třídičem textur a tvarovým filtrem.....	23
6.2.2 Detekce mediánovým filtrem a Houghovou transformací.....	24
6.3 Odstranění škrábnutí.....	27
6.4 Vyhodnocení.....	28
7. Šum.....	29
7.1 Úvod.....	29
7.2 Potlačení šumu pomocí lineárních metod.....	29
7.3. Potlačení šumu pomocí nelineární metody.....	30
7.3.1 Jádrové funkce.....	31
7.3.2 Prostorově variantní transformace.....	32
7.3.3 Využití třetí dimenze v transformaci.....	33
7.3.4 Redukce šumu pomocí jádrování.....	34
7.3.5 Vyhodnocení redukce šumu po použití jádrové funkce.....	34
8. Korekce nestability obrazu.....	36
9. Digitální vytváření vad filmu pro potřeby objektivního testování.....	39
9.1 Úvod.....	39
9.2 Vytváření jednotlivých vad.....	39
9.3 Aplikování vad na filmovou sekvenci.....	40
9.4 Výsledky aplikování vad na nepoškozenou filmovou sekvenci.....	41
10. Subjektivní testy.....	42
10.1 Subjektivní testy pro porovnání digitální a analogové filmové projekce v současné praxi.....	43
10.1.1 Test celkové kvality s dvojitém podnětem.....	43
10.1.2 Test párového porovnání.....	45
11. Objektivní testy.....	46

11.1 MSE (Mean Square Error).....	47
11.2 PSNR (Peak Signal to Noise Ratio).....	47
11.3 Metoda využívající hodnocení narušení struktur v obraze	48
12. Implementace.....	50
12.1 Implementace detekce a odstranění skvrnové chyby.....	50
12.1.1 Implementace detekce skvrnové chyby pomocí algoritmu SDIa a odstranění chyby pomocí algoritmu s přesahem.....	50
12.1.2 Implementace detekce skvrnové chyby pomocí algoritmu sROD.....	53
12.1.3 Implementace detekce skvrnové chyby pomocí adaptivního algoritmu.....	53
12.1.4 Implementace odstranění skvrnové vady pomocí algoritmu s přesahem....	55
12.1.5 Porovnání jednotlivých typů detekce.....	55
12.2 Implementace korekce nestability obrazu pomocí globálního vektoru pohybu.	62
12.3 Implementace korekce blikavého šumu pomocí ekvalizace histogramu.....	64
12.4 Implementace detekce vady škrábnutí mediánovým filtrem a Houghovou transformací.....	68
13. Vyhodnocení subjektivních testů.....	74
13.1 Test vad s dvojitým podnětem.....	74
13.2 Test párového porovnání.....	75
14. Vyhodnocení objektivních testů.....	76
15. Závěr.....	78
16. Seznam použitých videosekvencí.....	79
17. Přehled implementací a jejich parametry.....	80
18. Použité zdroje.....	82
19. Přílohy.....	84
Příloha č. 1: Formulář pro subjektivní Test párového porovnání.....	84
Příloha č. 2: Formulář pro subjektivní Test vad s dvojitým podnětem.....	85
Příloha č. 3: Výsledky Testů vad s dvojitým podnětem:.....	86
Příloha č. 4: Výsledky Testů párového porovnání:.....	87
Příloha č. 5: Obsah přiloženého DVD:.....	88

Přehled použitých zkratk

DCT	Discrete Cosine Transform (Diskrétní kosínová transformace)
DWT	Discrete Wavelet Transform (Diskrétní vlnková transformace)
FIR	Finite Impulse Response (Systém s konečnou odezvou)
HEQ	Histogram Equalization (Ekvalizace histogramu)
HVS	Human Visual System (Lidský zrakový systém)
MAD	Mean Absolute Difference (Střední hodnota absolutního rozdílu)
MSE	Mean Square Error (Střední kvadratická chyba)
MMF	Multistage median filter (Vícefázový mediánový filtr)
MMSE	Minimum Mean Square Error (Minimální střední kvadratická odchylka)
PSD	Power Spectral Density (Výkonová spektrální hustota)
PSNR	Peak Signal to Noise Ratio (Špičkový poměr signálu k šumu)
SNR	Signal to Noise Ratio (Poměr signálu k šumu)
SSIM	Structural SIMilarity index (index podobnosti dvou obrázků v závislosti na strukturální informaci)

1. Úvod

Narozdíl od literatury je film nerozlučně spjat s technikou, k jeho vzniku je nezbytné technické vybavení, jako jsou filmové kamery, střížny nebo filmové laboratoře. Prezentace díla je také technicky podmíněna, hotový film ožívá jen díky promítačce. Technologický proces výroby filmu proto vždy směřoval k výrobě filmové kopie, která by poskytovala tu nejlepší možnou prezentaci hotového díla. V 19. století tuto nejlepší prezentaci představoval kinematograf. S rozvojem technologie přicházely ve 20. století další formáty jako televizní vysílání, videozáznam atd. Projekce z filmového pásu byla však stále nejkvalitnější. Změna přichází až s vývojem digitálních projektorů s vysokým rozlišením. Tyto projektory nejprve dosáhly a poté i výrazně překonaly kvalitu filmových projektorů. Nastala tedy digitalizace, kina začala po celém světě přecházet na digitální projektory. V současnosti je velká většina filmů produkována na filmových i digitálních kopiích. Postupem času ale bude film jako distribuční médium pomalu zanikat. To znamená, že pokud filmové archivy chtějí dále uchovávat kulturní dědictví, kterým filmy v nich uskladněné jsou, musí podstoupit digitalizaci. Přejato z literatury [17].

V srpnu 2006 vydala Evropská komise doporučení týkající se digitalizace kulturního filmového materiálu a jeho uchování. V roce 2009 vznikla Národní strategie digitalizace kin v České republice. Tato strategie navrhuje mimo jiné i podpoření filmových kin z veřejných zdrojů, aby se využil jeden z hlavních přínosů digitalizace kin – rozšíření kvalitní programové nabídky. Tato nabídka by právě měla obsahovat i klasická díla z našich archivů od počátku české kinematografie v roce 1898 do roku 1993 [17].

Cílem digitální restaurace filmového díla je pořízení identické kopie filmového díla v podobě původních filmových materiálů. Je velice důležité zabezpečit proces převodu filmu do digitální podoby tak, aby nemohlo dojít k zásahu do původního díla, aby nebyly jednotlivé složky nijak měněny, nesmí dojít ke změně charakteru díla. Zjednodušeně řečeno, musí se jednat o stále totéž dílo a divák by měl mít z digitálního záznamu stejný prožitek jako při sledování originální filmové stopy. Digitální restauraci filmového díla je možné obecně rozdělit do několika výrobních etap:

- 1) příprava zdrojových filmových materiálů
- 2) scanování, respektive přepis zdrojového materiálu obrazu a zvuku do digitální podoby
- 3) digitální restaurování obrazu a zvuku
- 4) uchování restaurované digitální kopie
- 5) výroba digitálních masterů pro jednotlivé druhy šíření

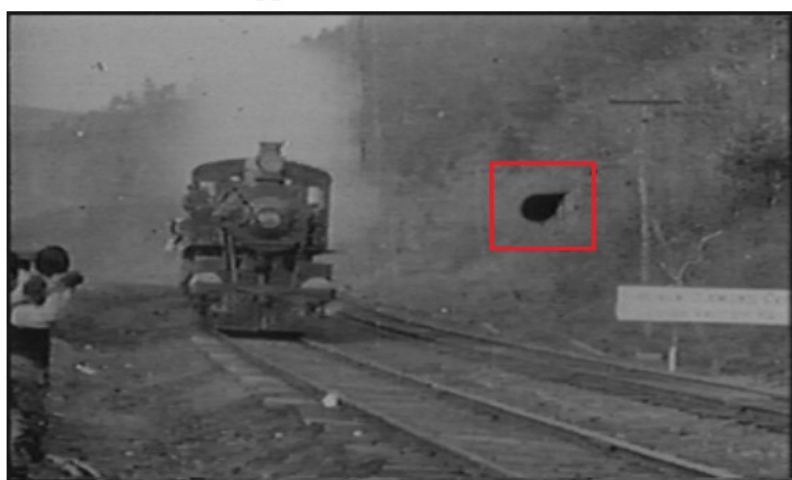
Tato práce se bude věnovat digitální restauraci obrazu. Po nasnímání nastává etapa postprodukčního zpracování získaných dat, která je velice důležitá pro výslednou kvalitu obrazu. Materiál je nejprve podrobně analyzován, následně je podroben jak automatické, tak i ručně prováděné rekonstrukci. Ta vede k odstranění velkého množství vad, které by zážitek ze sledování filmu mohly drasticky ovlivnit. Jedná se o vady způsobené prachem, škrábanci či jiným mechanickým poškozením. Také je zapotřebí doplnit chybějící části obrazu, nebo dokonce celé snímky. Musí se potlačit vertikální i horizontální nestability a šum. Aby mohl být obraz považován za platný, je také zapotřebí provést tzv. barevné korekce. Nejedná se o žádný kreativní zásah do díla, jde

pouze o korekci tonálního a barevného podání obrazu dle platného vzoru pro daný film. Výsledkem odstranění těchto nedostatků, kterým se bude tato práce věnovat, by měl být vyčištěný digitální otisk původního obrazového materiálu.

2. Typy vad a jejich rozdělení

2.1 Skvrnová chyba

Jednou z běžných vad starých filmů jsou skvrnové chyby, v anglické literatuře označované jako „blotch“ chyby. Tyto skvrny vznikají obvykle kvůli ztrátě povrchu na originálním filmu nebo kvůli elektrostatickým přilnavým materiálům, jako jsou prachové částice, vlasy atd., které nemohly být odstraněny fyzicky z filmu před digitalizací. V digitálním záznamu vypadají jako černé nebo bílé skupiny pixelů. Tyto vadné skupiny pixelů mají různou velikost, tvar a pozici. Klasický příklad skvrnové chyby je vidět na obrázku 2.1.



obrázek 2.1: Ukázka skvrnové vady ve filmu

2.2 Vada poškrábáním

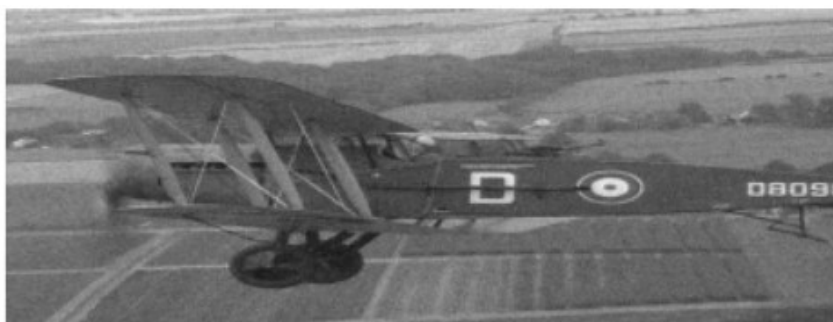
Vada poškrábáním je častým defektem starých filmů. Je způsobena odřením filmového materiálu během transportu nebo přehrávání. Tato poškození se projevují jako vertikální škrábance nebo skvrny v obrazu se šířkou 3 až 5 pixelů. Z pohledu zpracování filmu můžeme tyto škrábance definovat jako několik pixelů rozlehlá část obrazu, kde byla úplně ztracena hodnota intenzity. Příklad takovéto vady je vidět na obrázku 2.2.



obrázek 2.2: Ukázka vady poškrábání ve filmu „Star“ [7]

2.3 Šum v obraze

Abychom mohli pochopit, kde vzniká a jak se projevuje šum při procesu restaurace analogové sekvence, je třeba si nejprve uvědomit fakt, že kopie obrazu vznikají pomocí zařízení, které skenuje celuloidový film, upravuje ho a zapíše jej na digitální nebo analogové médium. (Dále budu toto zařízení označovat anglickým názvem Telecine [21], protože v českém jazyce tento výraz nemá alternativu). Bohužel teleciny, které byly používány dříve, měly poměrně špatné šumové charakteristiky, které se na kopii poté projeví. Často je právě taková kopie, vytvořená jedním ze starých telecinů, jediným záznamem filmu, který zbývá. Pro představu, jak se tento šum projeví na filmu, je uveden obrázek 2.3. V části (a) je vidět obrázek z kopie filmu a v části (b) je vidět obrázek po odstranění šumu (a).



(a)

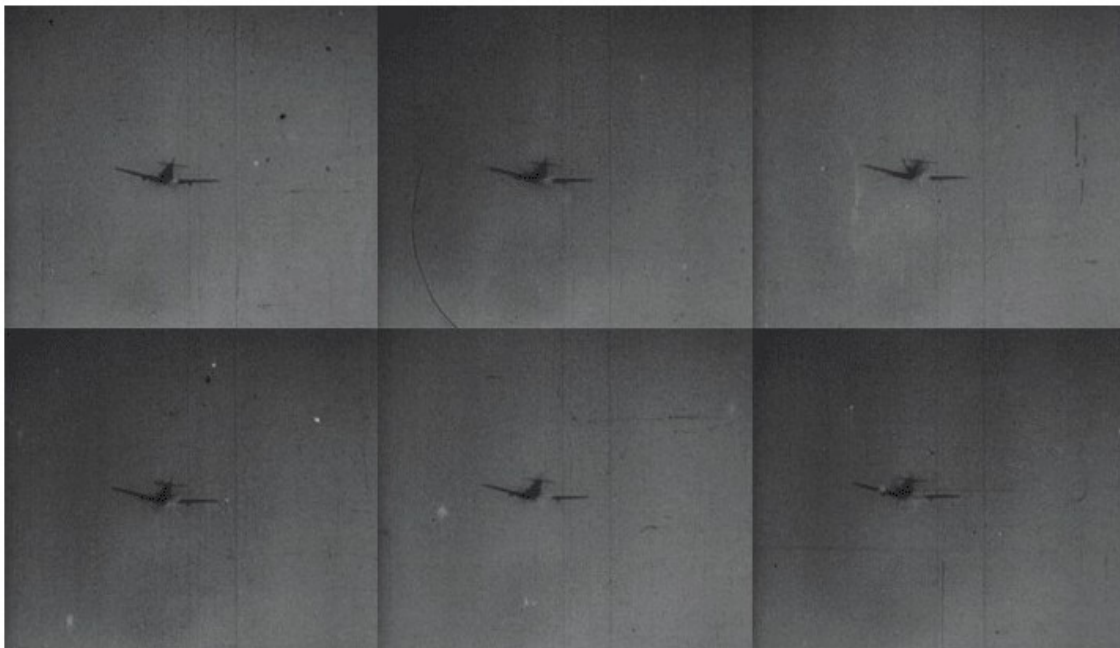


(b)

obrázek 2.3: Záznam z archivů BBC, (a) obrázek obsahující šum
(b) obrázek po odstranění šumu [3]

2.4 Blikavý šum

Blikavý šum představuje náhodné výkyvy v intenzitě jasu na určité ploše obrázku. V archivních filmech je to velice častý artefakt. Hlavní příčinou vzniku blikavého šumu je nekonzistentní filmová expozice při pořizování snímku. Dalšími příčinami vzniku blikavého šumu mohou být chyby při zpracování filmu, stárnutí, několikanásobné kopírování, plíseň a prach. Tato vada je závažná, protože může zakrývat důležité detaily filmu, zároveň je snadno viditelná a může působit velmi nepříjemně pro diváka. Ukázka takové vady je na obrázku 2.4.



obrázek 2.4: Sekvence starého filmu z druhé světové války obsahující intenzity flicker

2.5 Degradace barev

Film se skládá z barevných vrstev, jejichž kombinací vzniká obraz. S postupem času a stárnutím filmu se tyto barvy vytrácejí. Důležité je podotknout, že se vytrácejí nerovnoměrně. To znamená, že se obraz nejenom vytrácí, ale zároveň se i barevně mění. Co situaci ještě zhoršuje, je fakt, že často není k dispozici reference, podle které bychom mohli stanovit, jak skutečně barvy měly vypadat. Typická barevná degradace je vidět na obrázku 2.5, kde je většina modré informace a část zelené informace ztracena.



obrázek 2.5: Ukázka degradace barev [4]

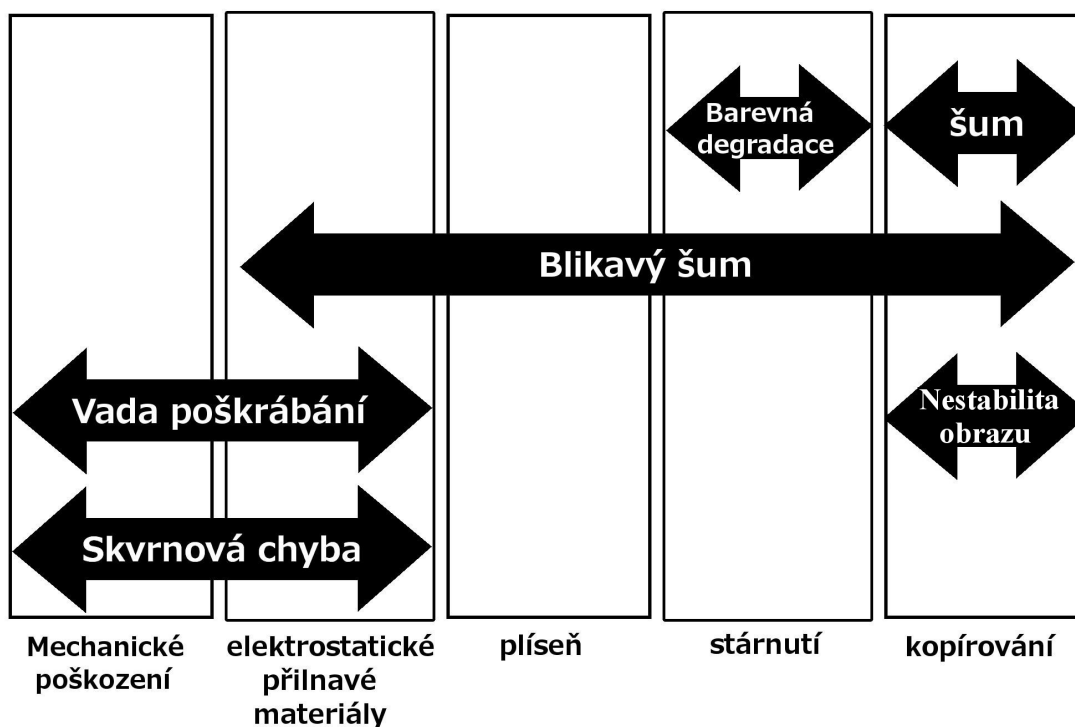
2.6 Nestabilita obrazu

Nestabilita obrazu je vada, která vzniká většinou přímo při natáčení filmu, v některých případech vzniká také při kopírování filmu pomocí telecinu. U starších kamer dochází k této vadě i v případě správného upevnění kamery a to z důvodů mechanické tolerance u technických parametrů kamery. Tato vada se projevuje jako nechtěný globální pohyb obrazu. Je to velice rušivá vada pro pozorovatele a navíc komplikuje odstranění ostatních vad, které se na filmové sekvenci mohou nacházet. Proto se ve většině případů odstraňuje jako první.

2.7 Rozdělení chyb podle příčiny vzniku a média, na kterém vznikly

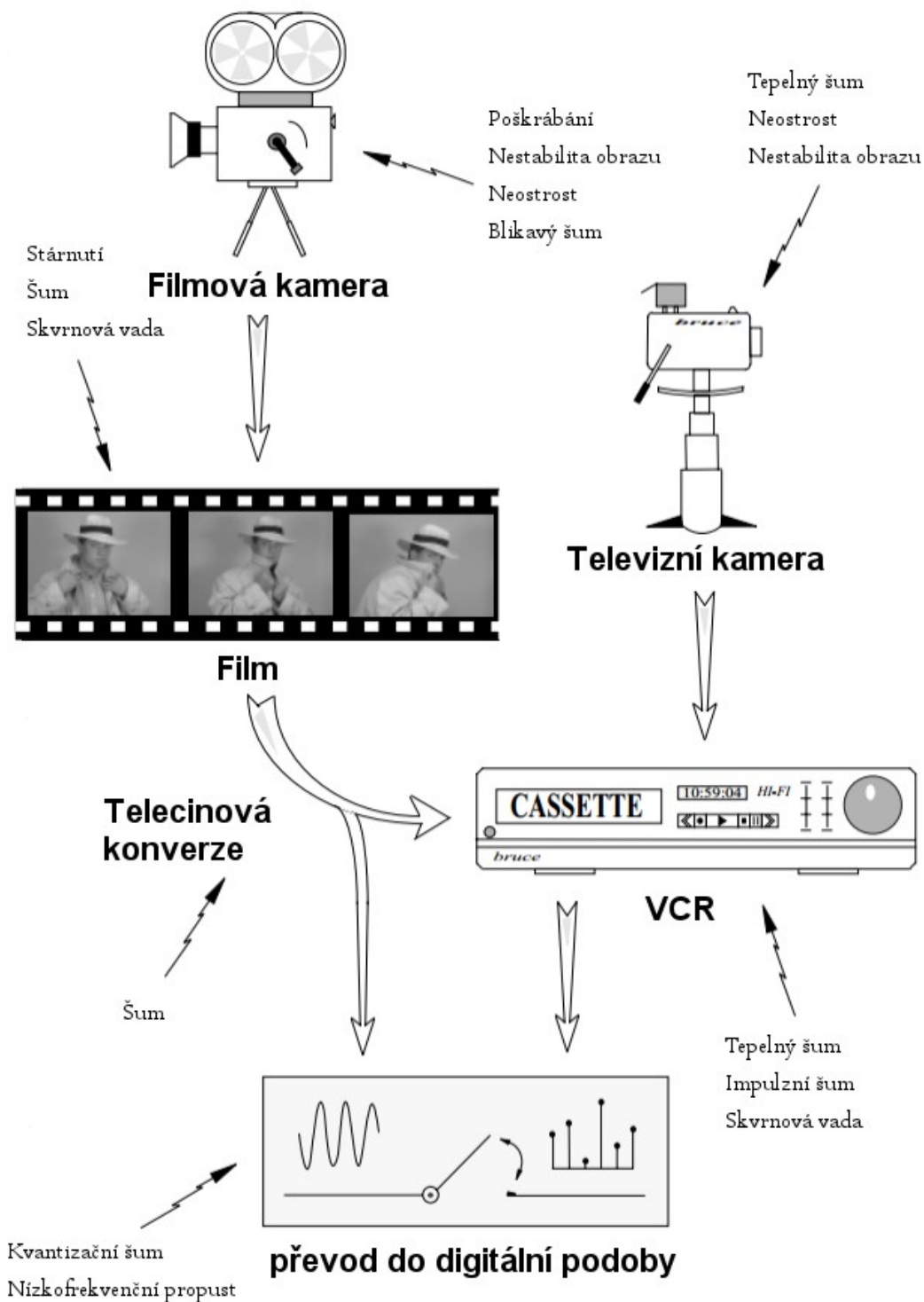
Vad je v archivních filmech poměrně velké množství. Každá z těchto vad je způsobena určitými faktory. Jedná se především o mechanické poškození, elektrostatické přilnavé materiály, plíseň, stárnutí a kopírování. Mechanickým poškozením je myšleno fyzické poškození filmu, které vzniká nejčastěji při kopírování, přehrávání a transportu. Přilnavými materiály jsou myšleny především elektrostatické materiály, jako například prach či vlasy, které přilnou k povrchu filmu a nedají se mechanicky odstranit, aniž bychom trvale nepoškodili film. Plíseň se na filmu vytváří především tehdy, když není správně skladován, tzn. není dodržena správná skladovací teplota, vlhkost vzduchu a tma. Stárnutím je myšleno přirozené stárnutí filmu bez jakéhokoliv zásahu člověka. Posledním z hlavních faktorů pro vznik chyby je kopírování. Je zapotřebí si uvědomit, že velké množství starých filmů bylo dříve kopírováno na nekvalitních zařízeních, která vnášela chyby do kopie filmu. V mnoha případech se bohužel stalo, že takto vytvořená kopie je jediným materiálem, který nám zbyl, originál je ztracen, popřípadě zničen, nebo se nachází v horším stavu než takto vytvořená kopie.

Na grafu 2.1 jsou zobrazeny nejčastější chyby v archivních filmech, viz kapitola 2, 2.1-2.6 a příčiny jejich vzniku.



graf 2.1: Nejčastější chyby a jejich příčiny

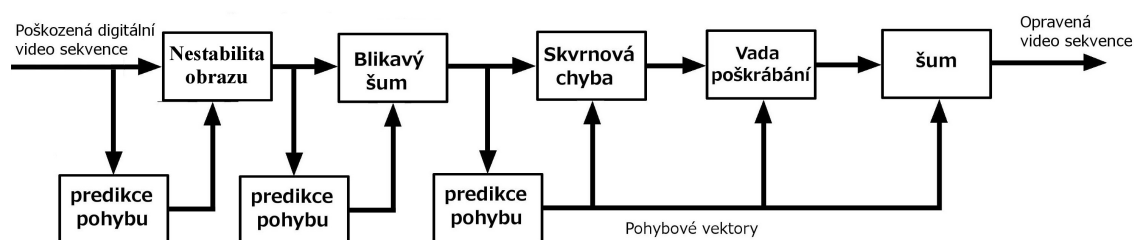
Pro lepší přehled vad filmů je uveden obrázek 2.6. Na tomto obrázku je vidět životní cyklus filmu, od jeho vzniku natočením analogovou kamerou, přes jeho skladování v podobě filmové pásky, až po jeho konverzi do digitální podoby pomocí telecinu. V každé této životní etapě filmu je uvedeno, jaké chyby mohou vznikat.



obrázek 2.6: Vznik analogového filmu a jeho převod do digitálu spolu s vadami, které vznikají v každé fázi [3]

2.8 Rozdělení chyb podle postupu při odstraňování

Při procesu odstraňování vad, které byly uvedeny v kapitole 2.1-2.6, je zapotřebí si uvědomit, že film nemusí být poškozen jen jednou vadou, ale může být poškozen hned několika vadami najednou, v nejhrošším případě všemi. Vytvořit algoritmus, který by odstranil všechny vady, je sice možné, ale nepraktické. Proto byl zvolen modulární přístup, založený na programátorské metodě rozděl a panuj. Modulární přístup zvětšuje pravděpodobnost úspěchu při rekonstrukci videa a má mnoho výhod. Například je přirozené, že pokud se nám ve filmové sekvenci nenachází žádná skvrnová chyba, je nerozumné použít algoritmus pro její odstranění, protože by mohlo dojít k falešným detekcím a nežádanému zásahu do filmu. Pro odstranění všech vad se použijí jednotlivé algoritmy pro odstranění vad. Pořadí, ve kterém je doporučeno algoritmy použít, je vidět na obrázku 2.7.



obrázek 2.7: Postupné odstranění všech vad na filmu

První blok v obrázku 2.7 reprezentuje algoritmus pro odstranění nestability obrazu. Ten je použit prioritně v případech, ve kterých se nestabilita obrazu projevuje výrazně. Výrazná nestabilita obrazu by totiž ovlivnila následující algoritmy. Druhým blokem je algoritmus pro odstranění blikavého šumu. Ten odstraní rušivé výkyvy v intenzitě jasu. Tato korekce je důležitá pro správnou predikci pohybu, která je použita při odstraňování ostatních vad. V případech, kdy nestabilita obrazu není výrazná a naopak vada blikavého šumu je snadno pozorovatelná, doporučuji použít algoritmus pro korekci blikavého šumu jako první a až po něm korekci nestability obrazu. Následujícími bloky jsou odstranění skvrnové chyby a vady poškrábání. Tyto dva bloky je možné aplikovat i v obráceném pořadí, než je uvedeno v obrázku 2.7. Záleží na konkrétních použitých algoritmech, ale pokud jsou použity kvalitní algoritmy, nemělo by na pořadí těchto bloků záležet (pro algoritmy, které prezentuji v této práci na tomto pořadí skutečně nezáleží). Poslední použitý blok v tomto schématu slouží k odstranění šumu v obraze. Zatímco skvrnové chyby a vady poškrábání jsou lokálním poškozením filmu, šum je po celé ploše a může ovlivňovat všechny pixely. Protože se tedy jedná o algoritmus, který pracuje po celé ploše obrazu, mohly by následující metody být ovlivněny artefakty, které tato šumová redukce vnese do obrazu. Abychom tomuto předešli, je odstranění šumu provedeno až na závěr celého procesu.

Predikce pohybu je u prvních dvou bloků udělána samostatně. Je tomu hlavně proto, že algoritmus pro odstranění nestability obrazu a algoritmus pro odstranění blikavého šumu by měly výrazně zlepšit detekci pohybu pro následující bloky. Proto je tato detekce pro následující bloky udělána znovu a přesněji.

Pro některé algoritmy zpracovávající odstranění blikavého šumu je nutno nejprve odstranit šum. Takovýto algoritmus je uveden v kapitole 4.1. (Naopak algoritmus v kapitole 4.2 je aplikován podle obrázku 2.7). V takovém případě je tedy nutno aplikovat metodu pro odstranění šumu před metodou pro odstranění blikavého šumu.

3. Detekce a odstranění skvrnové chyby

3.1 Úvod

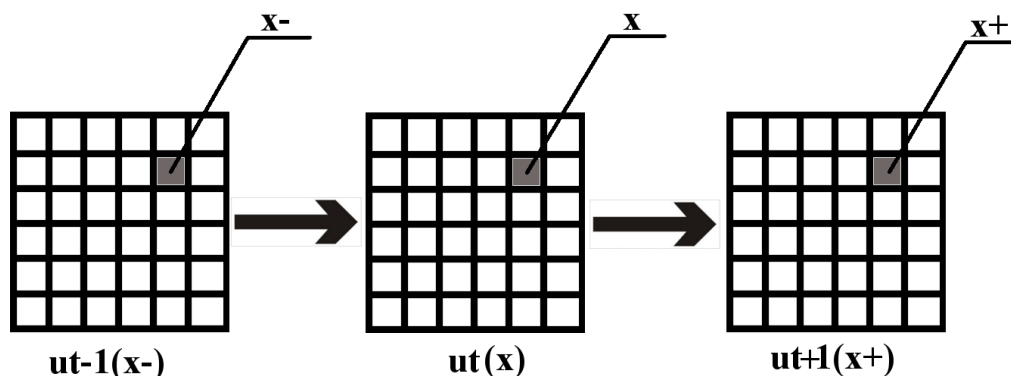
Po digitalizaci analogového filmu je třeba odstranit množství vad, které způsobují degradaci kvality obrazu. Tato část práce pojednává o detekování a odstraňování chyb typu "blotch" neboli skvrnové chyby v digitálním záznamu. Tyto skvrny vznikají obvykle kvůli ztrátě povrchu na originálním filmu nebo kvůli elektrostatickým přilnavým materiálům, jako jsou prach, vlasy atd., které nemohly být odstraněny fyzicky z filmu před digitalizací. V digitálním záznamu vypadají jako černé nebo bílé skupiny pixelů. Tyto vadné skupiny pixelů mají různou velikost, tvar a pozici.

Jak bude popsáno dále, největší problém je v nastavení kritérií pro detekování skvrn, protože taková kritéria nemohou být jednotná pro celý obraz. Jejich nastavení závisí na tom, jestli námi sledovaný pixel leží na rychle se měnící heterogenní části obrazu, nebo zda se jedná o pixel ležící na homogenní části obrazu, která se mění pomalu. Navíc, pokud se chceme vyhnout takzvaným falešným poplachům, což znamená označení platných pixelů za poškozené, vyžaduje většina metod používaných pro detekci skvrnové chyby velice jemné nastavování několika různých parametrů.

Klasický postup pro odstranění skvrnové chyby se skládá ze tří částí: "motion estimation" neboli výpočet odhadu pohybu v obraze, detekce skvrn a odstranění skvrn. Každá část může být realizována několika algoritmy. V této práci se budu věnovat především detekci skvrn a obnovení obrazu z hlediska porovnání jednotlivých metod a výběru té nejhodnější pro restauraci filmů v nejlepší možné kvalitě. Ke kompenzaci pohybu v obraze je zapotřebí pouze uvést, že je použita metoda založená na blokovém přístupu. Tato metoda, pokud je správně použita, nabízí dobrý poměr mezi výpočetní rychlostí a kvalitou, čímž nám umožňuje dosáhnout kvalitních výsledků v procesu detekování skvrn.

3.2 Detekce skvrn

Pro popsání algoritmů na detekci skvrn je nejprve třeba vysvětlit a zadefinovat některé pojmy. Film je označený jako $u = (u_t(x))$, kde t reprezentuje čas, u_t značí obraz (frame) v čase t , x vyjadřuje bod v obraze určený prostorovými souřadnicemi. Pro každé x aktuálního obrazu známe x^- , což je bod odpovídající bodu x v předchozím obraze, tedy může to být bod se stejnými souřadnicemi, pokud v obraze nebyl žádný pohyb. Pokud pohyb v obraze byl, souřadnice bodu x^- může být jiná než souřadnice bodu x tak, aby si body skutečně odpovídaly z hlediska nesené informace. Stejně tak x^+ je bod, který odpovídá bodu x v následujícím obraze u_{t+1} . Pokud máme zadefinovány tyto pojmy, můžeme určit difference d^- a d^+ jako: $d^-(x) = u_t(x) - u_{t-1}(x^-)$ a $d^+(x) = u_t(x) - u_{t+1}(x^+)$.



obrázek 3.1: Znáznornění odpovídajících si pixelů

3.2.1 Detektory skvrnové chyby

Dva klasické testy pro detekci skvrn jsou založeny na nastavování prahových hodnot pro d^- a d^+ . Tento práh je obvykle zvolen globálně prostorově i časově, tedy je stejný pro všechny obrázky, ze kterých se film skládá a v rámci jednoho obrázku stejný po celé jeho ploše. První z těchto klasických testů porovnává hodnotu prahu α s funkcí (3.1).

$$b(x) = \min(|d^-(x)|, |d^+(x)|) \cdot \text{sign}[d^-(x) \cdot d^+(x)] \quad (3.1)$$

Tento test se jmenuje SDIa [1]. Druhý přístup je postaven na statistických údajích spočítaných z přilehlých pixelů k x^+ a x^- . Tento přístup je znám tím, že s jeho pomocí jsme schopni dosáhnout dobrých výsledků, ovšem za cenu složitějšího nastavování tří prahových hodnot. Tento test byl nazván ROD. Jedna z jeho variant, která je nyní používána nejčastěji, se nazývá sROD. Při jejím použití získáváme hodnoty $(r1, r2, r3)$ a $(r4, r5, r6)$ jako reprezentanty stupně šedi ve třech pixelech přilehlých x^+ respektive x^- . Bod x je detekován jako vadný, pokud jsou splněny následující podmínky: $u(x) - \max(r1, \dots, r6) > S$ nebo $\min(r1, \dots, r6) - u(x) > S$, kde S je námi nastavená pozitivní konstanta.

Důležitým typem detekce skvrn je takzvaná adaptivní metoda [1]. Adaptivní v tomto případě znamená, že jednotlivé prahové hodnoty se přizpůsobují podle lokálních statistik v obraze, tedy liší se v prostoru. Zároveň jsou rozdíly v úrovni šedi sledovány v celém okolí testovaného bodu. Pro každý pixel x je cílem rozhodnout zda rozdíl mezi okolím x a okolím x^- dává smysl a pokud ne, tak zda se jedná o skvrnovou chybu nebo zda tato anomálie může být vysvětlena lokálními statistikami v obraze.

Řešení problému rozlišení vadného pixelu od správného je realizováno pomocí nulové hypotézy. Nechť je N_0 okolí bodu 0 a $|N_0|$ jeho velikost. Pro bod x v čase t , předpokládejme mřížku pixelů $u_t(x + N_0)$ a odpovídající pixely $u_{t-1}(x^- + N_0)$ v předchozím obrázku. Stejně tak pixely $u_{t+1}(x^+ + N_0)$ v následujícím obrázku. Nulová hypotéza deklaruje, že rozdíly $u_t(x + y) - u_{t-1}(x^- + y)$, kde y zahrnuje N_0 , jsou realizací nezávislých a identicky rozložených náhodných hodnot s normálním rozložením $N(0, \sigma^2)$. V naší metodě pro detekování skvrnové chyby je použit stejný předpoklad na rozdíl $u_t(x + y) - u_{t+1}(x^+ + y)$, kde y reprezentuje N_0 . Pokud je hypotéza H naplněna, potom součet (3.2) je realizací distribuční funkce $N(0, \sigma^2 / |N_0|)$. Hypotéza H může být tedy testována spočítáním $F_x(|D^+|)$ a $F_x(|D^-|)$, kde $F_x(\sigma)$ se spočítá jako (3.3).

$$D_x^- = \frac{1}{|N_0|} \sum_{y \in N_0} (u_t(x+y) - u_{t-1}(x^- + y)) \quad (3.2)$$

$$F_x(\sigma) = 1 - \frac{\sqrt{|N_0|}}{\sqrt{(2\pi\sigma_x)}} \int_{-\sigma}^{\sigma} e^{-\frac{|N_0|s^2}{2\sigma_x^2}} ds \quad (3.3)$$

Pokud jsou $F_x(|D^+|)$ a $F_x(|D^-|)$ příliš malé a zároveň pozorované rozdíly $|D^+|$ a $|D^-|$ jsou příliš velké, můžeme vyvodit, že tyto rozdíly by jen stěží mohly být vysvětleny nulovou hypotézou. V těchto případech je nulová hypotéza H zamítnuta, což znamená, že v bodě x se nachází skvrnová chyba. Tedy vše, co potřebujeme pro to, abychom odhalili skvrnovou chybu, je spočítat ASDI (adaptive spike detection index) a zvolit správně práh ϵ :

$$ASDI(x) := \max(F_x(|D_x^-|), F_x(|D_x^+|)) \cdot \text{sign}(D_x^- \cdot D_x^+) \quad (3.4)$$

Pro zadaný práh ϵ je bod x z obrázku u_t detekován jako postižený skvrnovou chybou pokud:

$$0 \leq ASDI(x)^- \leq \frac{\epsilon}{|\Omega|} \quad (3.5)$$

Za všimnutí stojí to, že funkce F_x závisí pouze na velikosti mřížky pixelů N_0 a na lokální standardní odchylce σ_x . V další sekci se budu věnovat tomu, jak je odchylka σ_x spočítána.

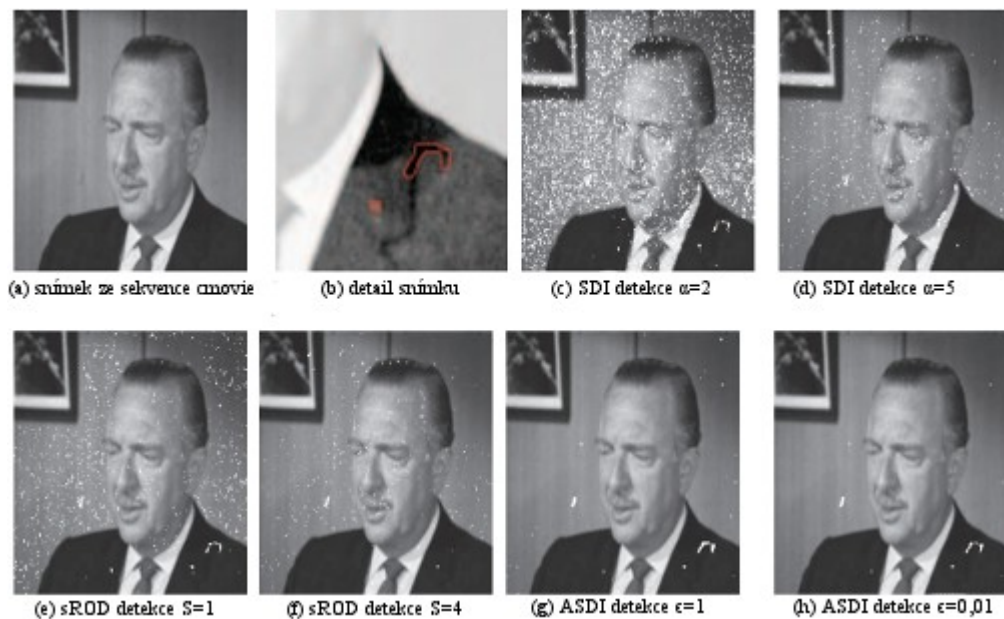
Pro námi daný práh ϵ nám tento přístup generuje rozdílné adaptivní prahy pro různé body x . Čím menší je hodnota σ_x , tím menší bude práh pro absolutní hodnoty rozdílu. Jinými slovy, málo kontrastní skvrna oproti pozadí má větší šanci být detekována než u neadaptivních algoritmů.

Při vyhledávání chyby je zapotřebí počítat s pohybem ve filmu. Odhad pohybu pro ASDI algoritmus hledání skvrn je naimplementován pomocí porovnávání bloků. Celý obrázek je rozdělen do pevných mřížek překrývajících se bloků o velikosti $(2f + 1)^2$. Každý blok zasahuje do mřížky sousedících bloků do hloubky f pixelů. Pro každý blok W v obrázku u_t je vyhledán nejlépe odpovídající blok v obrázku u_{t-1} a je označen jako W^- . Nejlépe odpovídající blok je hledán na všech blocích na ploše $(2s + 1)^2$. Nejlépe odpovídající blok je potom ten, který minimalizuje L^2 vzdálenost mezi bloky. Nejlépe odpovídající blok W^+ v obrázku u_{t+1} je spočítán stejně. Poté je použit test ASDI pro každý pixel x z W , standardní odchylka σ_x je odhadnuta empiricky jako standardní odchylka pixelového rozdílu mezi W^+ a W^- . To znamená, že standardní odchylka je stejná pro všechny body v W . Pokud bod x náleží několika blokům a aspoň v jednom z nich je detekován jako chybný, pak je automaticky označen za chybný i globálně.

3.2.2 Porovnání výsledků testů

Nyní se budu věnovat porovnání metod pro detekci skvrnové chyby. Jsou zde porovnány detektory ASDI, SDI a sROD na dvou sekvencích nazvaných *cmovie* a *super8*. Velikost bloků W byla zvolena 21×21 a přesah mezi jednotlivými bloky je nastaven na 10 pixelů. Obrázek 3.2 obsahuje jeden obrázek ze sekvence *cmovie* se dvěma málo kontrastními bloky, zářivým bodem a tmavým pruhem. Jak demonstruje obrázek 3.2(c) 3.2(e), tmavý pruh může být detekován pomocí SDI a sROD za použití velice nízkých prahů, ale pouze za cenu velkého množství falešných detekcí. Mimoto černá skvrna rychle zmizí z detekčních map, když je práh zvýšen, zatímco falešné detekce zůstanou (obrázek 3.2(d), 3.2(f)). Naproti tomu ASDI správně detekuje skvrnu při prahu $\epsilon = 1$, dokonce i při prahu $\epsilon = 0,01$, čímž limituje počet falešných detekcí, jak je vidět na obrázku 3.2(d), 3.2(f). Tento pokus demonstruje význam adaptivních prahů, jejichž výsledky jsou výrazně lepší než výsledky klasických testů.

Obrázek 3.3 demonstruje chování detektorů na obrázku ze sekvence *super8*. ASDI detekuje všechny skvrny v obrázku při prahu $\epsilon = 1$, konkrétně dvě skvrny na levé straně obrázku a skvrnu vespod spolu s několika chybnými detekcemi na ruce muže, kvůli pohybu. Pro porovnání, SDI a sROD vyvolávají spoustu falešných detekcí, hlavně v místech ovlivněných pohybem, při prahové hodnotě, která může sotva detekovat jednu ze tří přítomných skvrnových chyb.



obrázek 3.2: Detekce skvrnových chyb v sekvenci *cmovie*; (a) originální snímek; (b) detail snímku, který obsahuje tmavou skvrnovou chybu; (c) SDI detekce s prahem $\alpha=2$; (d) SDI detekce s prahem $\alpha=5$; (e) sROD detekce s prahem $S=1$; (f) sROD detekce s prahem $S=4$; (g) adaptivní metoda s prahem $\epsilon=1$; (h) adaptivní metoda s prahem $\epsilon=10^{-2}$ [1]

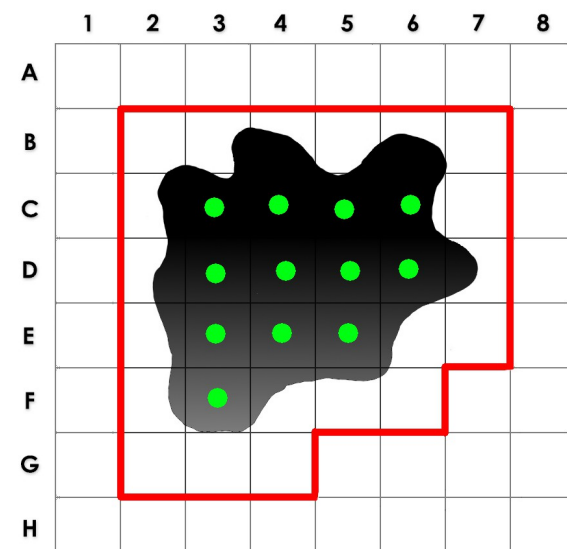


obrázek 3.3: Detekce skvrnových chyb v sekvenci *super8*; (a) originální snímek; (b) SDI detekce s prahem $\alpha=15$; (c) sROD detekce s prahem $S=15$ (d) adaptivní metoda s prahem $\epsilon=1$ [1]

3.3 Odstranění skvrn

Odstraňování skvrn může být považováno za problém chybějících dat v obrazu. Proto se k jejich odstranění používají interpolační algoritmy, které k restauraci chybějících dat používají přilehlá nepoškozená data. MMF (multistage median filter) je zřetězení operací mediánového filtru. Jedná se o heuristickou metodu pro interpolování chybějících dat.

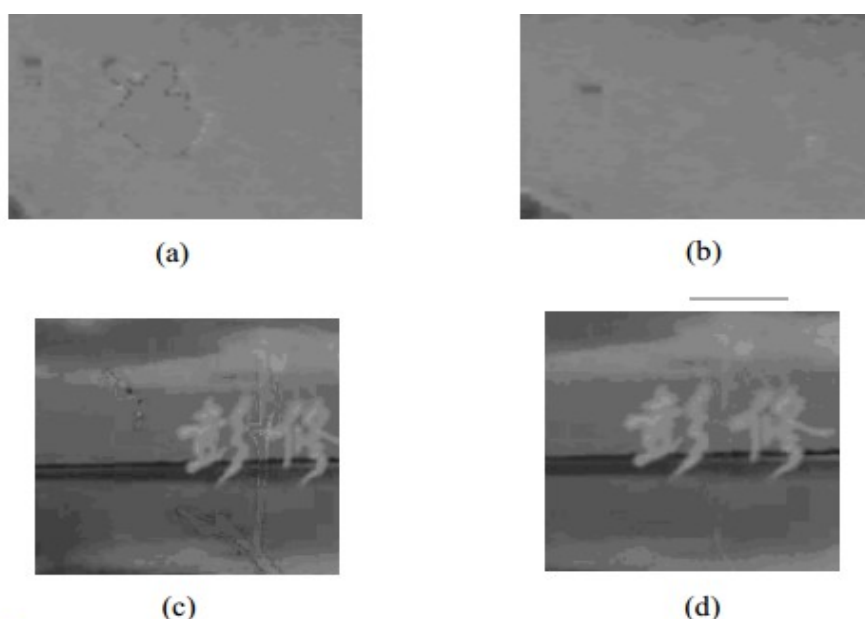
Standardní skvrnové chyby mají rozmazané okraje. Všechny metody pro nahrazení chybějících dat poznamenávají okraje skvrny, jak je vidět v obrázku 3.5(a), (c). Pro odstranění tohoto nedostatku byl navržen v práci [5] přesahující algoritmus pro obnovení chybějících dat (obrázek 3.4).



obrázek 3.4: Výběr přesahu pro odstranění skvrnové chyby. Zelené body představují pixely detekované jako poškozené, červený rámeček reprezentuje rozšířenou detekci skvrnové chyby

Tento algoritmus pracuje nejen s místem poškozeným skvrnovou chybou, ale i s jeho okolím. Nejprve tedy algoritmus nalezne pixely, který nám vyznačují místo se samotnou chybou; na obrázku 3.4 jsou zobrazeny zeleným bodem. Poté je tato detekce rozšířena o n pixelů tak, aby pojmla i přesahující okraj skvrnové chyby. Po provedení velkého počtu experimentů bylo zjištěno, že dostačující hodnota pro odstranění defektu na hranách je $n=1$. Tato maska je použita pro opravení skvrnové chyby. Ta je odstraněna, jak bylo uvedeno výše, MMF interpolačním algoritmem.

Výsledky testů MMF interpolačního algoritmu a přesahového algoritmu jsou zobrazeny na obrázku 3.5. Místo se skvrnou je zvětšeno, aby se výsledky daly lépe porovnávat. Obrázky 3.5(a), (c) jsou skvrny opravené pomocí MMF, zatímco obrázky 3.5(b), (d) jsou výsledky algoritmu s přesahem. Okrajový efekt, který je na obrázcích 3.5(a), (c) jasně vidět, se na obrázcích 3.5(b), (d) neobjevuje. Tato nová metoda obnovení míst na filmu poškozených skvrnovou chybou je tedy velkým přínosem pro celý proces restaurace filmů.



obrázek 3.5: Příklad odstranění blotch chyby (a), přiblížení skvrny opravené MMF (b), přiblížení skvrny opravené algoritmem s přesahem (c), opravený obrázek pomocí MMF (d), obrázek opravený pomocí algoritmu s přesahem [5]

3.4 Vyhodnocení

Skvrnové chyby jsou jedním z nejrušivějších elementů ve filmovém obraze. Pokud máme zajistit divákům filmu co možná nejlepší zážitek, je třeba tyto chyby odstranit takovým způsobem, aby obraz vypadal pokud možno stejně jako před poškozením. K tomu je zapotřebí využít co nejefektivnějšího algoritmu pro detekci těchto chyb. Jak je vidět v předchozích kapitolách, nejvhodnější je adaptivní vyhledávání, které nejenom chyby hledá s největší úspěšností, ale zároveň má malý počet falešných detekcí. Je také možné kombinovat tento algoritmus s jedním z klasických testů a použít oba. Pro odstranění skvrnové chyby, tedy nahrazení chybějících pixelů, byly dříve používány obyčejné mediánové filtry. Jak je ovšem vidět v sekci 3.3, použití mediánových filtrů s přesahem velmi zlepšil zrekonstruovaný obraz a odstraní některé nepříjemné artefakty. Pokud tedy budeme postupovat s pomocí těchto algoritmů, měli bychom v současnosti dosáhnout nejlepších možných výsledků pro odstranění skvrnové chyby.

4. Vada blikavého šumu

Flicker neboli vada blikavého šumu se projevuje jako nežádoucí výkyvy světlosti v obraze, které nemají vazbu na originální scénu. Tato vada je způsobena několika faktory: stárnutím filmu, nestabilním chemickým zpracováním, kopírováním filmu, neprecizní telecinovou konverzí atd. V posledních letech byla věnována této chybě velká pozornost a bylo vytvořeno několik metod na odstranění blikavého šumu. V této práci budou rozebrány dvě metody, jedna plně automatická [6], druhá částečně manuální [15]. Každá z těchto metod má své výhody a nevýhody. Automatická nepotřebuje obsluhu technika (s výjimkou nastavení prahů), ale její algoritmus má výrazně větší výpočetní složitost. Výsledná kvalita korekce blikavého šumu je přibližně stejná a každá může být preferována na jiné filmové sekvenci, proto tyto metody budou popsány obě.

4.1 Automatická metoda na odstranění blikavého šumu

Tato část práce se bude věnovat automatické metodě, která je založená na pohybových vektorech obrazu. Ty jsou spočítány pomocí tříkrokového algoritmu, který dokáže detekovat oblasti s pohybem velice kvalitně. Mimoto se v této vylepšené metodě setkáváme i s nastavitelným prahem citlivosti. Díky jeho správnému nastavení se můžeme vyhnout tomu, aby se opravovala i místa, která vadou blikavého šumu nejsou postižena.

Není možné vytvořit explicitní fyzický model pro každý mechanismus, který způsobuje blikavý šum. Většina těchto modelů je založena na vlastnosti blikavého šumu, konkrétně na tom, že se blikavý šum prostorově rozkládá plynule. Díky této vlastnosti Van Roosmalen [6] přišel s myšlenkou, jak využít šumový model reprezentovaný rovnicí (4.1).

$$Y_t(x, y) = \alpha(x, y)I_t(x, y) + \beta_t(x, y) + \sigma_t(x, y) \quad (4.1)$$

V této rovnici x, y reprezentují prostorové souřadnice a t reprezentuje pořadové číslo obrázku ve filmu. $Y_t(x, y)$ a $I_t(x, y)$ označují pozorované a vadou blikavého šumu postižené intenzity obrázku. $\alpha_t(x, y)$ a $\beta_t(x, y)$ jsou multiplikativní a aditivní parametry vady blikavého šumu. Šum, který je v obraze a nesouvisí s blikavým šumem, je označen $\sigma_t(x, y)$; tento parametr tedy obsahuje šumy, jako je například skvrnový šum, gausův šum atd.

Je nezbytné, aby se před samotným detekováním a odstraňováním blikavého šumu nejprve prošlo fází předzpracování a byly odstraněny všechny ostatní šumy označené jako $\sigma_t(x, y)$. Po odstranění těchto šumů můžeme rovnici 4.1 zjednodušit na rovnici 4.2. Pro odstranění chyby blikavého šumu je poté zapotřebí pouze správně odhadnout parametry α a β .

$$Y_t(x, y) = \alpha_t(x, y)I_t(x, y) + \beta_t(x, y) \quad (4.2)$$

Jak bylo řečeno výše, intenzita chyby blikavého šumu se prostorově rozprostírá plynule. Její parametry tedy mohou být považovány za konstantní v určitých oblastech. Obraz se rozdělí do několika stejně velikých čtvercových oblastí a uvnitř jednoho takového bloku $\Phi_{m,n}$ (kde m, n označují prostorové souřadnice levého horního rohu bloku) $\alpha_t(x,y)=\alpha_{m,n,t}$ a $\beta_t(x,y)=\beta_{m,n,t}$ pro všechna x,y ležící na $\Phi_{m,n}$, podle rovnice 4.2 tedy platí pro všechny pozice v $\Phi_{m,n}$:

$$E[Y_t(x, y)] = E[\alpha_{m,n,t} I_t(x, y) + \beta_{m,n,t}] = \alpha_{m,n,t} E[I_t(x, y)] + \beta_{m,n,t} \quad (4.3)$$

$$\sigma^2[Y_t(x, y)] = E[Y_t(x, y) - E[Y_t(x, y)]]^2 = \alpha_{m,n,t}^2 \sigma^2[I_t(x, y)] \quad (4.4)$$

E a σ reprezentují předpokládanou hodnotu a standardní odchylku. Z rovnic 4.3 a 4.4 můžeme vypočítat parametry blikavého šumu α a β , jak je vidět v rovnicích 4.5 a 4.6:

$$\alpha_{m,n,t} = \frac{\sigma[Y_t(x, y)]}{\sigma[I_t(x, y)]} \quad (4.5)$$

$$\beta_{m,n,t}(t) = E[Y_t(x, y, t)] - \alpha_{m,n,t}(t) E[I_t(x, y, t)] \quad (4.6)$$

Pro místa, ve kterých se v obrazu neděje žádný pohyb, by měly přibližně platit rovnice 4.7 a rovnice 4.8.

$$E[I_t(x, y)] \approx E[\hat{I}_{t-1}(x, y)] \quad (4.7)$$

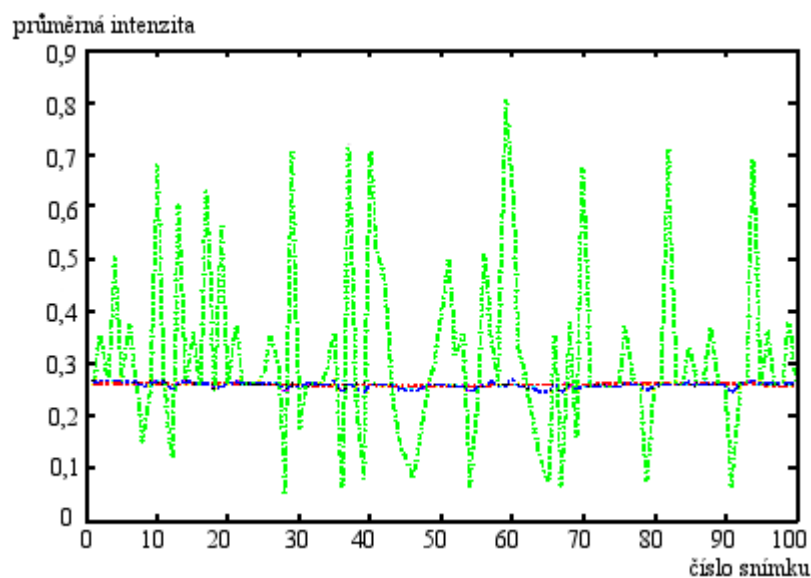
$$\sigma[I_t(x, y)] \approx \sigma[\hat{I}_{t-1}(x, y)] \quad (4.8)$$

Zde $E[\hat{I}_{t-1}(x,y)]$ a $\sigma[\hat{I}_{t-1}(x,y)]$ reprezentují předpokládanou hodnotu intenzity a standardní odchylku pro všechny pozice ve stejném bloku v předchozím obrázku.

Pro bloky, ve kterých dochází k pohybu, nejsou rovnice 4.7 a 4.8 platné. Nejprve se tyto rovnice musí přizpůsobit, aby bylo možné je aplikovat i na bloky, ve kterých dochází k pohybu. Pozice (x,y) v předchozím obrázku je nahrazena $x'=x-V_h$ a $y'=y-V_v$, kde V_h a V_v jsou horizontální a vertikální pohybový vektor. Po této kompenzaci pohybu je postup zpracování obdobný jako u stacionárních bloků. Obecné schéma této metody vypadá takto:

- a) přechod z RGB prostoru do YCbCr barevného prostoru
- b) rozdělení obrazu do sekcí 16x16 pixelů a spočítání pohybových vektorů pro každý blok
- c) spočítání průměrné intenzity bloku v obrazu a bloku, který odpovídá tomuto bloku v minulém obrázku; pokud je rozdíl těchto dvou hodnot větší než prahová hodnota c , provede se krok d, jinak krok e
- d) odhad parametrů alfa a beta v současném bloku, poté vyhlazení těchto dvou parametrů pomocí filtru
- e) pokud současný blok není posledním blokem v obrázku, přechází se na další blok, pokud je poslední, je nutné přejít k dalšímu obrázku ve filmu a zopakovat krok c, dokud se neprojdou všechny obrázky ve filmu.

Tato metoda byla testována na filmové sekvenci, která obsahovala jak uměle vytvořený blikavý šum, tak blikavý šum vzniklý přirozenou degradací filmu. Výsledky tohoto testu ukazují, že tato nová metoda umožňuje kvalitně odstranit veškerý přítomný blikavý šum a zároveň zachovat efekty, které sekvence má obsahovat, ale mající podobné parametry jako blikavý šum. Výsledky jsou poměrně jasně vidět z grafu intenzit (obrázek 4.1) . V obrázku 4.2 je také vidět sekvence, která je poškozená (obrázek 4.2 (a)) a opravená touto metodou (obrázek 4.2(b)). Tato metoda je v současnosti jedna z nejlepších používaných metod pro odstranění blikavého šumu.



obrázek 4.1: Zobrazení průměrné intenzity ve filmové sekvenci, v originální sekvenci (červená), v sekvenci s přidáním blikavým šumem (zelená) a v sekvenci po korekci blikavého šumu (modrá) [6]



(a)



(b)

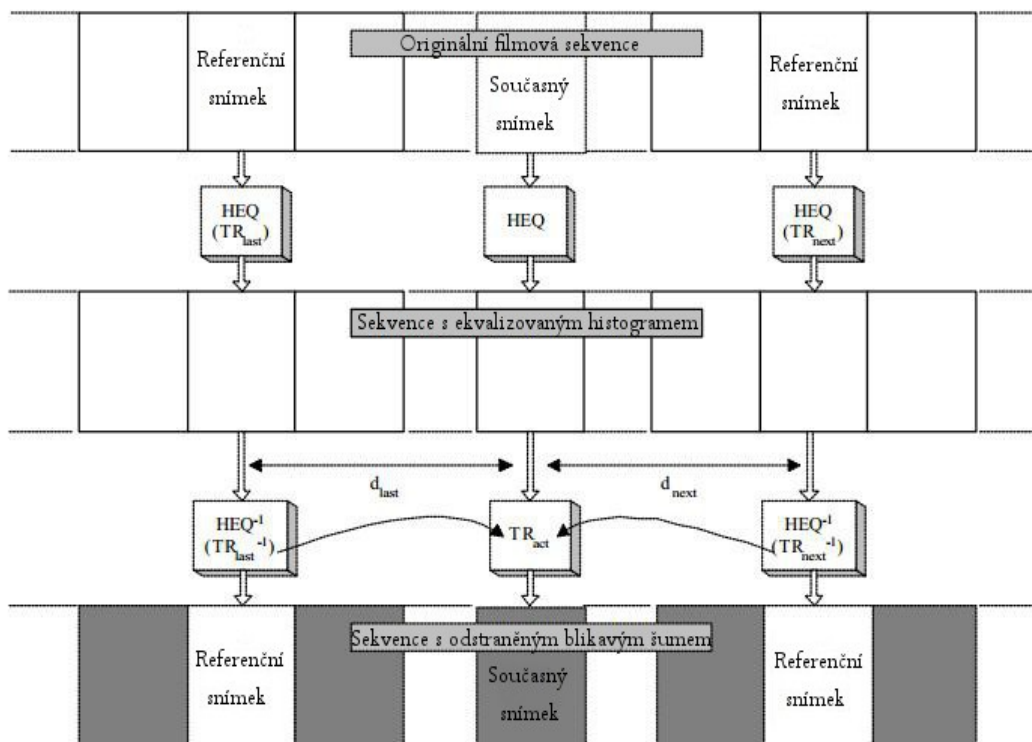
obrázek 4.2: první tři obrázky z filmu „Landmine Warfare“ postížené vadou blikavého šumu (a) originální sekvence (b) sekvence po úpravě pomocí metody na odstranění blikavého šumu [6]

4.2 Odstranění blikavého šumu pomocí ekvalizace histogramu

Korekce blikavého šumu touto metodou je rozdělena do dvou kroků. Nejprve musí technik, který řídí restauraci filmu, označit ve filmové sekvenci obrázky, které jsou blikavým šumem nepostiženy. Druhým krokem je poté korekce rozložení hodnot šedé v šedotónovém obrázku, nebo rozložení všech tří barevných složek v případě barevného filmu, ve všech ostatních obrázcích filmové stopy, tedy ve všech obrázcích, které nebyly označeny jako „nepostižené blikavým šumem“.

Referenční obrázky označené jako „nepostižené blikavým šumem“, by od sebe měly být vzdálené 5 až 20 snímků. Vzdálenost referenčních snímků je limitována především množstvím změny obsahu obrazu. Pokud tedy máme v obrazu rychlé změny, jako například jedoucí auto, výbuch atd., je důležité mít referenční snímky k sobě co možná nejbližší. Naopak pokud v obraze není téměř žádný pohyb, například záběr na moderátorku zpráv, můžeme si dovolit mít od sebe referenční snímky vzdálené i přes 20 snímků. Pokud dodržíme toto pravidlo a zvolíme referenční snímky správně, můžeme vycházet z předpokladu, že rozložení šedé (histogram) bude pro sousední referenční snímky téměř totožné.

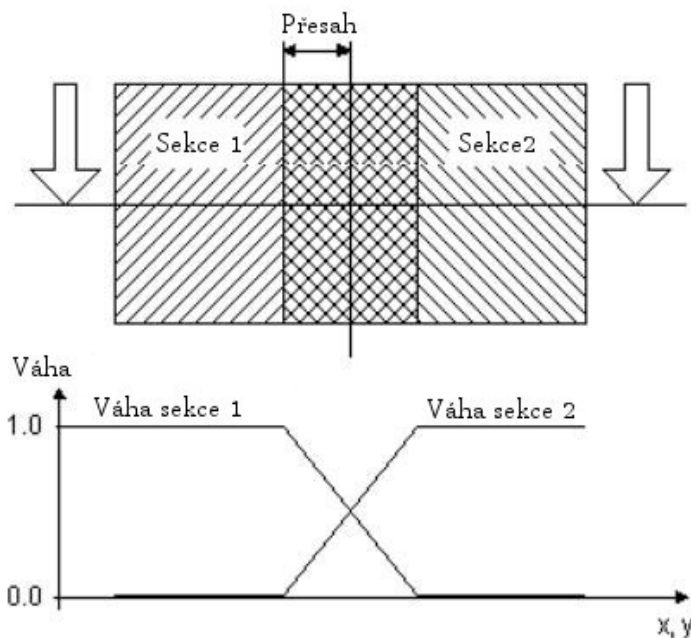
Hlavní myšlenka algoritmu je poté poměrně jednoduchá. Na všech snímcích filmové sekvence provedeme ekvalizaci histogramu. Snímky, které mají stejný obsah (například pořád zabírají mluvící moderátorku), vypadají po provedení ekvalizace stejně. Abychom poté dosáhli stejného rozložení šedi, jako měly původně referenční snímky, použijeme jejich inverzní funkci k ekvalizaci histogramu. TR_{last} je inverzní funkce nejbližšího minulého referenčního snímku a TR_{next} je inverzní funkce nejbližšího následujícího snímku. Lineární kombinace těchto funkcí je použita, abychom předešli nespojitostem. Čím blíže je referenční snímek k opravovanému snímku, tím větší je jeho váha v lineární kombinaci s druhým referenčním snímkem.



obrázek 4.2: Odstranění blikavého šumu pomocí ekvalizace histogramu [15]

V obrázku 4.2 HEQ představuje obrázek po provedení ekvalizace histogramu, HEQ^{-1} reprezentuje obrázek po provedení inverzní funkce k ekvalizační funkci, tedy funkci, která nám umožní získat zpět původní hodnoty jasu ve snímku. TR a TR^{-1} představují ekvalizační funkci a funkci k ní inverzní, d_{last} a d_{next} jsou vzdálenosti k nejbližšímu minulému a následujícímu referenčnímu snímku.

Tento přístup lze uplatnit pouze tehdy, pokud je blikavým šumem postižen celý obraz. Pokud by byl blikavý šum lokální, musíme výše uvedený postup modifikovat. Celý snímek rozdělíme do několika sekcí stejných rozměrů. Tyto sekce se musí překrývat, abychom ve výsledném spojení mohli dosáhnout plynulých přechodů. Velikost sekcí je adaptivní. Přizpůsobuje se pohybu v obraze: pokud máme v obraze málo pohybu, můžeme mít sekce velice malé a tím můžeme velmi kvalitně určit přesnou lokaci vady blikavého šumu a opravit ji. Pokud je však v obraze pohybu hodně, musí být tato sekce větší, aby metoda fungovala správně. Pro spojení jednotlivých sekcí do výsledného opraveného obrazu je použito váhování přesahů, tak jak je vidět v obrázku 4.3. Pomocí tohoto váhování dosahujeme plynulého přechodu mezi jednotlivými sekcemi.



obrázek 4.3: Váhování sekcí pro plynulý přechod intenzit[15]

Touto metodou lze dosáhnout velice dobrých výsledků, obdobně jako u Automatické metody. Přitom jsou obě metody založené na odlišném principu. Významným rozdílem těchto principů je, že Automatická metoda je citlivá na všechny druhy šumů. Oproti tomu metoda založená na vyvážení histogramu na šum citlivá není. Na druhou stranu Automatická metoda, jak její název napovídá, nepotřebuje ke své obsluze technika, kterého metoda založená na ekvalizaci histogramu potřebuje na označení referenčních snímků. Podle volby metody na odstranění blikavého šumu se liší pořadí aplikování algoritmů na odstranění ostatních vad tak, jak je tomu uvedeno v kapitole 2.8.

5. Degradace barev

Pro pochopení procesu, jak barvy na filmu degradují, je nejprve potřeba pochopit, jakou má vlastně film strukturu. Většina barevných filmů (především ty, které v současnosti potřebují restauraci) je vyrobena tak, jak vidíme na obrázku 5.1.



obrázek 5.1: Zobrazení barevných vrstev na filmu

Všechny vrstvy takového filmu jsou citlivé na modré světlo. Vrstva, která je nejvýše, je citlivá pouze na modré světlo. Další vrstvy jsou citlivé jak na modré světlo, tak na zelené (platí pro zelenou vrstvu), nebo na modré a červené světlo (platí pro červenou vrstvu). Při záznamu filmu je zapotřebí ze světla, které prochází skrz modrou vrstvu, odstranit modrou složku tak, aby neovlivňovala nižší vrstvy. Protože bílá bez modré vytváří žlutou, je pro vyjmutí modré složky použit žlutý filtr.

Při promítání filmu dochází k procesu, kdy intenzita bílého světla, které prochází skrz film, je ovlivněna intenzitou pigmentů ve třech vrstvách takovým způsobem, aby originální intenzity tří primárních barev - červené, modré a zelené byly zrekonstruovány na ty, které skutečně obsahoval zaznamenávaný obraz. To znamená, že pokud bylo modré světlo intenzivní, je malá hustota pigmentů ve vrstvě, která byla citlivá na modré světlo. Totéž platí pro ostatní barvy. Je také zapotřebí mít na paměti, že barvy na filmu musí být subtraktivní (Cyan, Magenta, Yellow = azurová, fialová, žlutá), protože pigmenty na barevných vrstvách se odčítají od procházejícího bílého světla během filmové projekce. Konverze barev z RGB do CMY je uvedena zde:

$$100=100\%$$

$$\text{Cyan} = 100 - \text{červená} \quad (\text{červená} = 100 - \text{Cyan})$$

$$\text{Magenta} = 100 - \text{zelená} \quad (\text{zelená} = 100 - \text{Magenta})$$

$$\text{Yellow} = 100 - \text{modrá} \quad (\text{modrá} = 100 - \text{Yellow})$$

Nyní můžeme přikročit k popisu degradace barev. Ta je způsobena blednutím pigmentových vrstev, především kvůli vystavení vzduchu. Vzhledem k tomu, že vrstva citlivá na modrou složku světla je nejvíce vystavena účinkům vzduchu, se předpokládá, že se začne vytrácet jako první. Další vrstva (zelená) nebledne tolik jako modrá, ale pořád bledne výrazně více než vrstva, která je nejnižší (červená). Proto se v praxi můžeme setkávat s filmy, které vypadají jako obrázek 5.2.



obrázek 5.2: Barevně degradovaný obrázek [4]

Obnovení barev se tedy sestává z:

- vyvážení nerovnoměrně vytrácejících se barevných vrstev
- zvýraznění slabší modré a zelené
- aplikování jednotné korekce pro obnovení věrohodnosti barev
- doladění rozdílů mezi jednotlivými snímky filmu

Restaurace filmů obvykle začíná tím, že se odstraní vady, které byly uvedeny v předchozích kapitolách, tedy skvrnové vady, škrábnutí, blikavý šum, šum. Žádný z algoritmů na úpravu těchto vad ale neopraví poškozenou kolorometrii.

Prvním krokem k obnovení barev je převod filmu do digitální podoby pomocí telecinu nebo skeneru. Skener nám poskytuje lepší rozlišení, ale je výrazně pomalejší. V případě restaurování filmových archivů, které mají historickou a uměleckou hodnotu, by tedy bylo vhodnější zvolit skener.

Druhým krokem je vyvážení obrazu za použití RGB ovládní telecinu nebo skeneru. Cílem v této fázi není vytvořit obraz, který vypadá normálně, ale co nejlépe zaznamenat všechny barvy, které ještě v obraze zůstávají. Tato fáze je obtížná, protože červený kanál je výrazně citlivější než zelený nebo modrý. Je vidět, že výsledek této operace je zkreslený, vypadá více azurově, než byl originální snímek. Obraz je také poměrně kontrastově plochý. Výsledek takovéto operace je vidět na obrázku 5.3 (b).



obrázek 5.3: (a) Barevně degradovaný obrázek (b) obrázek upravený telecinem [4]

Třetím krokem je použití da Vinci 2k systému¹, což je systém specificky navržený pro obnovení barev při restaurování filmů. Existují i jiné systémy nebo algoritmy pro obnovení barev, ale u žádných z nich se podle výsledků, které jsem mohl vidět, nedosahuje takové věrohodnosti barev. Tento systém by mohl být použit i bez předchozí úpravy pomocí telecinu, ale výsledky by nebyly tak dobré.

Tento systém nejdříve vyvažuje barvy v obrázku za použití jednoznačných vodítek. Například v obrázku, který je zde uveden jako příklad (5.3 (b)), můžeme jako referenci použít černou plachtu od kočáru a bílé šaty kočího. Výsledek takové operace je vidět na obrázku 5.4.

¹ Dostupné na: <http://www.finalcolor.com/historyDV.htm>



obrázek 5.4: Da Vinci 2k úprava fotografie pomocí referencí v obraze

Posledním krokem v takovéto úpravě filmu je doladění barev v obraze pomocí křivek. Díky nim může technik doladit detaily barevné korekce filmu. Lze zde umístit masku tak, aby chom nepracovali se stinnými a jasnými místy, která poté můžeme doladit nezávisle na barevné korekci ostatních barev v obraze. Výsledek takovéto operace lze vidět na obrázku 5.5 a představuje finální verzi obrazu. Protože se předpokládá, že by celý film měl degradovat podobně, měla by tato úprava být aplikovatelná i na všechny ostatní obrázky na filmu. Je ale zapotřebí celý film snímek po snímku projít a přesvědčit se, že jsme opravdu dosáhli optimálního výsledku, popřípadě jednotlivé snímky restaurovat samostatně, nezávisle na zbytku filmu.



obrázek 5.5: Da Vinci 2k finální verze snímku

6. Detekce a odstranění vady poškrábáním

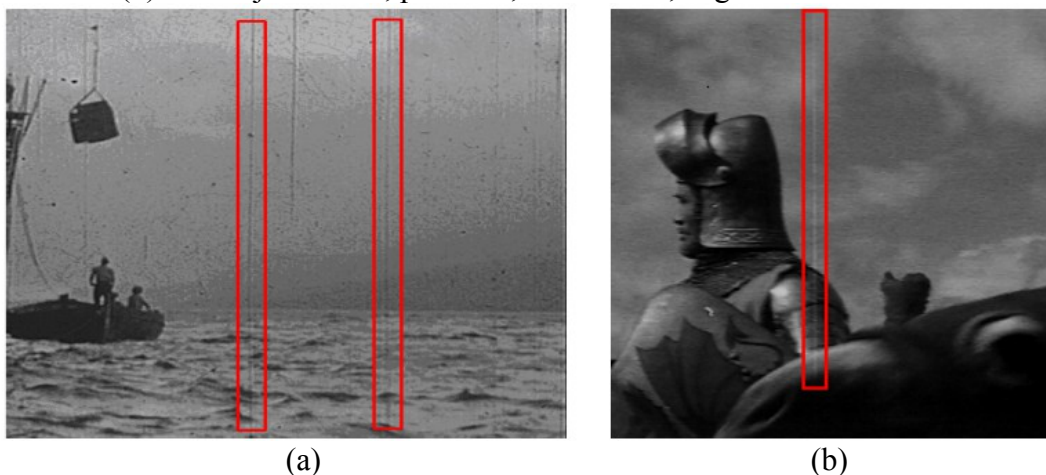
6.1 Úvod

Jak bylo řečeno v předcházejících kapitolách, archivní filmy jsou často poškozené degradací barev, blikavým šumem, škrábancemi a skvrnovými chybami. Nejčastější z vad je chyba škrábnutím. Škrábnutí na filmu obvykle vznikají mechanickým odřením při kopírování filmu nebo při jeho přehrávání. Archivní filmy obsahují několik druhů škrábanců a pro jejich detekci je důležité si je kategorizovat. Můžeme je rozdělit podle jejich délky, podle toho, zda se v průběhu sekvence pozice škrábnutí mění v jednotlivých obrázcích. Také je můžeme dělit podle toho, zda se škrábnutí projevuje jako světlý či tmavý pruh. Toto rozdělení je shrnuto v tabulce 6.1.

Druh poškrábání	Popis
statické poškrábání přesouvající se poškrábání	v obrázcích, které jdou po sobě v sekvenci, je pořad na stejné pozici může měnit svojí pozici v sekvenci
primární škrábnutí sekundární škrábnutí	je delší než 95% výšky obrázku ostatní škrábnutí
samostatné škrábnutí mnohonásobné škrábnutí	nemá žádné další přilehlé škrábnutí ve svém okolí má další škrábnutí
pozitivní škrábnutí negativní škrábnutí	po ploše škrábnutí je obrázek potměšlý po ploše škrábnutí je obrázek světlejší

tabulka 6.1: Rozdělení typů škrábnutí [2]

Všechny tyto typy škrábnutí jsou snadno viditelné jako vertikální linky světlé nebo tmavé intenzity. Příklad sekvencí, které obsahují škrábnutí, je uveden na obrázcích 6.1(a) a 6.1(b). Obrázek 6.1(a) obsahuje statické, primární, samostatné, pozitivní škrábnutí. Obsahuje ale také statické, sekundární, mnohonásobné, pozitivní škrábnutí. Obrázek 6.2(b) obsahuje statické, primární, samostatné, negativní škrábnutí.



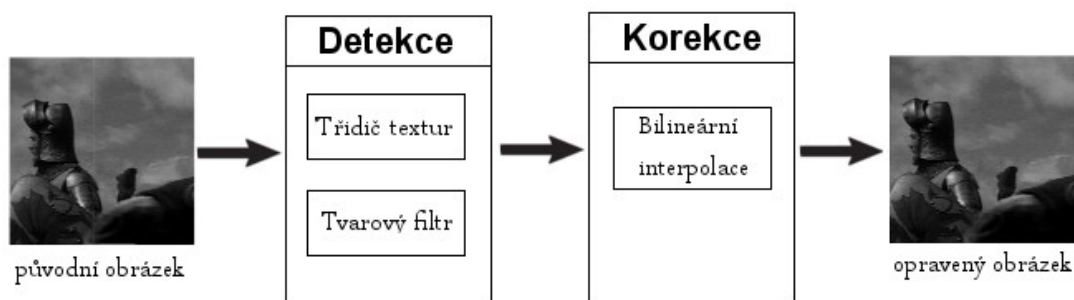
obrázek 6.1: Příklady škrábnutí: (a) statické, primární, samostatné, pozitivní škrábnutí a zároveň statické, sekundární, mnohonásobné, pozitivní škrábnutí (b) statické, primární, samostatné, negativní škrábnutí [2]

Všechna škrábnutí mají dvě hlavní společné vlastnosti: pixely na ploše škrábnutí mají menší nebo vyšší jas než pixely v jejich blízkém okolí a druhá vlastnost je, že škrábnutí se projevuje jako dlouhá vertikální linka. Na těchto vlastnostech jsou založeny zde prezentované automatické metody na detekování a odstranění škrábanců z archivních filmů.

6.2 Detekce vady poškrábáním

6.2.1 Detekce třídičem textur a tvarovým filtrem

Pro správnou detekci všech druhů škrábnutí ve filmové sekvenci je použita metoda využívající tvarové a texturové vlastnosti škrábnutí[2]. Každý obrázek ze sekvence je za pomoci texturové vlastnosti škrábnutí rozdělen do oblastí obsahujících škrábnutí a oblastí, které škrábnutí neobsahují. Poté jsou jednotlivé oblasti, které obsahují škrábnutí, filtrované pomocí tvarových parametrů škrábnutí. Proto je navrhovaná metoda složená ze dvou hlavních modulů: třídiče textur a tvarového filtru.



obrázek 6.2: Průběh metody pro odstranění chyb škrábnutím [2]

A. Třídič textur

Neuronová síť je použita jako filtr, který vytváří lokální klasifikace obrazových pixelů. Klasifikuje je jako „obsahující škrábnutí“ nebo „neobsahující škrábnutí“ pomocí analyzování vlastností částí obrázku. Systém obdrží šedotónové hodnoty pixelů na ploše o rozměrech $N \times M$. Hodnoty výstupního uzlu jsou vektor dvou čísel v rozsahu od 0 do 1. Pokud je první hodnota větší než druhá, jedná se o poškozený pixel. Pokud je první hodnota menší než druhá, jedná se o pixel, který je v pořádku. Výsledkem této klasifikace je binární obrázek, kde pixely, které byly detekovány jako vadné, jsou bílé a pixely, které byly označeny jako neobsahující vadu, jsou černé.

B. Tvarový filtr

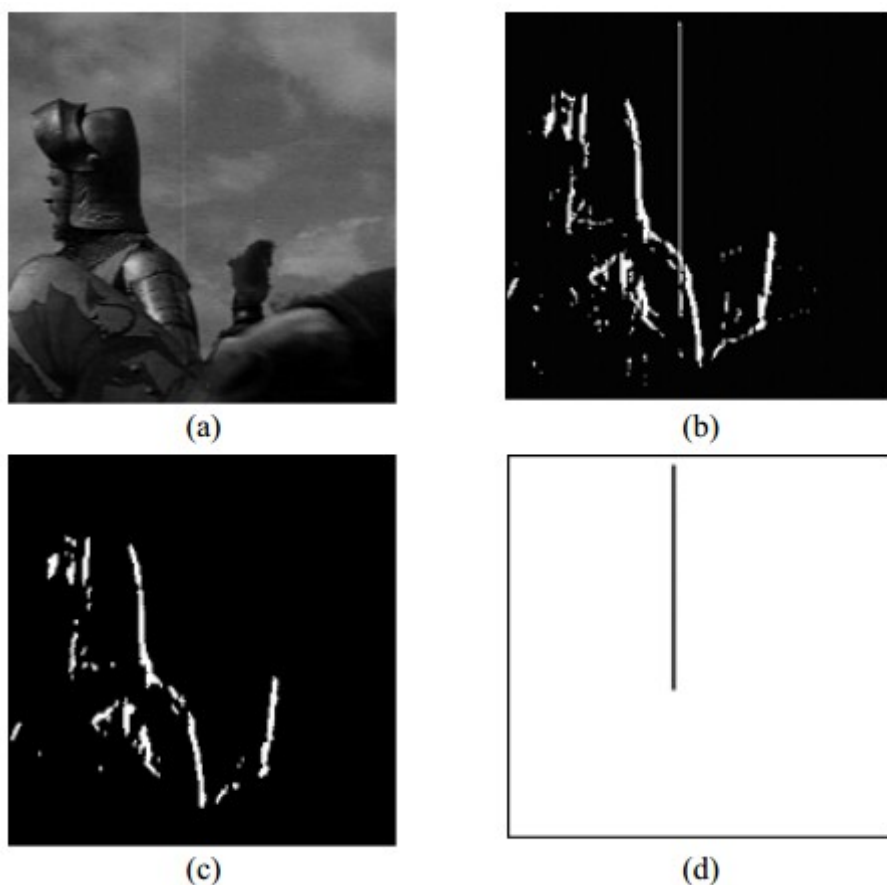
Pokud bychom vycházeli pouze z výsledků, které nám poskytne třídič textur, byly by oblasti obsahující vysoké frekvence označeny za obsahující škrábnutí, i když ho ve skutečnosti neobsahují. Abychom se zbavili těchto falešných detekcí, je použit tvarový filtr. Škrábnutí se obvykle projevuje jako dlouhá linie široká 3-5 pixelů. Díky této tvarové vlastnosti lze definovat tři parametry pomocí rovnic (6.1):

$$\beta_n = \begin{pmatrix} X & X & X & X & X \\ D & 1 & 1 & 1 & D \\ X & X & X & X & X \end{pmatrix} \quad \beta_L = \begin{pmatrix} X & X & 1 & 1 & D \\ X & D & 1 & D & X \\ D & 1 & 1 & X & X \end{pmatrix}$$
$$\beta_n = \begin{pmatrix} D & 1 & 1 & X & X \\ X & D & 1 & D & X \\ X & X & 1 & 1 & D \end{pmatrix} \quad (6.1)$$

V rovnicích 6.1 je X proměnná, která nás nezajímá, D je proměnná, která je určena rozlišením obrázku: je nastavena do 1 pro obrázky s vysokým rozlišením, jinak je nastavena na 0. B_H je definován pro detekování horizontálních komponent a B_L je definován pro detekování diagonálních komponent. Morfologickými operacemi s těmito strukturami jsou odstraněny špatně určené škrábance. Pokud N reprezentuje výsledky třídiče textur, jsou výsledky morfologického filtrování S získány pomocí rovnice 6.2.

$$S = N - [\gamma_{B_L}(N) + \gamma_{B_K}(N) + \gamma_{B_H}(N)] \quad (6.2)$$

Obrázek 6.3 ukazuje výsledky vyhledávacího algoritmu. 6.3(a) reprezentuje originální obrázek. Obrázek 6.3(b) ukazuje výsledek po použití pouze texturové klasifikace. Na obrázku 6.3(c) potom můžeme vidět falešné detekce, které byly odhaleny pomocí tvarového filtru. Obrázek 6.4(d) zobrazuje výslednou detekci po použití obou filtrů.



obrázek 6.3: Příklad detekce škrábnutí: (a) originální obrázek, (b) výsledek detekce pomocí textur, (c) oblasti, u kterých došlo k falešné detekci, (d) potvrzená oblast obsahující vadu škrábnutím [2]

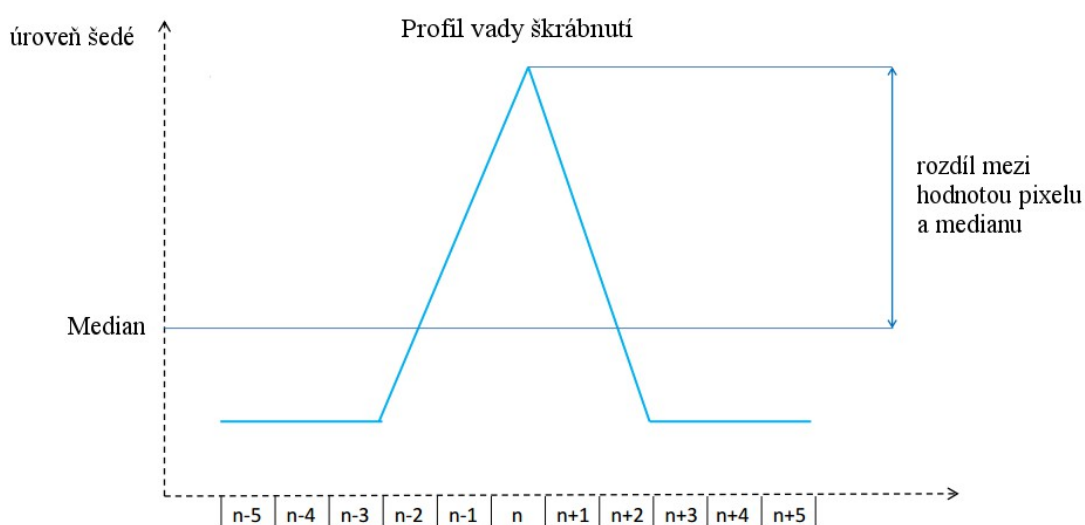
6.2.2 Detekce mediánovým filtrem a Houghovou transformací

Tento algoritmus [12] se skládá ze dvou kroků. Prvním je detekce potenciačních škrábnutí, ve kterém se rozhoduje, jestli pixel může být součástí škrábance. Druhým krokem je použití Houghovy transformace pro sjednocování pixelů, které byly označeny za potenciační součást škrábnutí, do skupin, které tvoří jednotlivá škrábnutí.

Nejprve je tedy třeba identifikovat pixely, které mohou být součástí škrábnutí. Je několik způsobů, jak mohou být škrábance detekovány, jako například použití morfologických filtrů, nebo 1-dimenzionální detekce extrémů. Metoda, která je zde prezentována pro tuto detekci, byla použita A. Kokaramem [12]. Je založena na prahování rozdílu vertikálně podvzorkované verze snímku a horizontálně mediánově filtrované verze snímku (6.3).

$$e(x, y) = G_s(x, y) - M_s(x, y) \quad (6.3)$$

V rovnici (6.3) $G_s(x, y)$ reprezentuje vertikálně podvzorkovanou verzi vstupního snímku a $M_s(x, y)$ je horizontálně mediánově filtrovaná verze $G_s(x, y)$. Tato rovnice tedy rozhodne, zda je pixel viditelně rozdílný od svého horizontálního okolí. Protože samotné takové kritérium detekuje i nechtěné okraje objektů v obraze, jsou navíc porovnávány hodnoty v pravém a levém okolí identifikovaného pixelu. Předpokládá se, že by pravé a levé okolí mělo být do určité míry koherentní, pokud je pixel skutečně součástí škrábnutí. Princip horizontální mediánové filtrace a hodnocení koherentnosti okolí je vidět na obrázku 6.4. V tomto případě by mediánová horizontální filtrace byla provedena po pěti pixelech. $n-5$ až $n-1$ a zároveň $n+2$ až $n+5$ tvoří pravé, respektive levé okolí skvrny, která se nachází na pixelech $n-1$ až $n+1$. Zjednodušeně řečeno, pokud provedeme horizontální mediánovou filtraci na místě, kde je vada škrábnutí, mediánová filtrace ji za cenu rozmazání obrazu vyhledá tak, že poškrábání z obrazu zmizí. Takovýto obraz odečteme od původního, ve kterém škrábnutí je. V místech se stálou úrovní barvy se při použití mediánového filtru nic zásadního nezmění, a proto po odečtení dostáváme hodnotu blízkou nule. Pokud od sebe ale odečítáme místa obsahující skvrnu a neobsahující skvrnu, dostaneme značně odlišnou hodnotu. Poté se podíváme do levého a pravého okolí námi právě detekovaného pixelu a pokud jsou okolí přibližně stejná, vyhodnotíme pixel jako vadný, pokud se okolí liší, pixel vyhodnotíme jako nepoškozený. Jako velice jednoduchý příklad je možno představit si fotografii vězně, který má pruhované triko. Jednotlivé hrany pruhů by v normálním případě byly detekovány jako vadné, proto porovnáváme ještě levé a pravé okolí. Levé okolí by v tomto případě bylo bílé, pravé okolí by bylo černé a z toho vyplývá, že se nejedná o skutečnou vadu, ale pouze o hranu objektu v obraze. Detailnější popis je dostupný v [12].



obrázek 6.4: Profil vady škrábnutí [12]

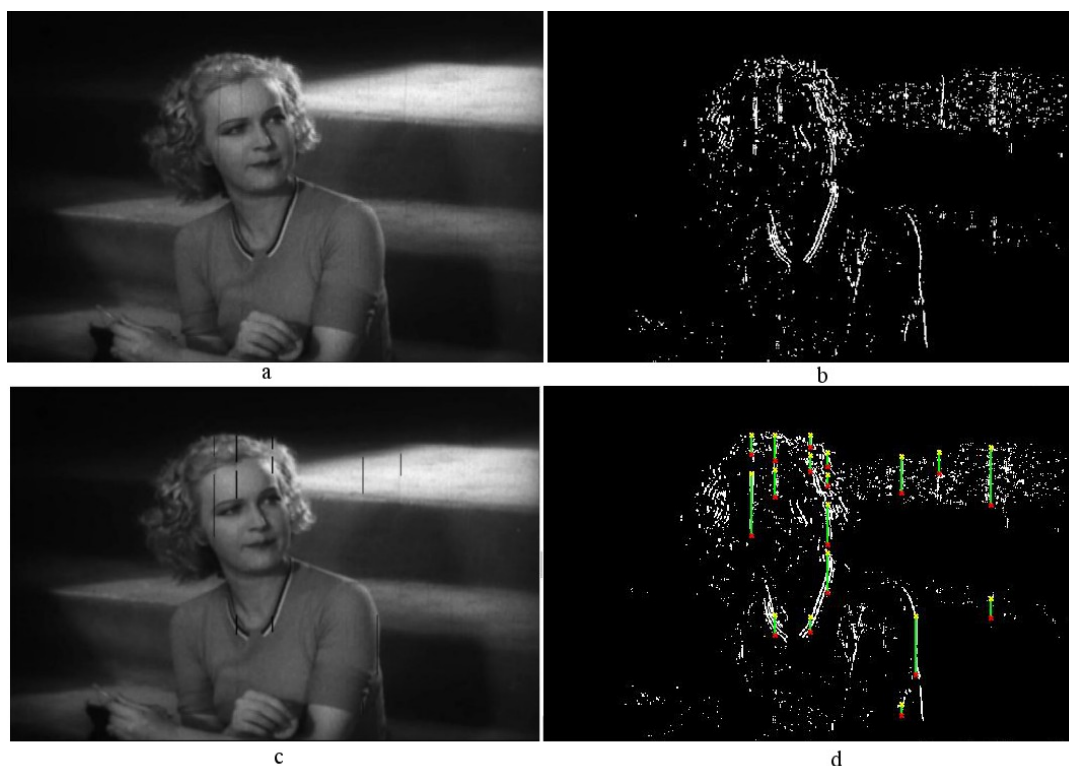
Předchozí kritérium je třeba reprezentovat rovnicí (6.4). V této rovnici $I_g(x,y)$ reprezentuje pixel ve vertikálně podvzorkovaném obrázku. $I_m(x,y)$ je pixel ze snímku po mediánové filtraci. $I_l(x,y)$ a $I_r(x,y)$ představují průměr levého a pravého okolí bodu x,y . S_{med} a s_{avg} jsou prahové hodnoty, které byly určeny empiricky na hodnoty $s_{med}=3$ a $s_{avg}=20$.

$$\begin{aligned} c_1(x, y): |I_g(x, y) - I_m(x, y)| > s_{med} \\ c_2(x, y): |I_l(x, y) - I_r(x, y)| < s_{avg} \end{aligned} \quad (6.4)$$

Po použití rovnic (6.4) můžeme definovat binární obrázek představující detekci vady škrábnutí jako (6.5). Ukázka takovéto binární detekce je vidět na obrázku 6.5(b).

$$\begin{aligned} I_b(x, y) &= 1 \text{ pokud } c_1(x, y), c_2(x, y) \text{ jsou splněny} \\ I_b(x, y) &= 0 \text{ v ostatních případech} \end{aligned} \quad (6.5)$$

Pokud je vytvořena detekce potencionálních pixelů postižených vadou škrábnutí, je potřeba v ní najít pixely, které skutečně škrábnutí reprezentují. K tomu je použita Houghova transformace a její schopnost detekce přímek v obraze. Jedinou skutečnou nevýhodou tohoto přístupu je nutnost složitého empirického nastavování několika parametrů pro detekci přímek. Po vhodném nastavení může detekce vypadat například jako na obrázku 6.5.

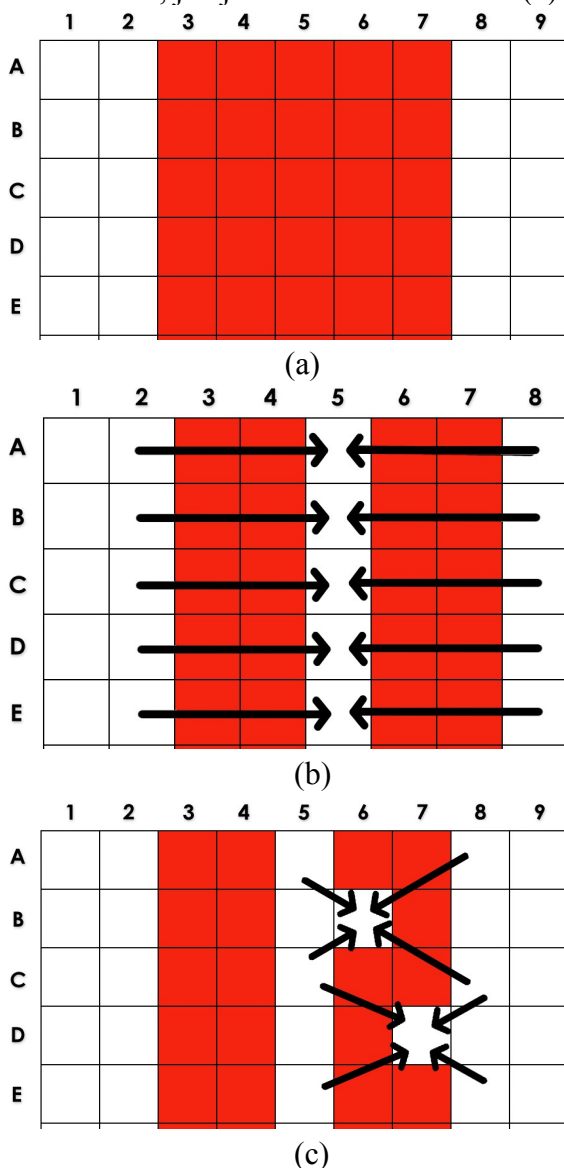


obrázek 6.5: (a) původní snímek (b) detekce všech pixelů, které mohou být součástí škrábnutí (d) výsledek Houghovy detekce (c) promítnutí detekce do původního obrázku

6.3 Odstranění škrábnutí

Pro obnovení původního obrazu je použita bilineární interpolační metoda. Bilineární interpolace je založená na vyplnění poškozené oblasti pomocí informace obsažené v přilehlých pixelech. Nově generované pixely jsou tedy vážená suma čtyř nejbližších pixelů. Váhování je definováno lineárně, kde každá váha je přímo úměrná vzdálenosti od daného nepoškozeného pixelu.

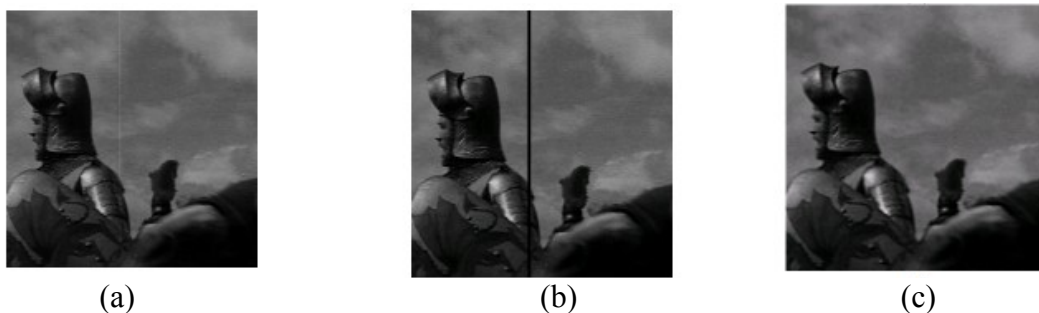
Obrázek 6.6 zobrazuje postup obnovení poškozených pixelů. V obrázku 6.6 (a) jsou škrábnuté pixely označeny jako červené a nepoškrábané pixely jsou označeny jako bílé. Nejprve jsou obnoveny pixely ve středu škrábnutí interpolováním nejbližších sousedních pixelů, které nejsou poškozené (obrázek 6.6 (b)). Poté jsou obnoveny pixely ležící mezi pixely, které nebyly poškrábané a mezi pixely uprostřed, které byly obnoveny v předchozím kroku tak, jak je vidět na obrázku 6.6.(c).



obrázek 6.6: Proces obnovení poškozených pixelů: (a) škrábnuté pixely (označeny červeně) a nepoškrábané pixely (označeny bíle), (b) obnovení pixelů ve středu škrábnutí, (c) obnovení ostatních pixelů za použití čtyř nepoškozených pixelů [2]

6.4 Vyhodnocení

Výše uvedený algoritmus pro odstranění škrábnutí byl použit na poškozený film. Zde uvedu výsledky, přejaté z literatury [2], kterých se pomocí algoritmu s třídičem textur a tvarovým filtrem dosáhlo u filmů „Knight“ a „Titanic“. Tyto sekvence obsahovaly škrábance všech typů. Pro potřeby testu bylo vybráno 100 obrázků z celé sekvence, které obsahovaly škrábance. Výsledky jsou vidět na obrázcích 6.7, 6.8.



obrázek 6.7: Odstranění škrábnutí z filmové sekvence „Knight“: (a) originální obrázek, (b) detekované škrábnutí, (c) opravený obrázek [2]



obrázek 6.8: Odstranění škrábnutí z filmové sekvence „Titanic“: (a) originální obrázek, (b) detekované škrábnutí, (c) opravený obrázek [2]

Výsledky, kterých lze dosáhnout pomocí těchto metod, jsou velice dobré. Navíc jsou tyto metody automatické, tzn. nepotřebují trvalou přítomnost technika. Vzhledem k těmto skutečnostem lze tyto metody považovat za vhodné pro restaurování filmových archivů, které jsou rozsáhlé a jejichž restaurace má být v nejlepší možné kvalitě.

7. Šum

7.1 Úvod

Všechny filmové záznamy jsou postiženy šumem. Jak moc jsou postiženy, závisí především na kvalitě nahrávacího zařízení, které bylo použito k jejich pořízení. Žádné nahrávací zařízení není ideální, tzn. šum je přítomen vždy. Tato šumová chyba se ještě výrazně zvětšuje při kopírování filmu. Jak bylo řečeno v úvodní kapitole 2.3, kopie filmů dříve vznikaly pomocí telecinů. Starší teleciny měly poměrně špatné vlastnosti z hlediska šumových charakteristik. V mnoha případech jsou kopie filmů jediné, co zbylo a originál byl buďto zničen, nebo ztracen. Proto je odstranění šumu z obrazu velice důležité při procesu restaurace většiny archivních filmů.

Už před 25 lety se mnoho badatelů zabývalo digitálními algoritmy pro redukci šumu. Například dobře známé a populární teorie vytvořené Wienerem a Kalmanem pro optimální lineární filtraci byly aplikovány v digitální doméně velice často. V dnešní době se nabízí mnoho různých přístupů k redukci šumu v obraze. Jedním z nich, který získal velký vzhlas v posledních letech a prokázal se být velmi úspěšným při odstraňování šumu z 2D obrázků, je využití jádrové funkce (z anglického coring function). V této práci se budu věnovat jak výše zmíněné Wienerově filtraci, tak nelineárnímu odstraňování šumu z obrazu pomocí jádrové funkce.

7.2 Potlačení šumu pomocí lineárních metod

Aby zde mohla být popsána Wienerova filtrace, zmíněná v kapitole 7.1, je nejprve zapotřebí definovat některé pojmy. Za předpokladu, že $y(i)$ je originální obraz nepostižený šumem a $n(i)$ je šum, by byl obraz, který můžeme vidět, popsán rovnicí:

$$z(i) = y(i) + n(i) \quad (7.1)$$

Jedním z typů lineárních filtrů jsou filtry FIR (finite impulse response) neboli filtry s konečnou impulzovou odezvou, které jsou definovány takto:

$$\hat{y}(i) = \sum_{k=1}^n h_k * z(i + q_k) \quad (7.2)$$

V této rovnici h_k představuje koeficienty filtru a q_k definuje podporu od filtru. Optimální koeficienty filtru v rámci MMSE (minimum mean square error) mohou být nalezeny takto:

$$\arg \min_{h_1, \dots, h_k} E[(y(i) - \hat{y}(i))^2] \quad (7.3)$$

Výsledkem hledání těchto optimálních koeficientů je Wienerův filtr. Wienerův filtr může být efektivně implementován nad Fourierovou doménou. Fourierova transformace vycházející ze vztahu 7.1 je rovnice 7.4 a odhad $Y(\omega)$ je dán rovnicí 7.5.

$$Z(\omega) = Y(\omega) + N(\omega) \quad (7.4)$$

$$\hat{Y}(\omega) = \frac{S_{yy}(\omega)}{S_{yy}(\omega) + S_{nn}(\omega)} Z(\omega) \quad (7.5)$$

V rovnici 7.5 $S_{yy}(\omega)$ a $S_{nn}(\omega)$ představují spektrální výkonovou hustotu (PSD) obrazu nepoškozeného šumem a samotného šumu. Je zde vidět, že každá frekvenční položka pozorovaných dat (dat z obrazu postiženého šumem) je váhovaná v závislosti na spektrální výkonové hustotě originálního šumem nepoškozeného obrazu a šumu.

7.3. Potlačení šumu pomocí nelineární metody

Využití jádrové funkce (z anglického coring function) je dobře známá technika pro odstraňování šumu z obrazu. Podrobně jí popisuje práce [3], na které je založena tato kapitola. Tato technika se skládá z převodu signálu do frekvenční domény a z redukování frekvenčních koeficientů pomocí jádrové funkce. Inverzní transformace koeficientů po jejich redukcii jádrováním dává obraz s redukováným obsahem šumu. Tato část práce bude věnována vytvoření struktury pro jádrování filmu, tedy sekvenci obrázků. Struktura bude založená na 3D dekompozici obrazu, která dovoluje využití časové domény. Tím můžeme dosáhnout lepšího výsledku, než kdybychom přistupovali ke každému obrazu z filmové sekvence jednotlivě.

Wienerova filtrace využívá FIR filtru pro dosažení nejlepších možných výsledků v rámci MMSE. Nejlepších výsledků při hledání funkce pro MMSE je většinou dosaženo, když na strukturu filtru nejsou kladeny žádné další požadavky. Taková funkce potom bývá často nelineární. Jak je uvedeno v literatuře [3], optimální odhad $Y(\omega)$, pokud máme pouze jednu realizaci $Z(\omega)$, je nalezen za splnění podmínky (7.6):

$$\begin{aligned} E[(Y(\omega) - \hat{Y}(\omega))^2] &= E[E[(Y(\omega) - \hat{Y}(\omega))^2 | Z(\omega)]] = \\ &= \int_{-\infty}^{\infty} E[(Y(\omega) - \hat{Y}(\omega))^2 | Z(\omega)] \cdot P[Z(\omega)] dz(\omega) \end{aligned} \quad (7.6)$$

Integrál v rovnici 7.6 je kladný pro všechna $Z(\omega)$; to znamená, že integrál je minimalizován minimalizováním $E[((Y(\omega) - Y(\omega))^2 | Z(\omega))]$ pro všechna ω . Toto minimum je dáno rovnicí (7.7):

$$\hat{Y}(\omega) = E[Y(\omega) | Z(\omega)] \quad (7.7)$$

Obecné řešení rovnice (7.7) dává nejmenší možný výsledek z hlediska MMSE pro odhadování $Y(\omega)$, při jedné známé realizaci $Z(\omega)$. Obecně Wienerovo řešení bude mít větší MSE. Dalším rozvojem rovnice (7.7) získáme rovnici (7.8):

$$\hat{Y}(\omega) = E[Y(\omega) | Z(\omega)] = \int_{-\infty}^{\infty} Y(\omega) \cdot P_{Y(\omega) | Z(\omega)}[Y(\omega) | Z(\omega)] dY(\omega) \quad (7.8)$$

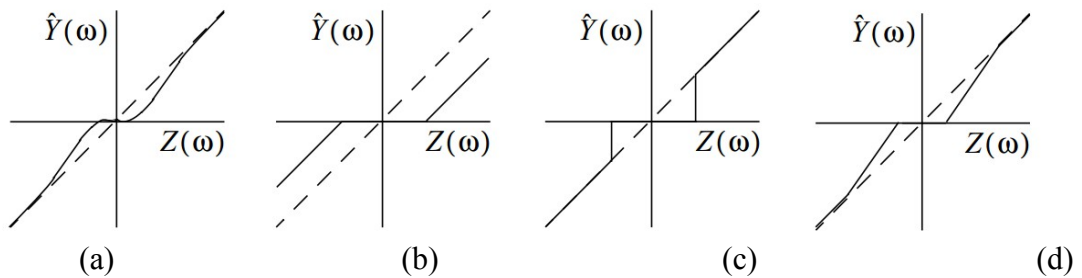
Pokud známe distribuci $Y(\omega)$ a $N(\omega)$, může být funkce P určena pomocí Bayesova pravidla, viz rovnice (7.9):

$$P_{Y(\omega)|Z(\omega)}[Y(\omega)|Z(\omega)] = \frac{P_{Z(\omega)|Y(\omega)}[Z(\omega)|Y(\omega)] \cdot P_{Y(\omega)}[Y(\omega)]}{P_{Z(\omega)}[Z(\omega)]} = \frac{P_N(\omega)[Z(\omega) - Y(\omega)] \cdot P_{Y(\omega)}[Y(\omega)]}{\int_{-\infty}^{\infty} P_N(\omega)[Z(\omega) - Y(\omega)] \cdot P_{Y(\omega)}[Y(\omega)] dY(\omega)} \quad (7.9)$$

V rovnicích (7.7), (7.8) a (7.9) ω reprezentuje frekvenci. Důležité je si uvědomit, že tato frekvence nemusí být nutně získána Fourierovou transformací. Jiné transformace, jako například DCT, vlnková transformace nebo pásmové transformace mohou být také použity a jak bude ukázáno dále v textu, budou daleko vhodnější.

7.3.1 Jádrové funkce

Obrázek 7.1(a) ukazuje typickou výslednou charakteristiku, která vychází z rovnice 7.8. Tato charakteristika se nazývá jádrová funkce. Obecně jádrové funkce zanechávají transformační koeficienty s velkými amplitudami nezměněné a koeficienty s malými amplitudami jsou zmenšovány k nule. Intuitivně je toto velice slibné. Koeficienty s velkými amplitudami jsou spolehlivé, protože jsou relativně málo ovlivněny šumem. Tyto koeficienty by se neměly měnit. Koeficienty s nízkými amplitudami nesou poměrně málo informace a jsou snadno ovlivněny šumem. To znamená, že tyto koeficienty jsou nespolehlivé a jejich příspěvek k pozorovaným datům by měl být redukován. Původně bylo jádrování vyvinuto jako heuristická technika pro odstraňování šumu. Tři obecně známé jádrové funkce jsou popsány níže.



obrázek 7.1: jádrové funkce (a) optimální jádrování (b) jemné prahování (c) striktní prahování (d) lineární jádrování po částech

Jemné prahování

Toto prahování je definováno podle [3]:

$$\hat{Y}(\omega) = \text{sgn}(|Z(\omega)| - T) \cdot (|Z(\omega)| - T) \quad \text{pro } |Z(\omega)| > T$$

$$\hat{Y}(\omega) = 0 \quad \text{jinde} \quad (7.10)$$

$\text{Sgn}(Z(\omega))$ udává znaménko (fáze) $Z(\omega)$. Obrázek 7.1(b) zobrazuje tuto jádrovou funkci. Reálné signály mívají tendenci mít slabé vysokofrekvenční komponenty. Proto jemné prahování nuluje vysokofrekvenční transformační koeficienty získané ze signálu. Výsledkem je, že kromě odstranění šumu, jsou rozmazané hrany na obrázku. Jemné prahování má také další efekt, konkrétně redukuje kontrast, protože zmenší rozptyl všech transformačních koeficientů.

Striktní prahování

Toto prahování je definováno podle [3]:

$$\begin{aligned}\hat{Y}(\omega) &= Z(\omega) \text{ pokud } |Z(\omega)| > T \\ \hat{Y}(\omega) &= 0 \text{ jinde}\end{aligned}\quad (7.11)$$

Tato funkce je zobrazena na obrázku 7.1 (c). Nevýhodou tohoto přístupu je vznik nežádoucích oscilací v obraze. Ty vznikají, protože striktní prahování neodstraňuje pouze šumovou energii na zvolených frekvencích, ale také signálovou energii. Odebrání signálové energie můžeme pozorovat jako přidání impulzů do originálního nepoškozeného obrazu. Amplitudy těchto impulzů jsou stejné jako ty, které originální obraz obsahoval, ale jejich znaménka jsou opačná. Když je pak signál transformován zpátky z frekvenční oblasti do prostorové, promítne se impulzní odezva syntetizujícího filtru do výsledku.

Lineární použití jádrové funkce po částech

Toto prahování je kompromisem mezi striktním prahováním a jemným prahováním a je zobrazeno na obrázku 7.1 (d):

$$\begin{aligned}\hat{Y}(\omega) &= Z(\omega) \text{ pro } |Z(\omega)| > T_1 \\ \hat{Y}(\omega) &= \frac{|Z(\omega)| - T_0}{T_1 - T_0} \cdot \text{sgn}(Z(\omega)) \text{ pro } T_0 < |Z(\omega)| < T_1 \\ \hat{Y}(\omega) &= 0 \text{ jinde}\end{aligned}\quad (7.12)$$

7.3.2 Prostorově variantní transformace

Implementace Wienerovy filtrace ve frekvenční doméně, jak byla popsána v sekci 7.2, může být vnímána jako jistá implementace jádrové funkce: každá frekvenční složka je upravená podle charakteristiky, která je daná PSD signálu a šumu. Použití Fourierovy transformace jako dekorelující transformace má ovšem nevýhodu ve ztrátě informace o prostorových lokacích dominantních signálových složek. Z toho vyplývá, že signál po aplikaci jádrové funkce není přizpůsobivý lokálním statistikám v obraze, ale je závislý pouze na globálních statistikách. Je zřejmé, že toto řešení není ideální, protože lokální statistiky se mohou výrazně lišit od globálních statistik.

Důvodem pro transformaci dat před aplikací jádrové funkce je co možná nejlepší oddělení signálu od šumu, abychom dosáhli optimálních výsledků. Narozdíl od Fourierovy transformace prostorově variantní transformace dovolují, aby byly brány v úvahu lokální signálové charakteristiky. V případě redukce šumu v obrazových sekvencích je adaptace na lokální statistiky výhodná kvůli nestacionaritě obrazu.

V této práci budou popsány dvě 2D prostorově variantní dekompozice. První je nedecimovaná vlnková transformace známá také jako *algoritmus a trous*[9]. Druhá je subpásmová dekompozice založená na směrově citlivých filtrech, která je známá jako *Simoncelliho pyramida*[9]. Poté bude vysvětleno, jak tyto dekompozice mohou být rozšířeny do třech dimenzí přidáním časové informace o obraze. Jak bude ukázáno níže, 3D separace poskytuje velmi dobré oddělení signálu od šumu.

Nedecimovaná vlnková transformace

Diskrétní vlnková transformace (DWT) je populární nástroj pro získání prostorově variantní reprezentace dat. Nejznámější implementace DWT je decimovaná DWT, ve které mají transformovaná data stejný počet koeficientů jako vstupní data. Problémem této transformace je prostorový posun vstupního obrazu, což může vést ke zcela rozdílným distribucím energie signálu nad transformačními koeficienty[10]. Tento nedostatek je způsoben podvzorkováním, které je použito v DWT.

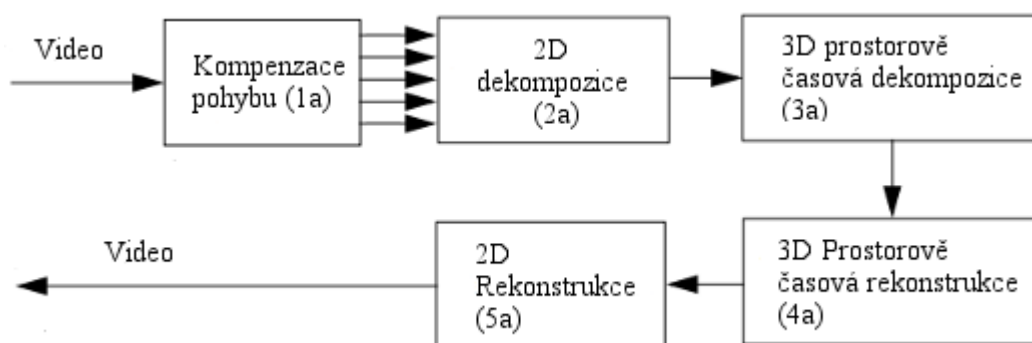
Prostorové neměnnosti lze dosáhnout pomocí nedecimované DWT. Algoritmus, který generuje nedecimovanou vlnkovou transformaci, je algoritmus a trous. Protože v tomto algoritmu není použito žádné podvzorkování, dekompozice má výrazně víc koeficientů, než měl původní prostorově reprezentovaný obraz. Například při třívrstvé dekompozici obrazu o N pixelech získáme $10N$ transformačních koeficientů.

Simoncelliho pyramida

Simoncelliho pyramida je subpásmová dekompozice založená na směrově citlivých filtrech[91]. To znamená, že distribuce energie signálu ve frekvenčních pásmech závisí na orientaci struktury v obraze. Prostorové neměnnosti je dosaženo tím, že se vyhýbáme efektu aliasingu tak, že zajistíme nepřítomnost komponent s frekvencí větší než $\pi/2$ před operací podvzorkování 2:1. Stejně jako nedecimovaná vlnková transformace i Simoncelliho pyramida obsahuje po transformaci výrazně víc koeficientů, než obsahoval původní obraz. Například čtyřstupňová pyramidová dekompozice se čtyřmi směrovými orientacemi nám z obrazu o N pixelech dává $9.3N$ koeficientů.

7.3.3 Využití třetí dimenze v transformaci

Dvoudimenzionální dekorelační transformace, jak byla popsána výše, dokáže oddělit signál od šumu. Tato separace však může být vylepšena využitím pohybově kompenzovaných obrazů, které jsou před současným obrazem. Využit lze samozřejmě i pohybově kompenzované obrazy, které následují po současném obrazu. Pohybově kompenzované obrazy $t-n, \dots, t+m$ by měly být stejné jako obraz t až na šum. Jak dosáhnout 3D prostorově variantní dekompozice/rekonstrukce je zobrazeno na obrázku 7.2.



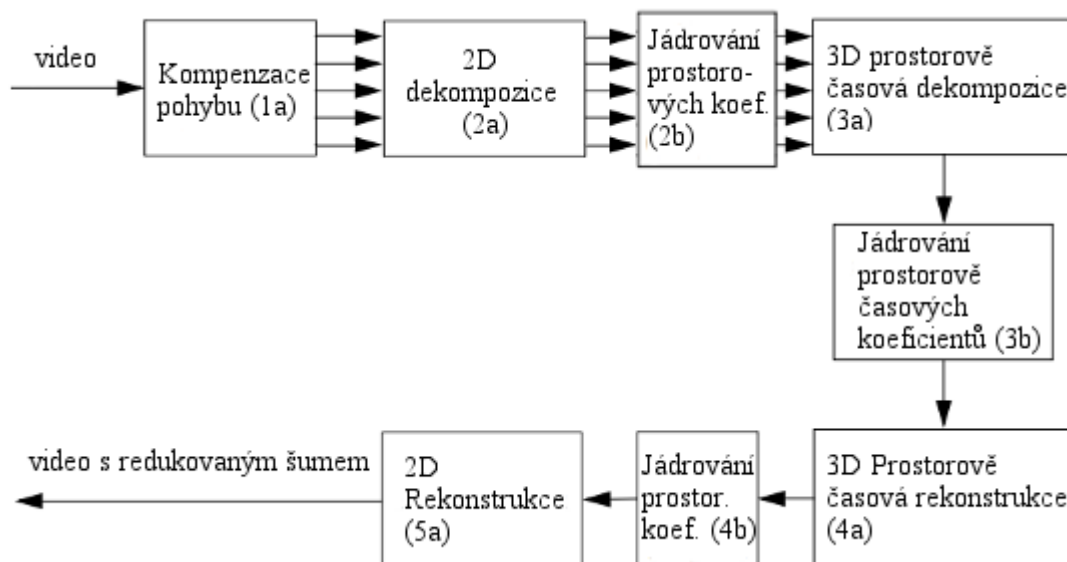
obrázek 7.2: Schéma 3D dekompozice [3]

7.3.4 Redukce šumu pomocí jádrování

Struktura navrhovaného algoritmu pro výše popsanou dekompozici/rekonstrukci obrazu je vidět na obrázku 7.3. Tento obrázek představuje kostru pro třídimenzionální prostorově variantní redukci šumu pomocí jádrování.

Nelineární podstata jádrových funkcí ztěžuje správné zvolení kombinace charakteristik v krocích 2b, 3b, a 4b, pro které by došlo k optimální redukci šumu. Také je třeba zauvažovat nad tím, zda je jádrování nezbytně nutné ve všech krocích 2b, 3b, a 4b. Abychom využili třetí dimenzi při dekompozici, jádrování je nezbytné v kroku 3b. Ale v bloku 2b a 3b se v některých případech jádrování nevyplatí použít.

Experimenty založenými na výpočtech bylo zjištěno, že je nejvýhodnější použít v kroku 2b optimální jádrovací funkci, kterou lze spočítat z rovnice (7.8) pro každé pásmo 2D dekompozice. K tomu je zapotřebí mít odhad nebo předpoklad rozložení signálu a šumu v jednotlivých pásmech. V kroku 3b je nejlepší použít striktní prahování (7.11). Určit optimální jádrovací funkci v kroku 4b je výrazně složitější, protože její volba by měla být závislá na tom, jaké funkce jsme použili v kroku 2b a 3b. Většinou je zde ale použito jemné prahování (7.10).



obrázek 7.3: Schéma redukce šumu pomocí jádrování [3]

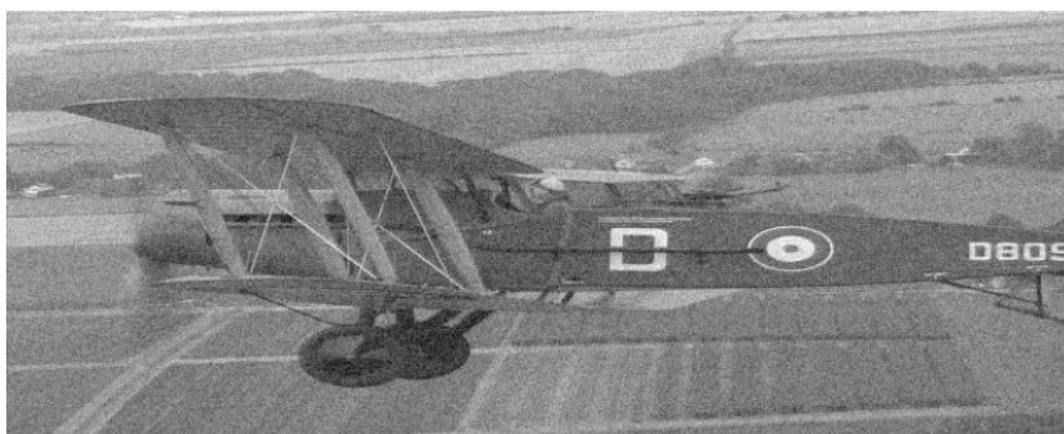
7.3.5 Vyhodnocení redukce šumu po použití jádrové funkce

V této sekci se budu věnovat experimentu, který byl proveden na filmové sekvenci nazvané *Plane*. Tato sekvence zobrazuje letadlo letící nad krajinou. Obsahuje velké množství detailů, ostré hrany a rychlý pohyb. Sekvence byla natočena kamerou s vysokým rozlišením a obraz je výrazně zašuměn. Na tuto filmovou sekvenci byly použity všechny varianty redukce šumu jádrováním: vlnková transformace bez použití 3D dekompozice, vlnková transformace s použitím 3D dekompozice, Simoncelliho pyramida bez 3D dekompozice a Simoncelliho pyramida s 3D dekompozicí.

V tabulce 7.1 jsou vidět hodnoty odstupe signálu od šumu (SNR) původní zašumělé sekvence a sekvence, na kterou byly použity jednotlivé algoritmy pro redukci šumu. Z tabulky je vidět, že nejlepší výsledků jsme dosáhli při použití 3D dekompozice Simoncelliho pyramidou, odstup signálu od šumu byl 37.8. Na obrázku 7.4 je vidět výsledek této redukce šumu.

Video sekvence	Sekvence obsahující šum (dB)	á trous prostorové Jádrování (dB)	á trous prostorové + časové Jádrování (dB)	Pyramid. prostorové Jádrování (dB)	Pyramidové prostorové+časové Jádrování (dB)
Plane 25	33.0	36.6	37.1	36.8	37.8

tabulka 7.1: Odstup signálu od šumu pro jednotlivé varianty algoritmu redukce šumu jádrováním [3]



obrázek 7.4: Ukázka ze sekvence Plane před a po redukci šumu [3]

8. Korekce nestability obrazu

Digitální stabilizace obrazu může být rozdělena podle [16] do tří kroků: predikce pohybu, detekce nežádoucího pohybu a kompenzace tohoto nechtěného pohybu. Detekce pohybu umožňuje určit pohyb mezi aktuálním snímkem a předchozím snímkem. Pohyb určuje na základě bloků libovolné velikosti (běžně používaná velikost je 16x16 pixelů) a vektorů pohybu, které nám zobrazují, kam se posunul blok oproti předchozímu snímku. Nevýhoda tohoto přístupu je, že všechny vektory pohybu nejsou stejně spolehlivé. Například blok, který byl vybrán na homogenní ploše obrazu, má poměrně často špatně určený vektor pohybu.

Prezentovaný algoritmus podle práce [16] pracuje se dvěma zónami a blokový detektor pohybu je použit v obou těchto zónách. Jedna zóna nám reprezentuje popředí obrazu a druhá pozadí. Empiricky bylo zjištěno, že v centrální oblasti obrazu se většinou pohybuje subjekt, který natáčíme a tato centrální oblast byla určena jako zóna popředí. Pro představu, film, ve kterém mluví postava. Takováto postava může hýbat rukama, otáčet hlavou, otevírat ústa atd. Proto v této zóně bude velké množství detekcí, které budou odpovídat lokálnímu pohybu v obrazu a ne globálnímu pohybu celého obrazu, tedy nestabilitě. Zóna pozadí obrazu bývá méně ovlivněna lokálními pohyby, ale často obsahuje homogenní zóny, ve kterých nejsou vektory pohybu určeny s dostatečnou jistotou. Na obrázku 8.1(a) vidíme bíle označené popředí obrazu a na obrázku 8.1(b) vidíme pozadí obrazu. Plocha a umístění pozadí obrazu byly také určeny empiricky. Pro správné určení globálního vektoru pohybu musíme tedy použít obě tyto zóny.



obrázek 8.1: (a) empiricky určené popředí obrazu (b) empiricky určené pozadí obrazu [16]

Je tedy třeba zjistit globální vektor pohybu. Ten je spočítán z blokových vektorů pohybu, které jsou vypočítány pro obě zóny. Aby byl odstraněn nedostatek této metody, který spočívá v detekci homogenního bloku na homogenním pozadí, je neprve použito váhování každého bloku pomocí průměrné absolutní odchylky intenzity jasu od průměru intenzity jasu v každém bloku (8.1). Čím bude tato hodnota MAD (Mean Absolute Difference) větší, tím bude blok obsahovat větší frekvence změny obrazu, tzn. určení jeho pohybu by mělo být přesné a bude mít proto velkou váhu pro určení globálního vektoru pohybu. Naopak bloky s malou hodnotou MAD budou úplně ignorovány. Přesné váhování vektorů na základě MAD je vidět v tabulce 8.1. Pro úplnost je zde prezentován obrázek 8.2, kde jsou jednotlivé bloky v zóně pozadí naváhovány pomocí MAD. Čím je blok světlejší, tím má větší váhu, čím je tmavší, tím má váhu nižší, pokud je blok černý, jeho váha je nulová.

$$MAD = \frac{1}{N^2} \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} |Y_{i,j} - Y_{average}| \quad (8.1)$$

MAD	Váha
MAD < 5	0
5 < MAD < 40	1
40 < MAD < 60	2
60 < MAD < 80	4
80 < MAD < 100	8
100 < MAD	16

tabulka 8.1: Váha bloku pro určení globálního vektoru pohybu podle MAD [16]

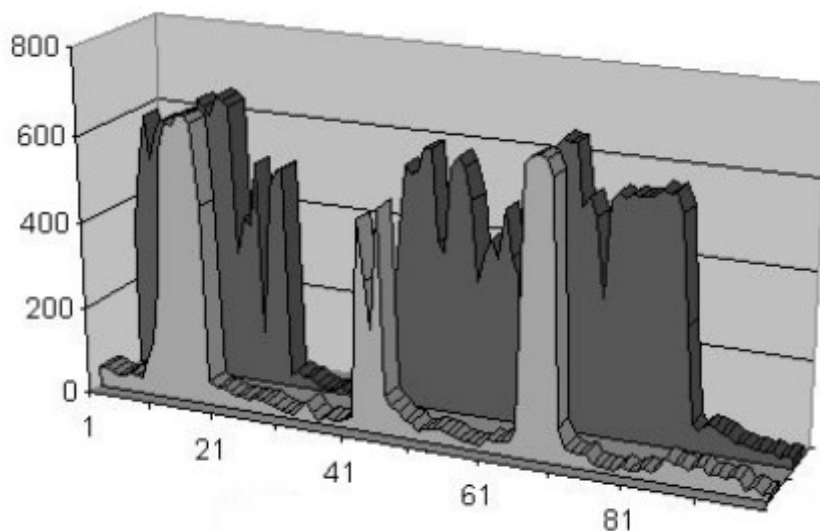


obrázek 8.2: Bloky ze zóny pozadí jsou naváhovány pomocí MAD, čím je blok světlejší, tím má vyšší váhu. Přední zóna je pouze vyznačena bílým obdélníkem, ale nemá jednotlivé bloky naváhovány

V každé zóně je určen nejčastěji se vyskytující blokový vektor pohybu. Je samozřejmě započítáno výše uvedené váhování. Pokud byla bloku přisouzena váha 4 pro vyhledání nejčastěji se vyskytujícího vektoru pohybu, počítá se vektor pohybu tohoto bloku, jako kdyby byly přítomny 4 bloky s tímto vektorem s váhou 1. Pokud se nejčastěji se vyskytující vektor pohybu v obou zónách shoduje, je proces ukončen, tento vektor pohybu je určen jako globální vektor pohybu a s jeho pomocí je provedena korekce nestability obrazu. Pokud se vektory pohybu neshodují, tzn. v každé zóně byl nejčastěji se vyskytující vektor jiný, potom je jako globální vektor pohybu určen ten z nich, který má větší váhu (tedy vektor, který byl přítomen vícekrát ve své zóně).

Pokud máme k dispozici globální vektor pohybu obrazu, který reprezentuje posunutí v obraze, ke kterému došlo vlivem nestability obrazu, oprava je jednoduchá. Stačí obraz posunout o tento globální vektor pohybu.

Výsledky této metody jsou velice dobré. Jsou dobře prezentovatelné grafem nulových vektorů pohybu (obrázek 8.3). Světleji jsou zobrazeny počty nulových blokových pohybových vektorů v jednotlivých snímcích před použitím korekce nestability obrazu. Černě jsou potom zobrazeny počty nulových blokových pohybových vektorů po aplikaci této metody pro korekci nestability obrazu.



obrázek 8.3: Počty nulových blokových vektorů pohybu ve filmové sekvenci před (bílé) a po (černé) provedení korekce nestability obrazu [16]

9. Digitální vytváření vad filmu pro potřeby objektivního testování

9.1 Úvod

Jak bylo uvedeno v kapitole 2, na historických filmech se nachází řada vad. Například skvrnová vada, vada škrábnutím, blikavý šum, šum a barevná degradace. Pro potřeby objektivního hodnocení kvality filmové sekvence po restauraci je třeba mít k dispozici filmovou stopu v původním neporušeném stavu. Takováto neporušená filmová sekvence ale bohužel většinou není k dispozici. Abychom otestovali objektivně efektivitu metod pro odstranění různých typů vad, je zapotřebí takovou stopu mít. Řešení je poměrně jednoduché. Použijeme neporušenou, vadami neposkvřněnou filmovou sekvenci a chyby do ní uměle vytvoříme. Tím se dostáváme do stavu, kdy máme neporušenou i vadami postiženou sekvenci k dispozici a můžeme objektivně testovat účinnost jednotlivých metod pro odstranění vad.

9.2 Vytváření jednotlivých vad

Všechny níže uvedené vady byly popsány podrobně v kapitole 2 a v navazujících kapitolách. V nich je uvedena nejen příčina vzniku vady, ale i jak se jednotlivé vady projevují a jaké jsou metody pro jejich odstranění. V této kapitole se budu věnovat vadám pouze z hlediska jejich umělého vytváření, založeného na práci [14].

Vada škrábnutím

Jak bylo uvedeno výše, vady škrábnutím jsou velice běžné na archivních filmech, a proto je nutné je věrohodně imitovat, aby se metody pro odstranění těchto vad daly efektivně testovat. Tato vada je pozorovatelná jako linie pixelů, kde se ztratila jasová informace. Při umělém vytváření takovéto vady se generuje linie černých nebo bílých pixelů pouze ve svislém směru, protože taková škrábnutí jsou nejčastější a jen málokdy se vyskytuje vodorovné škrábnutí. Délka škrábnutí je volena náhodně, protože v archivních filmech mohou být škrábnutí přes celý obraz i škrábnutí pouze přes část obrazu. Šířka škrábnutí je také náhodná a pohybuje se od dvou do šesti pixelů.

Skvrnová vada

Tato vada není uměle generována, jak tomu bylo v případě vady škrábnutím. Aby se dosáhlo maximálně věrohodných výsledků, jsou použity skutečné skvrny, které vznikly na jiných poškozených filmových sekvencích. Princip je jednoduchý: z jednotlivých obrazů ve filmových sekvencích, které obsahují skvrnové vady, jsou tyto obrázky extrahovány a umístěny na větší plochu. Pokud by obraz ve filmové sekvenci měl například velikost $M \times N$ pixelů, pak by plocha, na kterou tyto poškozené obrázky budeme umísťovat vedle sebe, mohla mít velikost $5M \times 5N$ pixelů. Je důležité si uvědomit, že vyjímáme pouze skvrny, ne celé obrázky. Ve výsledku tedy naše pomocná plocha vypadá jako $5M \times 5N$ pixelů velký, černobílý vzorník skvrnových vad.

Při generování skvrnové vady na nepoškozený obraz je tedy vygenerováno náhodné místo na naší ploše a od tohoto místa je vyjmut $M \times N$ pixelů velký obraz, který obsahuje pouze skvrnové vady. Ten je přičten k našemu nepoškozenému obrazu a tím získáme náhodné rozložení skutečných skvrnových chyb. Pokud bychom chtěli mít v obraze zároveň černé i bílé skvrny, stačí náhodně generovat barvu skvrny, kterou přidáváme do obrazu.

Prach

Prach na filmu je simulován přidáním šumu typu pepř a sůl (salt and pepper). Tento šum je náhodný a nezávislý na ostatních faktorech. Takováto vada se vyskytuje na archivních filmech poměrně málo a bude postihovat jen malé množství pixelů. Odstranit by ji měly metody, které jsou určeny pro odstranění skvrnové vady, protože se projevuje podobně, jen s odlišnou distribucí.

Blikavý šum

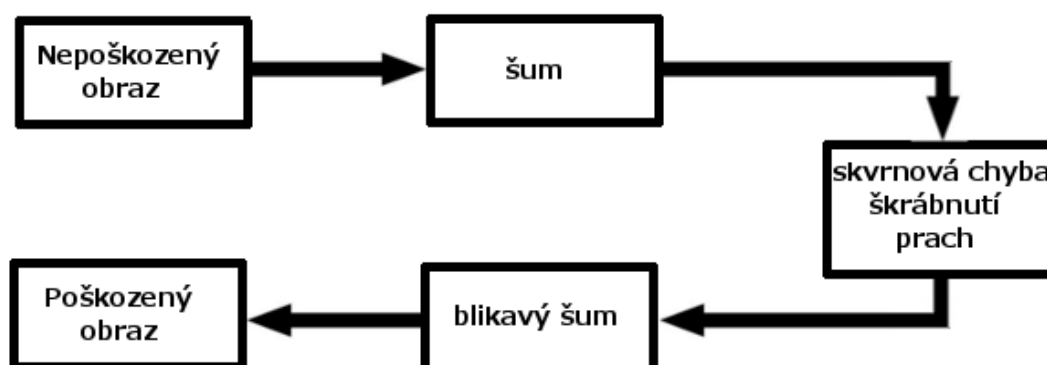
Vytvoření blikavého šumu může být realizováno dvěma způsoby. Je možné manipulovat s jednotlivými obrazy ve filmové sekvenci a zvyšovat či snižovat kontrast pomocí nástrojů, které to umožňují, jako například Photoshop či GIMP. Druhá možnost je extrahovat blikavý šum z filmové sekvence, která jej obsahuje a aplikovat ho na nepoškozenou sekvenci. První případ nám umožňuje blikavý šum generovat na místech, kde ve filmu dává skutečně smysl, například v místech, kde vychází slunce. Přednost druhého případu je, že se jedná o skutečný blikavý šum a ne uměle vytvořený.

Šum

Pro simulaci šumu je použita velmi podobná metoda jako pro generování skvrnových chyb. Nejprve je vytvořena textura obsahující pouze šum. Tato textura je výrazně větší než je velikost jednotlivých obrazů ve filmové sekvenci. Pro přidání šumu do obrazu je na této textuře náhodně vybrána plocha o velikosti shodné s velikostí obrazu a je aplikována na jeden obraz ve filmové sekvenci.

9.3 Aplikování vad na filmovou sekvenci

Abychom dosáhli správných výsledků a generovali poškozené obrazy tak, aby skutečně odpovídaly reálným poškozením, je zapotřebí jednotlivé vady aplikovat postupně a ve správném pořadí. To je uvedeno na obrázku 9.1.



obrázek 9.1: Postupné generování vad do filmové sekvence

9.4 Výsledky aplikování vad na nepoškozenou filmovou sekvenci

Pro představu, jak věrohodně mohou uměle vytvořené vady vypadat, si dovoluji uvést příklad. Všechny vady, které obraz obsahuje, byly vytvořeny uměle, podle návodu v kapitole 9.2 a 9.3. Na obraz je aplikována skvrnová vada, vada poškrábání a šum. Obraz z nepoškozené filmové sekvence je možné vidět na obrázku 9.2 (a) a obraz po aplikování výše uvedených vad je vidět na obrázku 9.2(b).



obrázek 9.2(a): Obraz z nepoškozené filmové sekvence [14]



obrázek 9.2(b): Obraz s uměle vytvořenými vadami [14]

10. Subjektivní testy

I přes pokročilý vývoj objektivních testů, kterým je v současné době věnováno hodně pozornosti, zůstávají subjektivní testy rozhodující a mají věrohodnější výsledky než testy objektivní. Proto se při vývoji objektivních testů využívají výsledky subjektivních testů jako reference pro vyhodnocení kvality objektivní metody. Na druhou stranu velkou nevýhodou subjektivních testů je finanční a časová náročnost. Další nevýhodou je, že subjektivní testy můžeme dělat jen v určitých časových intervalech a pozorovatele můžeme zatížit jen určitým množstvím promítaných filmových sekvencí. Při návrhu metod pro odstranění vad z historických filmů ale potřebujeme výsledky testů často a rychle.

Níže jsou popsány jednotlivé metody [19] subjektivních testů z hlediska metodiky promítání a hodnocení perceptuálního dojmu. Důležité je ovšem zaznamenat i další parametry než jen ty, které jsou výstupem těchto metod. Je třeba zaznamenat pozice hodnotitelů před plátnem, zasedací pořádek. Také by mělo být zaznamenáno, kde a kdy se subjektivní test prováděl. Test by měl podléhat přísně nastaveným podmínkám vzdálenosti pozorovatelů od plátna, osvětlení místnosti, celkové doby provádění testu a výběru pozorovatelů.

I. Test celkové kvality s dvojitým podnětem

V tomto testu je referenční a upravené video přehráno dvakrát. Pořadí videí je určeno náhodně tak, aby pozorovatelé nevěděli, které video je referenční a které upravené. Lze tedy promítat video například v pořadí: referenční, upravené, referenční, upravené. Na konci promítání jsou pozorovatelé požádáni, aby ohodnotili celkovou kvalitu videa na stupnici od 0 do 100 (0 představuje nejhorší a 100 nejlepší výsledek). Je použito několik párů referenčního a upraveného videa o délce přibližně 10 vteřin.

II. Test vad s dvojitým podnětem

Narozdíl od testu celkové kvality s dvojitým podnětem jsou při tomto testu pozorovatelé seznámeni s tím, na kterou sekvenci se zrovna dívají a každá sekvence je promítána pouze jednou. Referenční video je promítáno jako první a je následováno promítáním upravené filmové sekvence. Pozorovatelé hodnotí filmové sekvence pouze pěti možnými slovními ohodnoceními. Nejhorší hodnocení je „velice rušivé vady“, nejlepší hodnocení je „nepostřehnutelný rozdíl“.

III. Test celkové kvality s jedním podnětem

Pozorovatelům je promítnuta pouze upravená filmová sekvence. Promítá se výrazně delší dobu než v předchozích testech, obvykle 20-30 minut. Pozorovatelé mají k dispozici stupnici, kde nejhorší hodnocení je „špatné“, nejlepší hodnocení je „výborné“. Na této stupnici je jezdec (ukazatel na stupnici), kterého posouvají podle aktuálního dojmu z filmu.

IV. Absolutní hodnocení

Toto je podobná metoda jako „Test celkové kvality s jedním podnětem“. Uživatelům je také ukázána pouze upravená videosekvence. Nehodnotí ji však průběžně, ale až na úplném konci promítání. Stupnice má pět hodnot od „špatná kvalita“ po „výborná kvalita“.

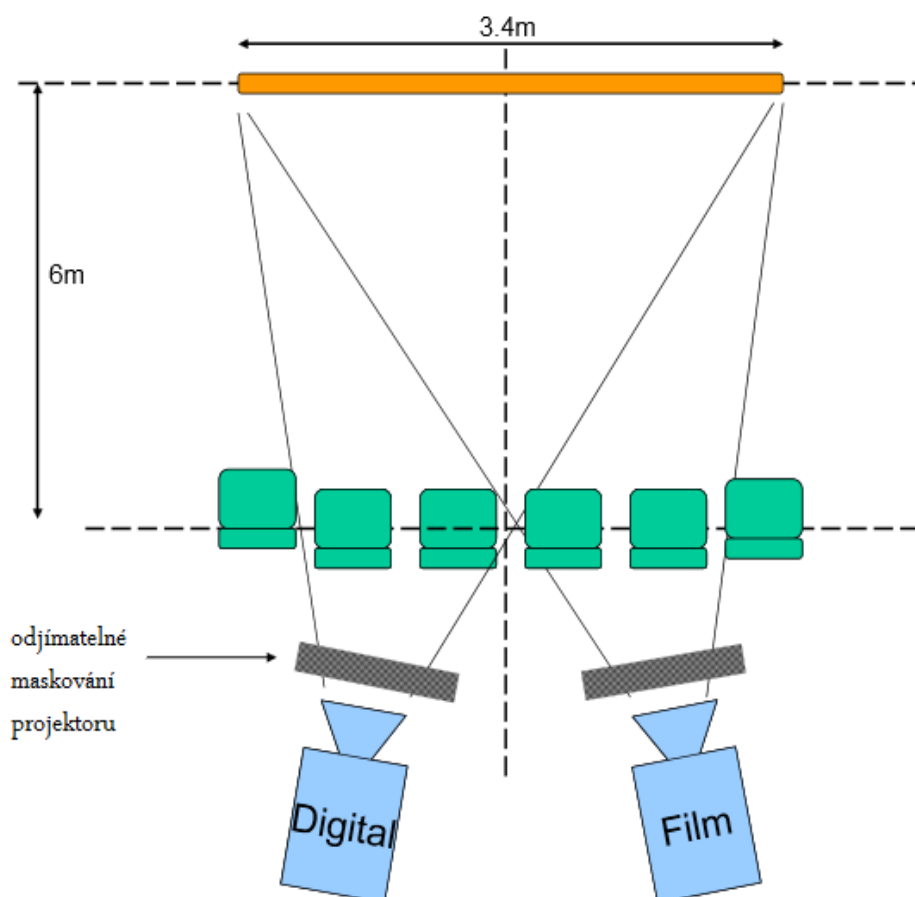
V. Párové porovnání

Tato metoda je poměrně jednoduchá a přesto nám poskytuje dobré výsledky, obzvláště v oboru restaurace filmů. Párujeme jednotlivé snímky z referenční filmové sekvence a upravené videosekvence. Pozorovatelé provedou porovnání snímků a vyjádří se k němu v několika větách.

10.1 Subjektivní testy pro porovnání digitální a analogové filmové projekce v současné praxi

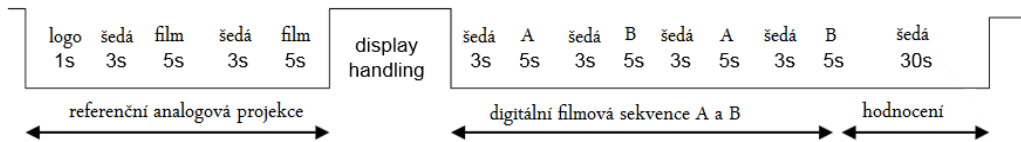
10.1.1 Test celkové kvality s dvojitým podnětem

Pro porovnání analogové a digitalizované filmové projekce je zapotřebí vhodně připravit místnost. Rozestavení pozorovatelů a promítacích zařízení je vidět v obrázku 10.1, které je navrženo v práci [11]. Promítací plocha je zde široká 3,4 metru a vzdálenost pozorovatelů od promítací plochy je 6 metrů. Pozorovatelé sedí v jedné řadě vedle sebe. Ideální počet pozorovatelů pro jednu projekci je 4-6. Test probíhá tak, že je nejprve promítnuta originální analogová filmová sekvence a poté jsou promítnuty dvě verze digitální filmové sekvence (verze se označují A a B). Běžně verze označená jako A reprezentuje pouze zdigitalizovanou verzi analogového filmu a ve verzi B jsou libovolné digitálně dodělané úpravy.



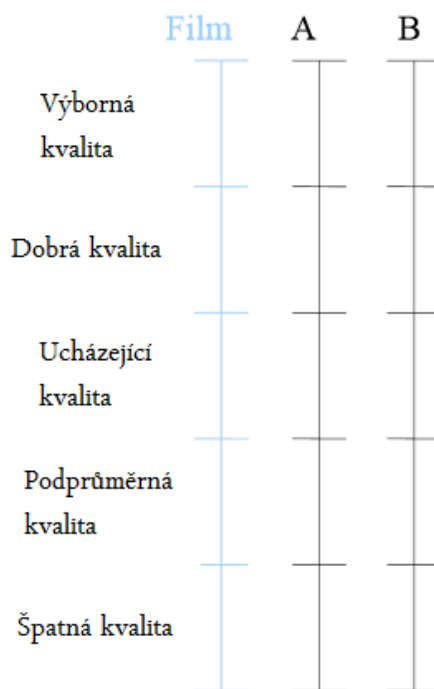
obrázek 10.1: Rozestavení pozorovatelů a promítacích zařízení pro subjektivní test celkové kvality s dvojitým podnětem [11]

Postup při promítání je následující: nejprve je promítnuto logo nebo jakýkoliv jiný symbol, který upozorní pozorovatele na začátek testu, poté se přehraje 10-ti vteřinový úsek z originální analogové filmové sekvence. Tento úsek je rozdělen na dva 5-ti vteřinové úseky a je proložen 3-mi vteřinami trvající šedou pauzou. Poté je zamaskován analogový projektor a je promítáno z digitálního projektoru. Nejprve je přehráno 5 vteřin ze stopy A, poté 5 vteřin ze stopy B. Opět následuje 3 vteřinová šedá pauza a následuje druhá část 10 vteřinového úseku z obou stop. Po promítnutí obou stop je nechána 30 vteřinová pauza, ve které pozorovatelé mohou ohodnotit kvalitu filmových sekvencí. Vše je vidět na obrázku 10.2.



obrázek 10.2: Postup při promítání filmových sekvencí při subjektivním testu celkové kvality s dvojím podnětem [11]

Pozorovatelé jsou požádáni o hodnocení filmové sekvence A a B na stupnici od „špatná kvalita“ po „výborná kvalita“. Toto hodnocení se zaměřuje na několik vlastností obrazu a pro každou tuto vlastnost je určen jeden dotazník, ve kterém pozorovatel označí odpovídající úroveň kvality. Tyto vlastnosti jsou barva, jas a kontrast. Ukázka dotazníku je na obrázku 10.3. Důležitá je také poznámka, že pozorovatelé smějí hodnotit pouze v čase určeném pro hodnocení.

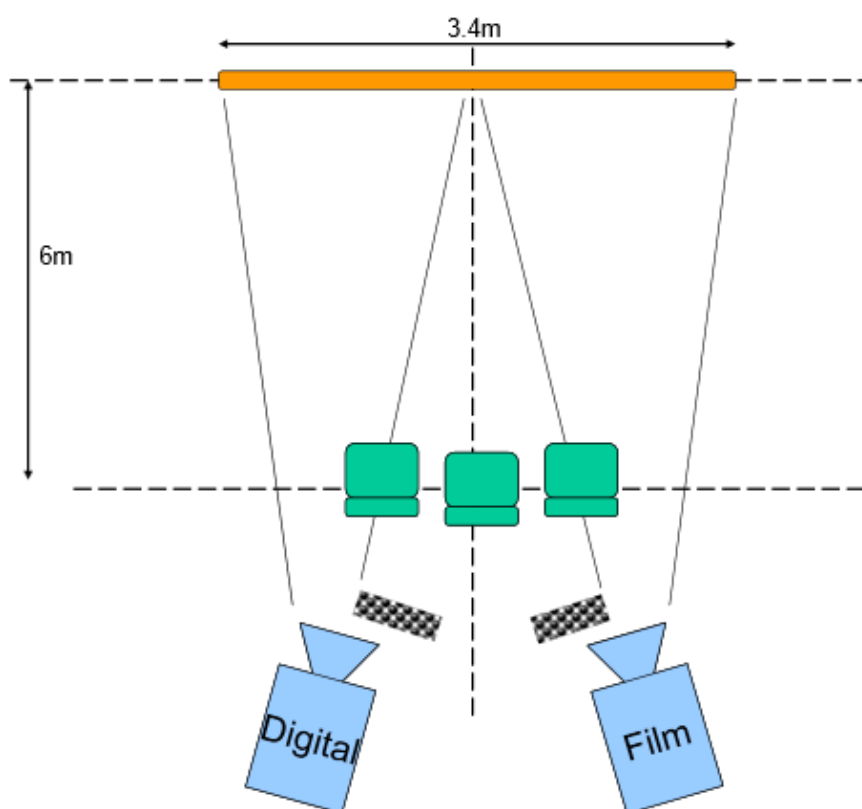


obrázek 10.3: Příklad dotazníku pro hodnocení kvality při testu celkové kvality s dvojím podnětem [11]

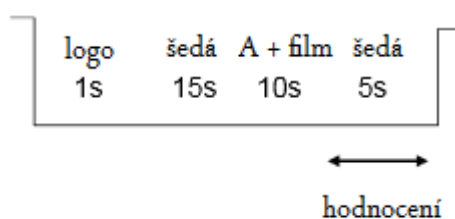
10.1.2 Test párového porovnání

Druhý test, který se provádí pro hodnocení rozdílu mezi digitální verzí a analogovou verzí filmu, je založený na promítání dvou stálých obrazů vedle sebe a jejich porovnání. (v předchozí kapitole byl tento test označen jako V. Párové porovnání). Tento test je prováděn pouze se třemi pozorovateli a jejich rozestavení je vidět v obrázku 10.4.

Nejprve je zapotřebí vybrat stálé obrazy, které se budou promítat. Tyto obrazy jsou vždy zaměřeny na určitý parametr filmové projekce. Mohou to být například různé druhy vad, které jsou uvedeny v kapitole 2, ale také to mohou být parametry, jako například barva kůže, saturace modré oblohy atd. Při promítání je nejprve zobrazeno logo, aby účastníci testu věděli, že test začíná a začali se soustředit. Poté následuje 15-ti vteřinová pauza, při které je zobrazena pouze šedá plocha, aby se uživatelé mohli přizpůsobit úrovni osvětlení v místnosti. Poté je 10 vteřin současně zobrazen snímek z referenčního analogového videa a snímek z digitální stopy A nebo B. Po promítnutí dvou statických snímků dostanou uživatelé 5 vteřin na hodnocení.



obrázek 10.4: Rozestavení pozorovatelů a promítacích zařízení pro subjektivní test párového porovnání [11]

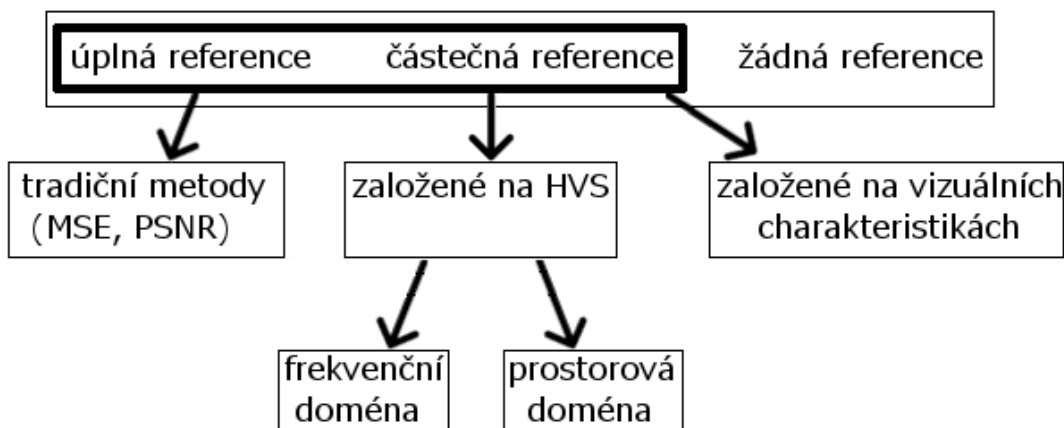


obrázek 10.5: Postup při promítání snímků při testu párového porovnání [11]

11. Objektivní testy

Objektivní testy [19] jsou důležité pro hodnocení kvality metod pro odstraňování vad v historických filmech a jiných poškozených filmových sekvencích. Tyto metody se dynamicky vyvíjejí a je třeba je testovat po každé úpravě. Také je třeba u jednotlivých metod testovat, jak efektivně je zvoleno nastavení parametrů (například prahových hodnot). Takovéto měření je třeba udělat i několikrát denně, pro což jsou subjektivní testy, i přes jejich dobré výsledky, naprosto nevhodné.

Objektivních testů je několik základních druhů. První kategorizací je rozdělení na testy s úplnou referencí, částečnou referencí a žádnou referencí (obrázek 11.1). Objektivní testy bez reference nedosahují tak dobrých výsledků, a proto se jimi v této práci nebudu zabývat. Dále se testy dělí na tradiční, založené na HVS (Human Visual System) a na testy založené na vizuálních charakteristikách. Tradiční metody jsou například MSE (Mean square error), SNR (Signal to Noise Ratio), PSNR (Peak Signal to Noise Ratio). Výsledky těchto metod jsou poměrně dost odlišné od výsledků subjektivních testů, což je špatné. Na druhou stranu jsou ale velice rychlé jak na implementaci, tak na časovou náročnost samotného testu. Proto se stále s oblibou používají jako první indikátor kvality filmové sekvence. Pro výsledky, které by se více shodovaly se subjektivními testy, jsou vyvíjeny metody založené na parametrech Lidského zrakového systému (z anglického HVS). Tyto metody se snaží hodnotit chyby, které ve filmové sekvenci jsou, podle toho, jak moc jsou rušivé pro lidský zrakový systém. Například výkyvy v hodnotě jasu vnímá lidské oko daleko více než výkyvy v hodnotě barvy. Lidské oko je také velice citlivé na narušení struktur předmětů, které se v obrazu nacházejí. Pokud je v obrazu rychlý pohyb, je oko daleko méně citlivé na rozmazání obrazu, než pokud je obraz statický. Na těchto a dalších podobných charakteristikách se snaží metody založené na lidském zrakovém systému stavět.



obrázek 11.1: Rozdělení objektivních testů [19]

11.1 MSE (Mean Square Error)

MSE je jeden z jednoduchých testů kvality obrazu. Je spočítán jako podíl druhé mocniny rozdílu v obou obrazech s plochou obrazu nebo její části, ze které počítáme MSE (typicky 8x8, 16x16). Tento ukazatel není nevhodnější, ale je velice jednoduchý a rychle spočítatelný. Ve vzorci (11.1) reprezentuje $f(i,j)$ originální obraz, zatímco $F(i,j)$ reprezentuje obraz po rekonstrukci.

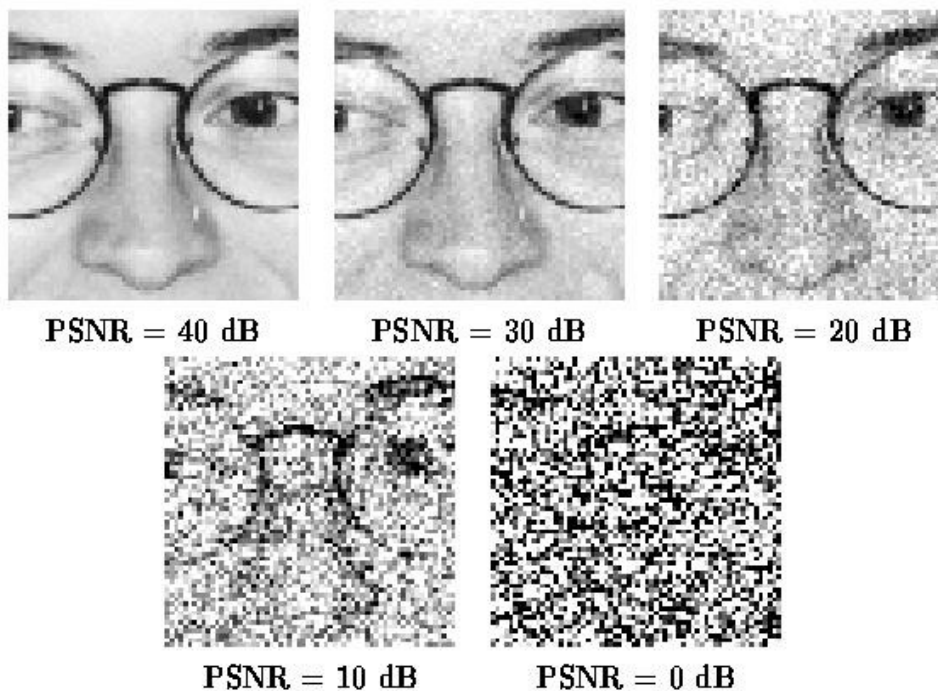
$$MSE = \frac{\sum_{i=1}^M \sum_{j=1}^N [f(i,j) - F(i,j)]^2}{M \cdot N} \quad (11.1)$$

11.2 PSNR (Peak Signal to Noise Ratio)

PSNR je získáno pomocí MSE (mean square error). Hodnota MSE je vztažena k největší možné hodnotě jasu. Typicky pro 8 bitovou hodnotu je nejvyšší úroveň jasu $2^8 - 1 = 255$. Výsledná hodnota je v decibelech a běžně se pohybuje od 25 do 40 decibelů, podle kvality videa. Ukázka takového ohodnocení kvality obrazu je vidět na obrázku 11.2.

$$PSNR = 20 \cdot \log_{10} \left(\frac{255}{\sqrt{MSE}} \right) \quad (11.2)$$

I přesto, že je v současnosti velké množství metod, jejichž výsledky jsou věrohodnější a více se blíží výsledkům subjektivních testů, PSNR a MSE stále zůstávají jedněmi z nejoblíbenějších testů, a to pro svoji jednoduchost a rychlost.



obrázek 11.2: Hodnocení kvality videa pomocí špičkového odstupu signálu k šumu (PSNR) [18]

11.3 Metoda využívající hodnocení narušení struktur v obraze

Metody, které byly uvedeny výše (MSE a PSNR), hodnotí kvalitu obrazu podle chyb, které se v něm nacházejí. Jakákoliv degradace obrazu je v těchto metodách vnímána jako chyba. Ale neplatí vždy, že velká chyba v obraze znamená výrazně zhoršený vizuální dojem z filmové sekvence. Metoda, která je uvedena v této kapitole [13], patří do kategorie metod založených na lidském zrakovém systému a využívá toho, že lidské oko se snaží získat strukturální informaci ze svého zorného pole. Proto je měření narušení těchto struktur dobrým ukazatelem pro to, jak kvalitní se obraz zdá lidskému oku.

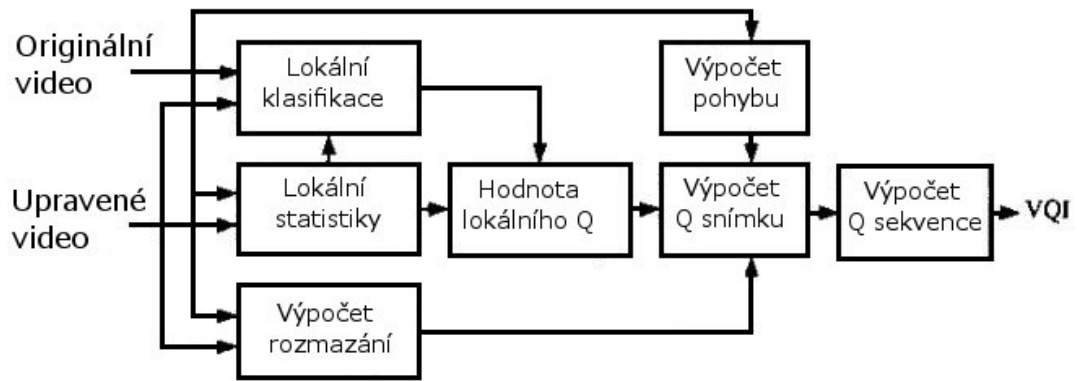
Na tomto principu (SSIM [13]) je založeno větší množství metod. Zde uvedu jednu z nich, kterou jsem zvolil pro její poměrnou jednoduchost a která přesto dosahuje jednoho z nejlepších výsledků. Nechť $x=\{x_i|i=1,2,\dots,N\}$ představuje snímky z originální videosekvence (nepoškozené) a $y=\{y_i|i=1,2,\dots,N\}$ představuje snímky z poškozené filmové sekvence. Index kvality videa je potom spočítán takto (11.3):

$$Q = \frac{4\sigma_{xy}}{(\sigma_x^2 + \sigma_y^2)[(x)^2 + (y)^2]} \quad (11.3)$$

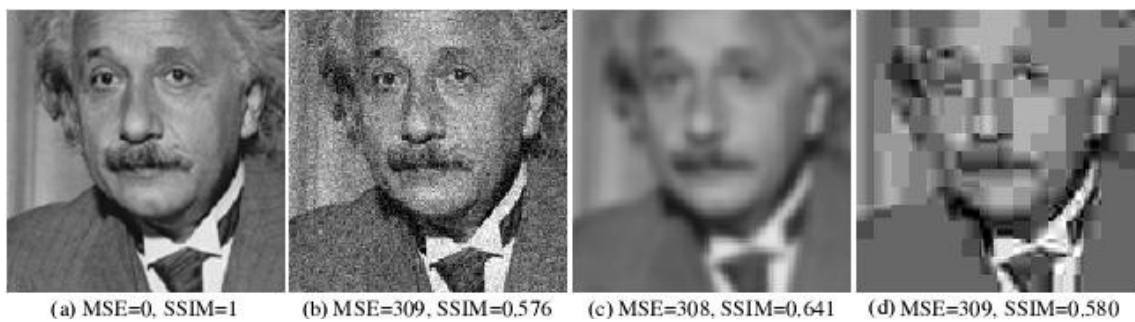
V této rovnici x a y představují střední hodnotu x a y . $(\sigma_x)^2$ a $(\sigma_y)^2$ představuje varianci (umocněný průměr rozdílu od střední hodnoty). $(\sigma_{xy})^2$ reprezentuje kovarianci x a y . Nejlepší výsledek, kterého lze dosáhnout, je, pokud je index $Q=1$. Toho je dosaženo pouze tehdy, když jsou videosekvence totožné. Rozsah, ve kterém se Q pohybuje, je $\{-1,1\}$.

V obrázku 11.3 je uveden diagram této metody. Kvalita videa je nejprve měřena na jednotlivých snímcích. Pro každý snímek je náhodně vybráno několik 8×8 oblastí, pro které je spočítán koeficient Q . To umožňuje výrazně zrychlit vyhodnocení kvality. Pokud bychom chtěli dosáhnout nejvěrohodnějšího výsledku, je možné vypočítat index Q pro celý obraz, ale výpočetní náročnost bude výrazně vyšší. V každé této 8×8 pixelů veliké zóně jsou spočítány lokální charakteristiky, jako jsou střední hodnota, variance a kovariance. Pomocí těchto ukazatelů je spočítán index Q . Všechny indexy Q jsou průměrovány tak, abychom dostali index Q , který hodnotí celý snímek. Tato hodnota je ještě upravena pro snímky, ve kterých dochází k velkému pohybu oproti předchozím a následujícím snímkům. U takových snímků je spočítáno rozmazání obrazu. To je získáno pomocí frekvenčního spektra, kde při rozmazání obrazu můžeme vidět posun z vysokých frekvencí do nízkých. Protože lidské oko není tolik citlivé na rozmazání obrazu při rychlém pohybu, je tento faktor brán v úvahu i při počítání výsledného indexu Q . Nakonec se indexy všech snímků zprůměrují a získáme výsledný index Q , který hodnotí kvalitu celé videosekvence.

Velké množství historických videosekvencí je černobílých a pro ty stačí uplatnit postup uvedený výše. Pro barevné filmy je zapotřebí pracovat se třemi složkami. Barevný snímek nám reprezentují složky Y , Cr a Cb . Stejný algoritmus je aplikován na každou z těchto složek a výsledky jsou průměrovány. Průměr je ovšem váhovaný tak, aby odpovídal vnímání lidského zrakového systému, tomu odpovídají váhy 0,7 pro Y , 0,15 pro Cr a Cb . Výsledek této metody porovnané oproti MSE je vidět na obrázku 11.4.



obrázek 11.3: Postup výpočtu indexu kvality Q pro celou videosekvenci [13]



obrázek 11.4: Porovnání MSE a SSIM (a) Originální snímek; (b) Snímek postižený Gausovským šumem; (c) rozmazaný snímek; (d) snímek ve formátu jpeg (nastavení kvality snímku 0,58) [20]

12. Implementace

12.1 Implementace detekce a odstranění skvrnové chyby

Jednou z běžných vad starých filmů jsou skvrnové chyby, v anglické literatuře označované jako „blotch“ chyby. Tyto skvrny vznikají obvykle kvůli ztrátě povrchu na originálním filmu nebo kvůli elektrostatickým přilnavým materiálům, jako jsou prach, vlasy atd., které nemohly být odstraněny fyzicky z filmu před digitalizací. V digitálním záznamu vypadají jako černé nebo bílé skupiny pixelů. Tyto vadné skupiny pixelů mají různou velikost, tvar a pozici.

Naimplementoval jsem tři algoritmy pro detekci skvrnové chyby, jmenovitě SDIa, sROD a adaptivní test. Pro odstranění všech detekovaných skvrn byl pro všechny jmenované detekční metody použit stejný algoritmus, aby bylo možné výsledky těchto metod snadno vizuálně porovnat (viz porovnání výsledků v kapitole 12.1.5).

12.1.1 Implementace detekce skvrnové chyby pomocí algoritmu SDIa a odstranění chyby pomocí algoritmu s přesahem

Pro detekci této chyby jsou v mé práci v kapitole 3.2.1 uvedeny tři druhy algoritmů. Zde se budu věnovat implementaci algoritmu pojmenovaného jako SDIa. Tento test je založen na rozdílu v přilehlých snímcích (12.1):

$$b(x) = \min(|d^-(x)|, |d^+(x)|) \cdot \text{sign}[d^-(x) \cdot d^+(x)] \quad (12.1)$$

Poté je manuálně zvolena prahová hodnota, proti které porovnávané jednotlivé pixely v rozdílových obrazech. Pokud byl práh překročen, vyhodnotí algoritmus pixel jako vadný.

Jak bylo řečeno výše, tento test je založen na rozdílu dvou přilehlých snímků k aktuálnímu snímku ve filmové sekvenci. To se pro moji testovací stopu, pojmenovanou city.avi, ukázalo jako největší nedostatek. Velká část (přibližně 50%) všech přítomných skvrnových chyb byla přes dva snímky. Jak je vidět ze vzorce, je aplikován pouze menší z obou rozdílů. Proto nebyly tyto skvrny detekovány.

Tato skutečnost byla důvodem pro první úpravu algoritmu, kterou bylo nutné implementovat. Bylo třeba projít jednotlivé snímky v obraze a pro ty, ve kterých se nacházela skvrnová chyba přes více než jeden snímek, upravit algoritmus na (12.2), kde d^{--} a d^{++} představují rozdíl současného snímku oproti snímkům, které jsou vzdálené o 2 snímky.

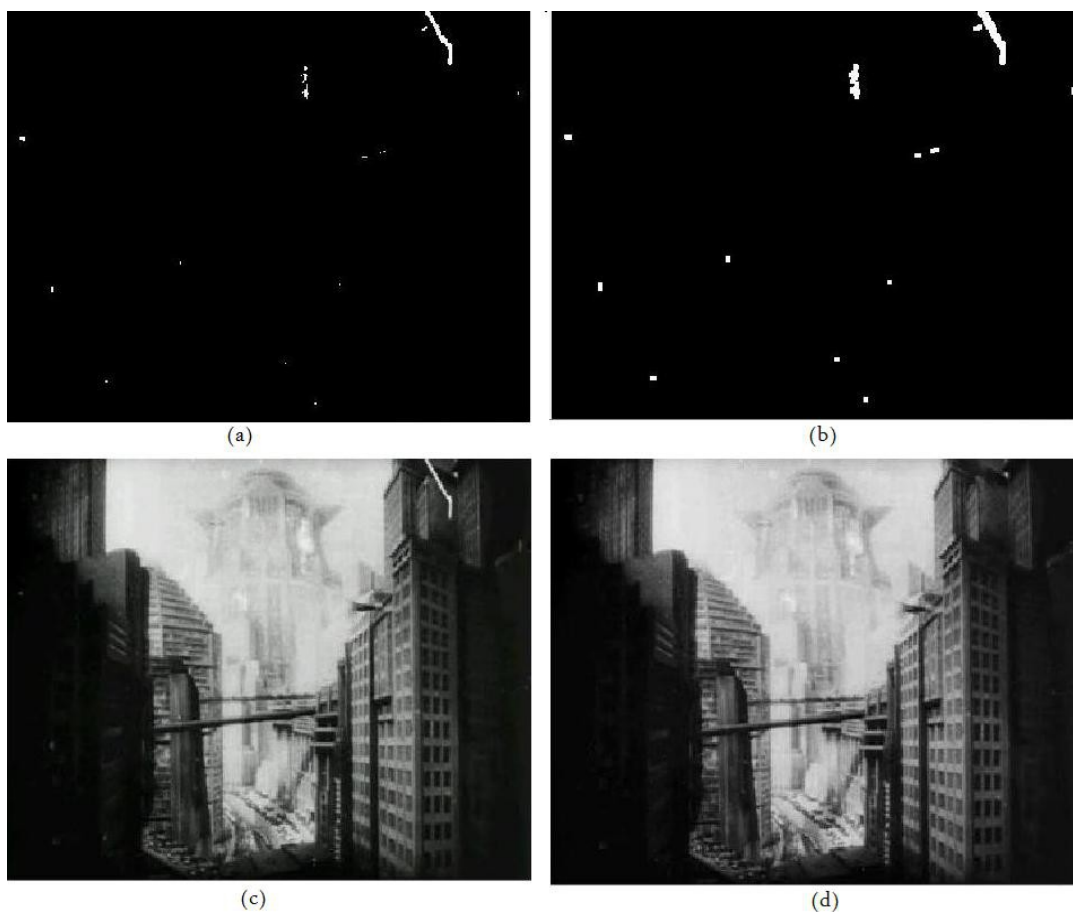
$$b(x) = \min(|d^{--}(x)|, |d^{++}(x)|) \cdot \text{sign}[d^{--}(x) \cdot d^{++}(x)] \quad (12.2)$$

Po takovéto detekci rozdílem je aplikovaný práh, který rozhoduje o chybě. Z principu by se dal práh považovat za globální pro celou sekvenci. Do algoritmu se dají ovšem jednoduše vložit podmínky, které umožní prahovou hodnotu měnit pro jednotlivé snímky v sekvenci. Toho jsem využíval, abych omezil počet falešných detekcí, ale zároveň detekoval všechny chyby. Je přirozené, že bílé skvrny na světlém pozadí potřebovaly výrazně nižší práh (vyšší citlivost), aby byly detekovány. V jednom případě jsem musel dokonce algoritmus upravit tak, že práh byl v části obrazu citlivější (nižší),

aby došlo k potřebným detekcím a ve zbytku obrazu musela být prahová hodnota vyšší, protože jinak docházelo k takovému počtu falešných detekcí, které po aplikování algoritmu na opravu skvrnové chyby výrazně znehodnotily obraz.

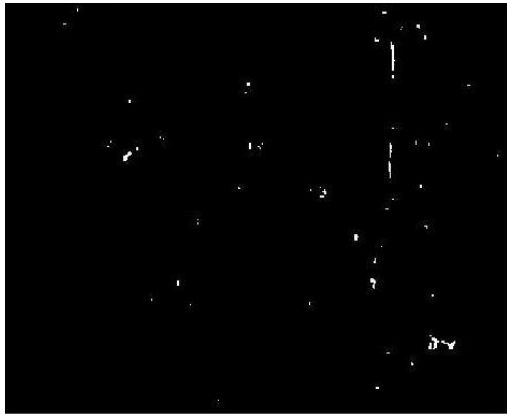
Výsledky

Výsledky naimplementovaného algoritmu na testované videosekvenci „city.avi“¹ prezentuji na několika snímcích níže. V levém horním rohu (a) se vždy nachází detekce skvrnové chyby. Vedle ní je obrázek prezentující rozšíření detekce této chyby o 1 pixel (b). V dolní části obrázku potom můžeme vidět originální (c) a opravenou sekvenci (d).



obrázek 12.1: Sekvence city.avi, snímek 2

¹ Zdroj: http://diamant.joanneum.ac.at/film_restoration/



(a)



(b)



(c)



(d)

obrázek 12.2: Sekvence city.avi, snímek 63



(a)



(b)



(c)

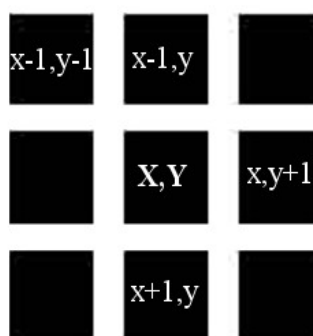


(d)

obrázek 12.3: Sekvence city.avi, snímek82

12.1.2 Implementace detekce skvrnové chyby pomocí algoritmu sROD

Druhý přístup je postaven na statistických údajích spočítaných z přilehlých pixelů k x^+ a x^- . V kapitole 3.2 bylo vysvětleno, že test sROD je založen na získání hodnot $(r1, r2, r3)$ a $(r4, r5, r6)$ jako reprezentantů stupňů šedi ve třech pixelech přilehlých x^+ respektive x^- . Bod x je detekován jako vadný, pokud jsou splněny následující podmínky: $u(x) - \max(r1, \dots, r6) > S$ nebo $\min(r1, \dots, r6) - u(x) > S$, kde S je námi nastavená pozitivní konstanta. Po provedení několika testů jsem se rozhodl změnit počet přilehlých pixelů pro sROD test. Lepších výsledků jsem dosahoval, pokud jsem počítal se čtyřmi přilehlými pixely namísto tří. Výpočetní složitost nebyla touto změnou viditelně ovlivněna. Volbu pixelů jsem provedl empiricky a nakonec jsem vybral rozložení uvedené na obrázku 12.4. Stejnými testy jsem určil prahovou hodnotu $S=15$.



obrázek 12.4: Výběr přilehlých pixelů pro sROD detekci

Abych dosáhl dobrých výsledků, bylo třeba, podobně jako při detekci SDIa, prahovou hodnotu měnit pro různé snímky. Tím tato metoda (stejně jako metoda SDIa) ale přichází o důležitou vlastnost automatické detekce, nezávislé na videosekvenci. Mezi nesporné výhody této metody ale určitě patří jednoduchá implementace a malá výpočetní náročnost algoritmu.

12.1.3 Implementace detekce skvrnové chyby pomocí adaptivního algoritmu

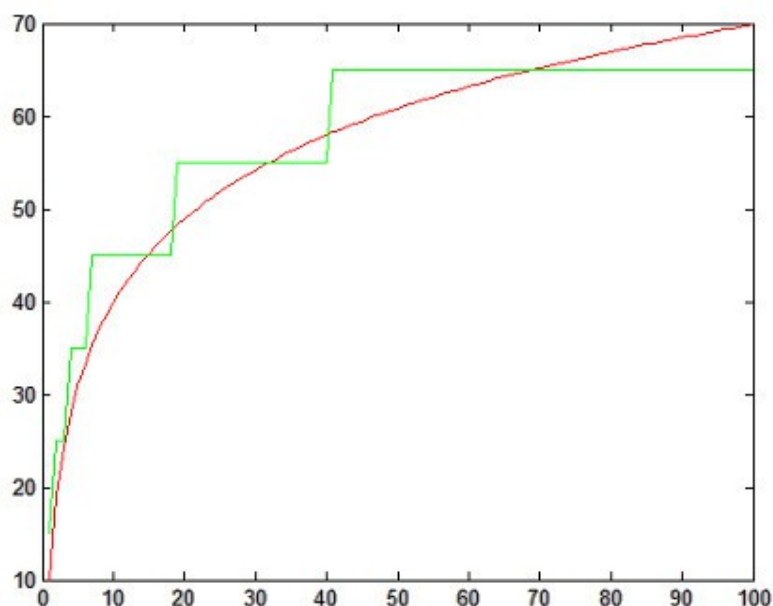
Detekce skvrnové chyby pomocí adaptivního algoritmu byla detailně popsána v kapitole 3.2. Algoritmus byl naimplementován podle předlohy v teoretickém rozboru. Jedinou změnou, kterou jsem v algoritmu udělal, byl výpočet adaptivní prahové hodnoty. Při testech na dvou 10 vteřinových videosekvencích z filmu "Street Scene" jsem totiž při původním výpočtu prahové hodnoty získával poměrně velké množství falešných detekcí. Tento problém jsem obešel jednoduše. Zůstal jsem u myšlenky, že práh by se měl adaptovat podle standardní odchylky. Čím je v testovaném bloku větší standardní odchylka, tím může být práh vyšší, protože skvrna by měla výrazně rozdílnou hodnotu jasu (barvu) než její okolí. Naopak pokud je standardní odchylka malá, nastaví se prahová hodnota nízko, aby bylo možno detekovat i skvrnu, která má jen mírně rozdílnou intenzitu jasu než okolí. Tato vlastnost je velice důležitá, protože obě testované sekvence měly velké množství bílých skvrn na světlém pozadí. Nejprve jsem tedy vytvořil jednoduchou, nespojitou funkci podle tabulky 12.1, která se dala určit provedením několika testů na videosekvencích. Při aplikaci tohoto nastavování prahové hodnoty se výsledky detekce zlepšily, bylo méně falešných detekcí. Proto jsem se rozhodl pokusit se najít funkci, která by tuto skokovou funkci co nejlépe aproximovala

tak ,aby byla odstraněna nespojitost funkce. Takovou funkci jsem skutečně našel (12.3). Pro představu, jak jsou tyto dvě funkce pro určení prahové hodnoty podobné, uvádím obrázek 12.5. Při použití funkce (12.3) jsem dosáhl nejlepších výsledků detekce a používal jsem ji pro všechny testované videosekvence.

Standardní odchylka	prahová hodnota
$0 < s < 0,75$	15
$0 < s < 3$	25
$3 < s < 6$	35
$6 < s < 18$	45
$18 < s < 40$	55
$40 < s$	65

tabulka 12.1: Funkce pro určení prahu podle standardní odchylky

$$\text{prahová hodnota} = \frac{\ln(\sigma)}{\ln(1,08)} + 10 \quad (12.3)$$



obrázek 12.5: Červená křivka je funkce určená tabulkou 12.1, zelená křivka představuje funkci (12.3)

12.1.4 Implementace odstranění skvrnové vady pomocí algoritmu s přesahem

Tento algoritmus byl popsán v kapitole 3.3 a jeho důležitou charakterovou vlastností je omezení okrajových artefaktů, které by mohly vznikat v některých případech opravení skvrnové chyby.

Protože v testovaných sekvencích mělo několik skvrn rozmazané hrany, tzn., že na hranách nebyla jasová informace úplně ztracena, docházelo k detekcím, kdy byl detekován pouze „vnitřek“ skvrnové chyby. To vedlo k velice nepříjemným okrajovým artefaktům po odstranění, protože obrys původní chyby byl stále viditelný. V některých případech tento nedostatek vedl k neodstranění skvrny. Při použití přesahu o pouhý jeden pixel, tak jak bylo navrhováno v kapitole 3.3, došlo k odstranění tohoto nedostatku.

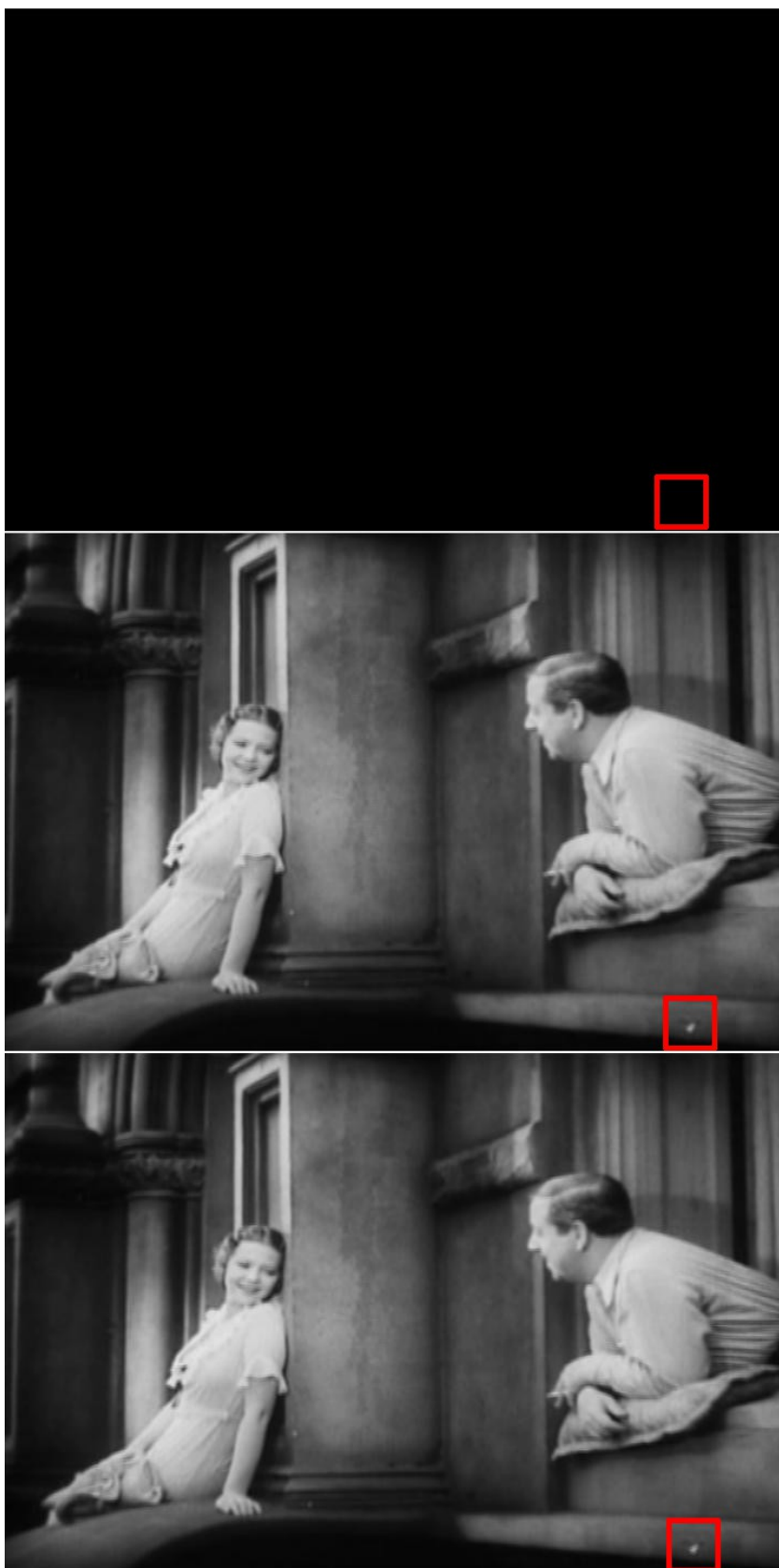
Tento algoritmus tak, jak byl uveden v kapitole 3.3, pracuje s mediánovým filtrem. Zkoušel jsem aplikovat jak mediánový, tak průměrující filtr a s aritmetickým průměrem jsem dosahoval lepšího vizuálního výsledku. Proto jsem ho v metodě aplikoval namísto mediánového filtru.

12.1.5 Porovnání jednotlivých typů detekce

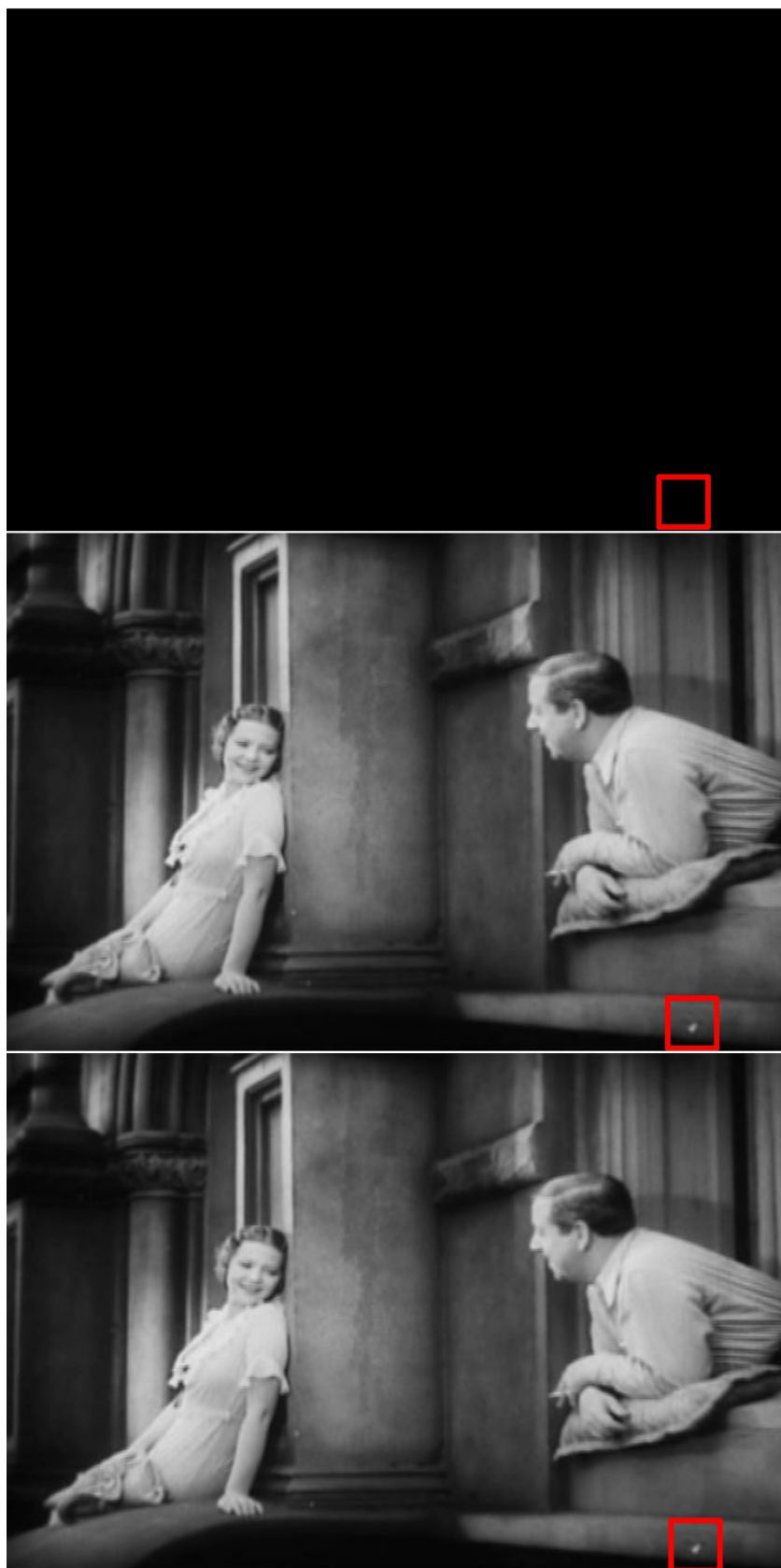
Všechny tři výše uvedené typy detekcí jsem použil na stejné videosekvence z filmů „Street Scene“ a „Pecks bad boy“. Odstranění detekovaných skvrn bylo u všech algoritmů implementováno stejně, pro spravedlivé porovnání jednotlivých typů detekcí. Výsledky jsou uvedeny na obrázcích 12.6 – 12.11. Po prohlédnutí výsledků metod a po porovnání jejich výpočetní náročnosti shrnuji výsledky do následujících bodů:

- 1) Adaptivní algoritmus dokázal jako jediný detekovat skvrny, které jsou na pozadí s podobnou úrovní jasu jako má skvrna. U SDIa a sROD testů tedy skvrna detekována nebyla (obrázky 12.6 – 12.8).
- 2) Při rychlém pohybu v obraze jsou SDIa a sROD testy zatíženy velkým množstvím falešných detekcí, které mají špatný vliv na rekonstrukci a rekonstruovaný obraz je na první pohled nekvalitní. Naopak adaptivní metoda se zabudovanou detekcí pohybu v obraze dokáže pohyb nejenom kompenzovat (jako algoritmus SDIa a sROD) ale zároveň v případech kdy shoda není dobrá, zvýšit práh citlivosti a redukovat tím množství falešných detekcí (obrázky 12.9 – 12.11).
- 3) Pro uspokojivou detekci je třeba u SDIa a sROD testu nastavit odlišné prahové hodnoty pro různé snímky ve videosekvenci. Adaptivní test je zcela univerzální a funguje na libovolné sekvenci.
- 4) Výpočetní složitost je pro SDIa a sROD přibližně stejná. (2,5 minuty pro 300 snímků na průměrném PC). Adaptivní test trval přibližně 2x déle.

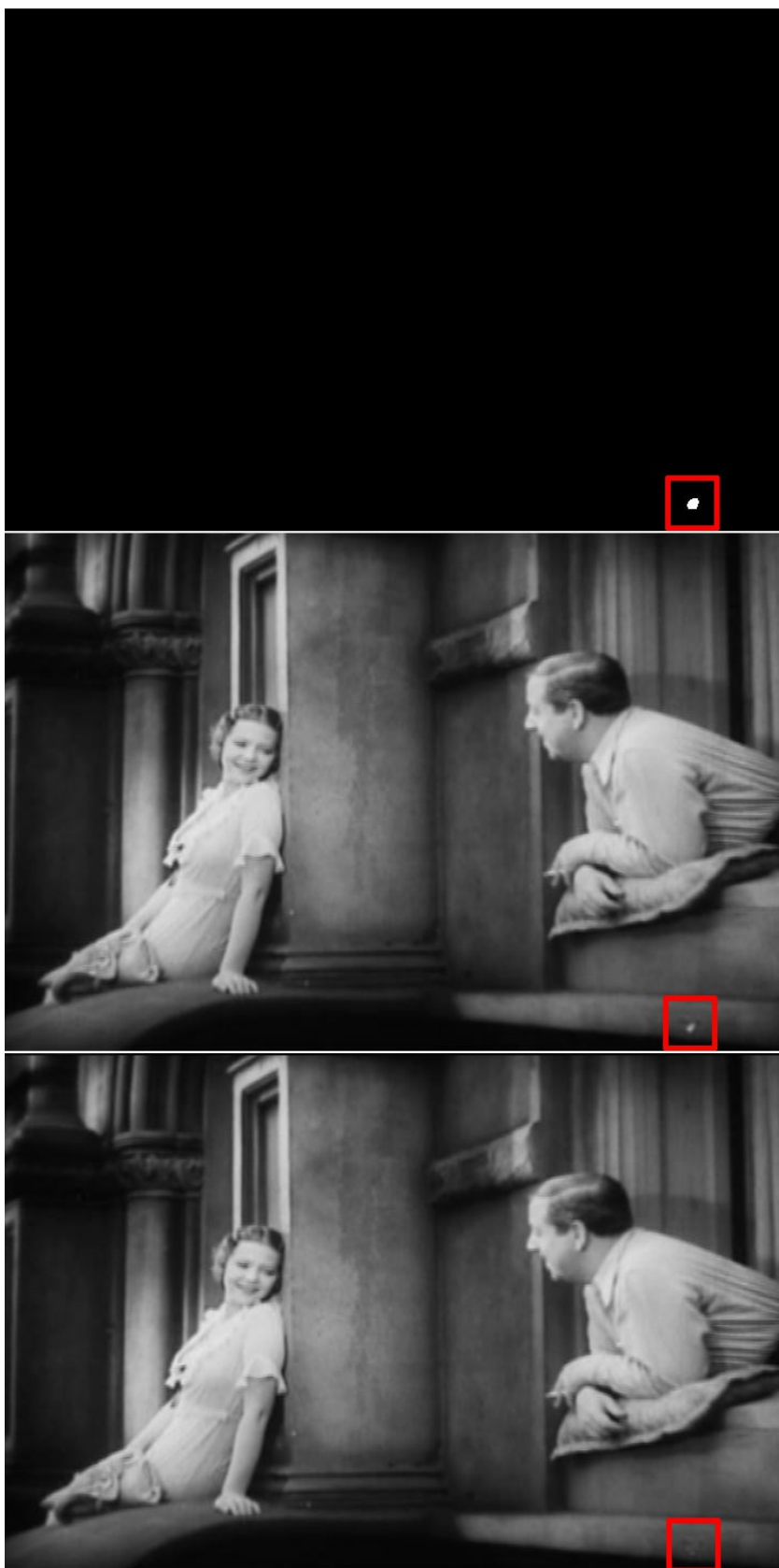
Při shrnutí výsledků jsem dospěl k jednoznačnému závěru, že adaptivní test je nejlepší volbou pro odstranění skvrnové vady z historických filmových sekvencí. Ve všech případech detekce produkoval nejlepší výsledky. Jeho jediná nevýhoda, výpočetní složitost, nehraje v případě restaurace archivních děl roli. Jako druhý z testů vyšel nejlépe test sROD. Test SDIa produkoval velice často neuspokojivé výsledky, především velké množství falešných detekcí a pro restauraci historických filmů bych ho doporučil nepoužívat.



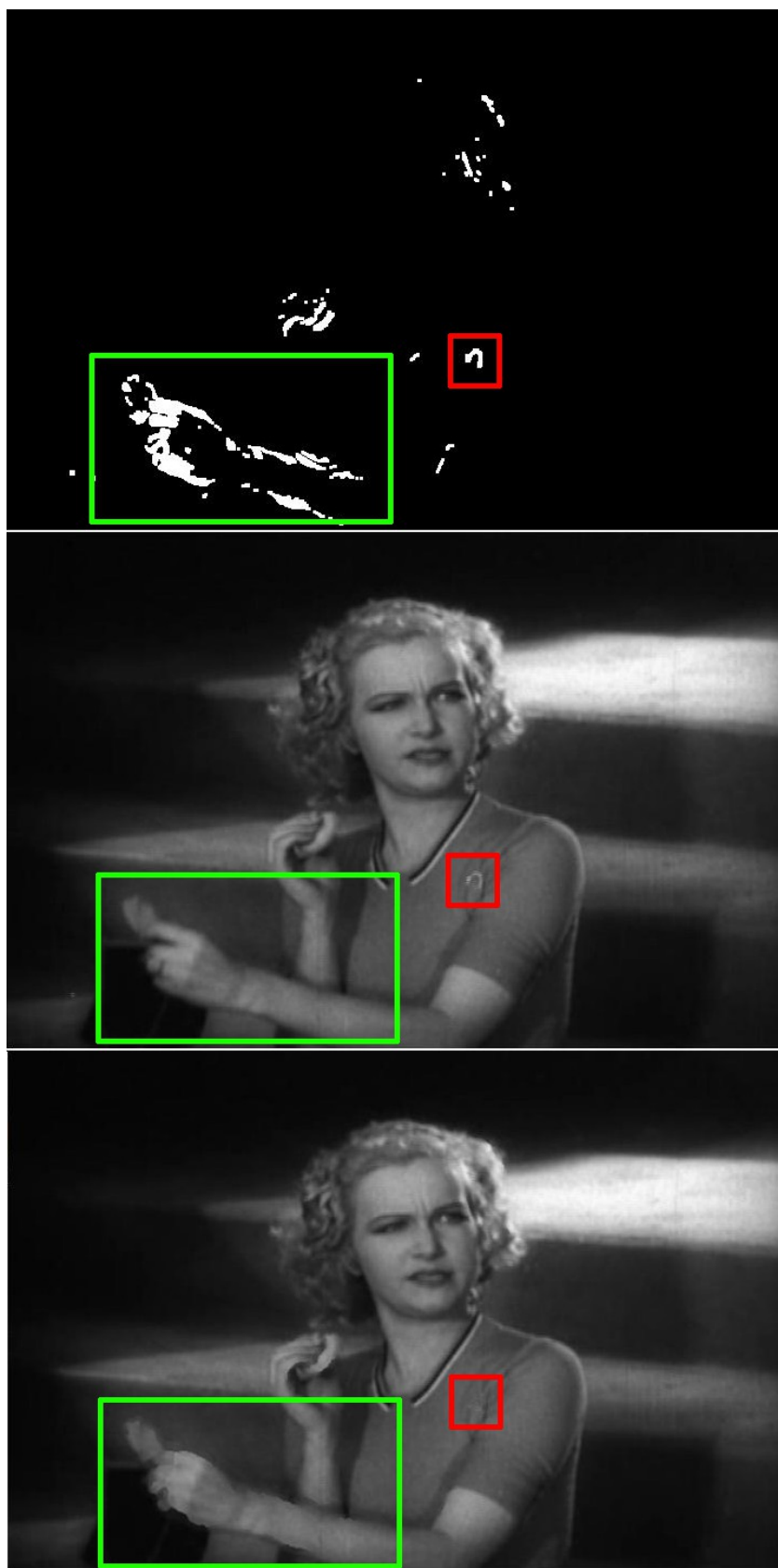
obrázek 12.6: Snímek z filmu *Street Scene*, SDIa test, na horním obrázku je uvedena detekce (není detekována žádná vada), na druhém obrázku původní snímek, dole je upravený snímek. Červeným rámečkem je označeno místo kde se nachází skvrnová chyba.



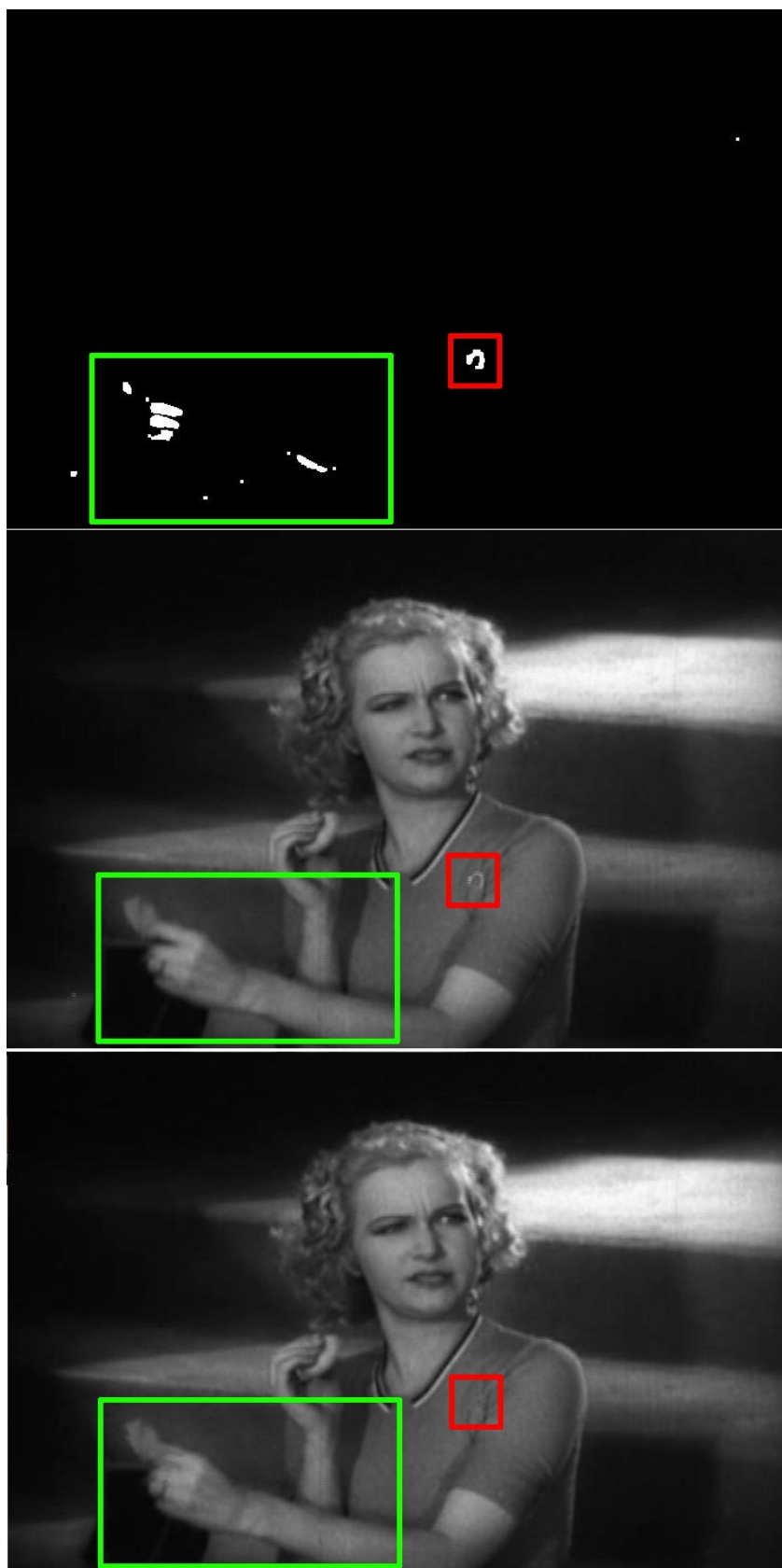
obrázek 12.7: Snímek z filmu *Street Scene*, sROD test, na horním obrázku je uvedena detekce (není detekována žádná vada), na druhém obrázku původní snímek, dole je upravený snímek Červeným rámečkem je označeno místo kde se nachází skvrnová chyba.



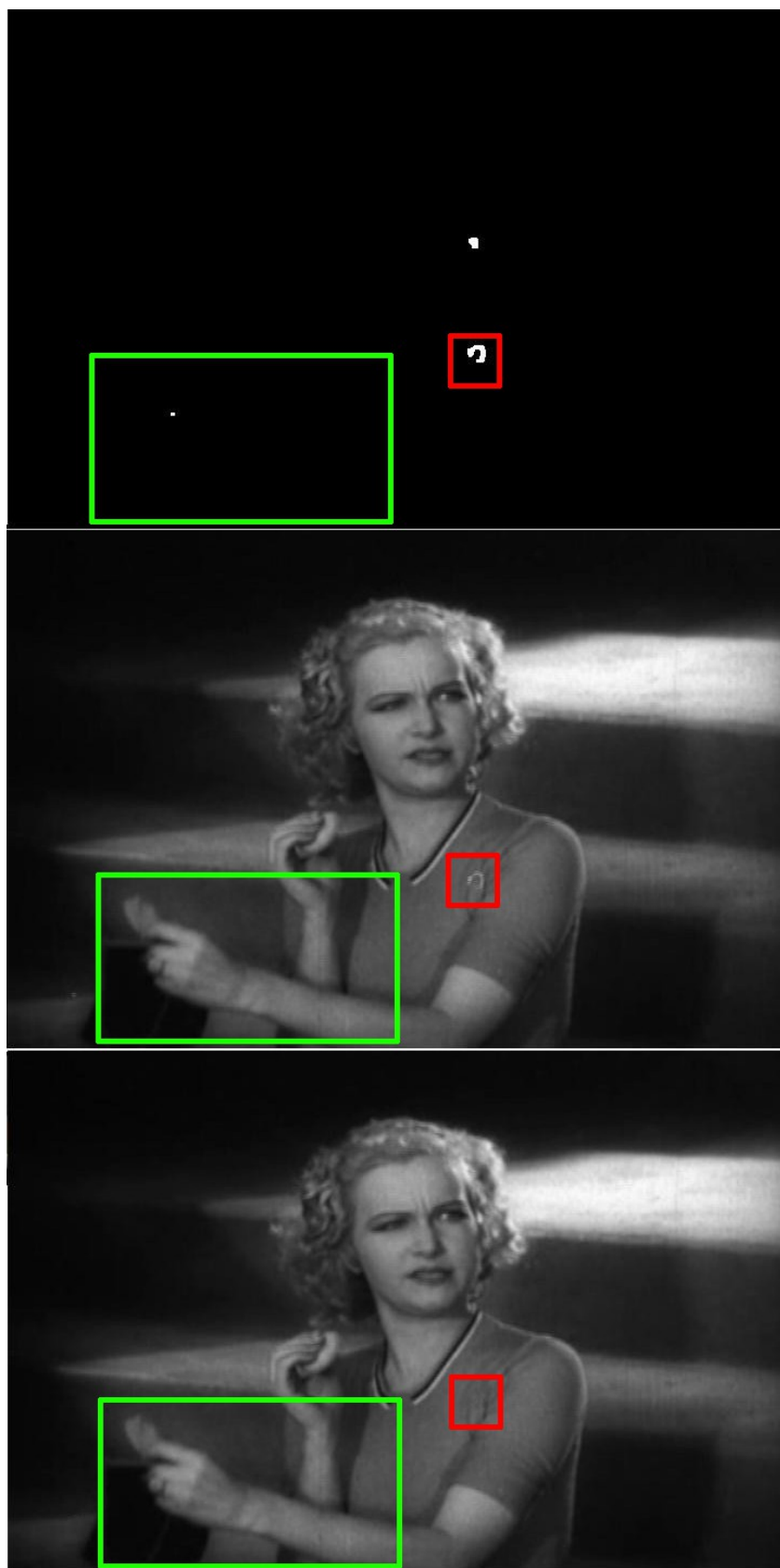
obrázek 12.8: Snímek z filmu *Street Scene*, adaptivní test, na horním obrázku je uvedena detekce, na druhém obrázku původní snímek, dole je upravený snímek Vada je detekována a potlačena. Červeným rámečkem je označeno místo kde se nachází skvrnová chyba.



obrázek 12.9: Snímek z filmu *Street Scene*, SDIa test, na horním obrázku je uvedena detekce, na druhém obrázku původní snímek, dole je upravený snímek. Červeným rámečkem je označeno místo se skvrnovou chybou, zeleným rámečkem místo, kde se skvrnová chyba nenachází, a přesto se zde objevuje velké množství falešných detekcí vlivem pohybu v obraze



obrázek 12.10: Snímek z filmu *Street Scene*, sROD test, na horním obrázku je uvedena detekce, na druhém obrázku původní snímek, dole je upravený snímek. Červeným rámečkem je označeno místo se skvrnovou chybou, zeleným rámečkem místo, kde se skvrnová chyba nenachází, a přesto se zde objevuje velké množství falešných detekcí vlivem pohybu obrazu

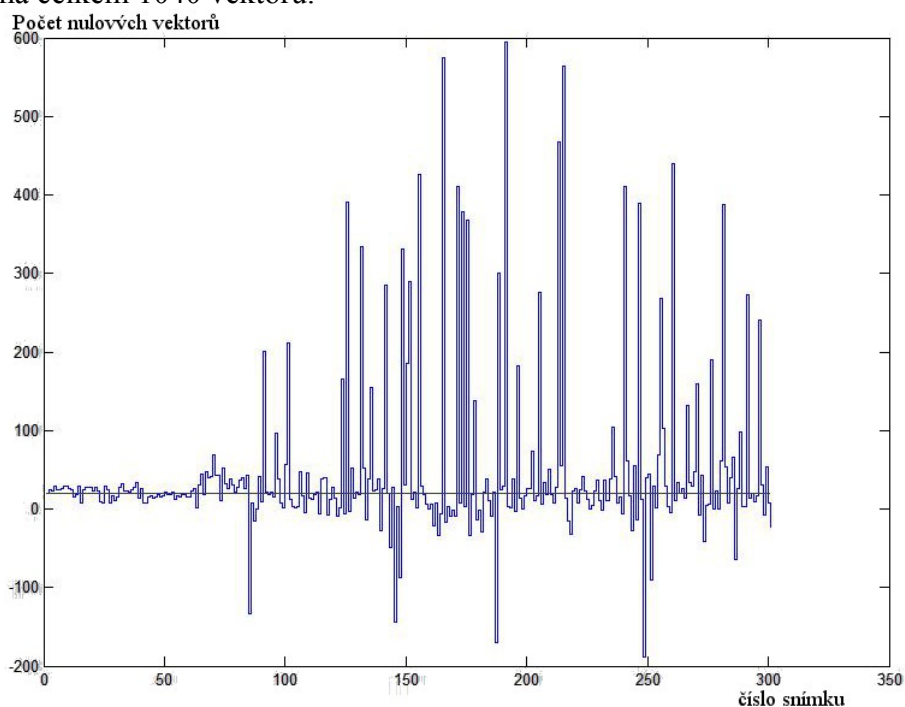


obrázek 12.11: Snímek z filmu *Street Scene*, adaptivní test, na horním obrázku je uvedena detekce, na druhém obrázku původní snímek, dole je upravený snímek. Červeným rámečkem je označeno místo se skvrnovou chybou, zeleným rámečkem místo, kde se skvrnová chyba nenachází, a přesto se zde objevuje v ostatních metodách velké množství falešných detekcí vlivem pohybu obrazu

12.2 Implementace korekce nestability obrazu pomocí globálního vektoru pohybu

V kapitole 8 byla detailně popsána metoda na odstranění nestability obrazu. Uvedenou metodu jsem implementoval na videosekvenci filmu "Pecks bad boy", konkrétně na jejích prvních 300 snímků (10 vteřin). Na začátku této videosekvence je totiž výrazně viditelná nestabilita obrazu, a proto byla zvolena jako vhodná videosekvence pro testování. Výsledky jsou porovnávány pomocí počtů blokových nulových vektorů pohybu. Pokud byla stabilizace úspěšná, měl by být vidět citelný nárůst nulových vektorů.

Při implementaci algoritmu přesně podle kapitoly 8 jsem nejprve rozdělil celou plochu snímku na dvě zóny: pozadí a popředí. Dále jsem provedl výpočet lokálních vektorů pohybu v těchto zónách, porovnal jsem je a vybral globální vektor pohybu. Bohužel jsem nedosáhl uspokojivých výsledků. Obraz byl opticky v lepším stavu než původní video sekvence, ale stále se v některých úsecích klepal. Objektivní hodnocení, tedy hodnocení příbytku nulových vektorů oproti původní videosekvenci, vykazovalo nárůst, ale kvůli viditelné nestabilitě obrazu nebyl výsledek příliš uspokojivý. Tento výsledek je vidět na obrázku 12.12, který zobrazuje příbytek nulových vektorů oproti původní sekvenci. Pro porovnání, jeden snímek (při provedeném nastavení velikosti bloku 18x18 pixelů) má celkem 1040 vektorů.



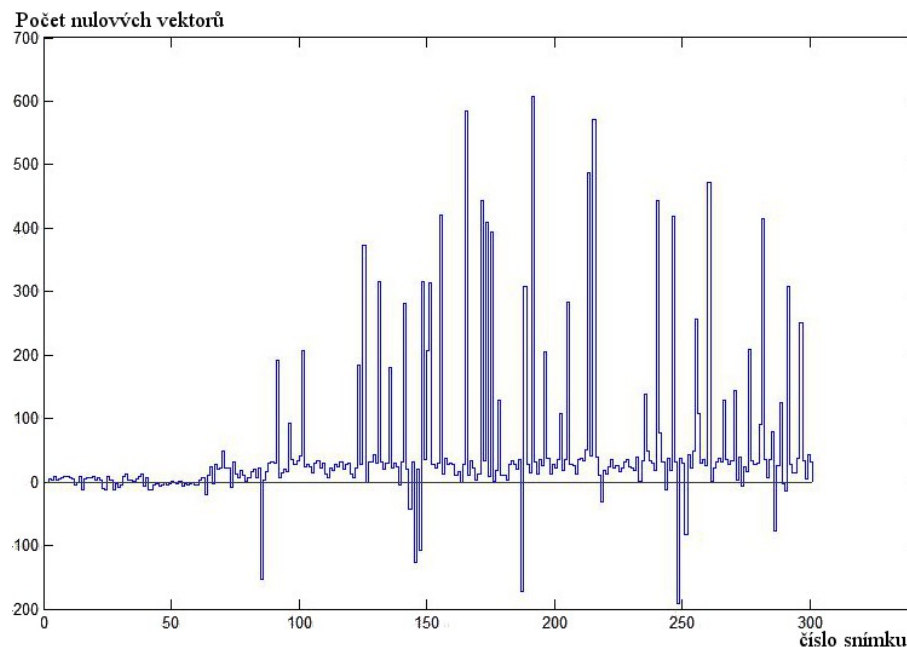
obrázek 12.12: Nárůst počtu nulových vektorů po použití neupravené metody

Metodu jsem se pokoušel upravit tak, abych dosáhl lepších výsledků. Prvního zlepšení se mi podařilo dosáhnout upravením tabulky 8.1. Bylo vidět, že podle původního nastavení vah blokových vektorů určených tabulkou 8.1 se ve filmové sekvenci „The peck boys“ nachází přibližně 75% bloků, které mají nulovou váhu. Zatímco bloky, které obsahovaly blotch chybu nebo pohyb v obraze, měly váhu příliš velkou. Tento nepoměr vedl často k falešné detekci globálního vektoru pohybu. Empiricky jsem tedy zkoušel měnit prahové hodnoty z tabulky 8.1 a podařilo se mi dosáhnout výrazného zlepšení detekce při nastavení prahů podle tabulky 12.1.

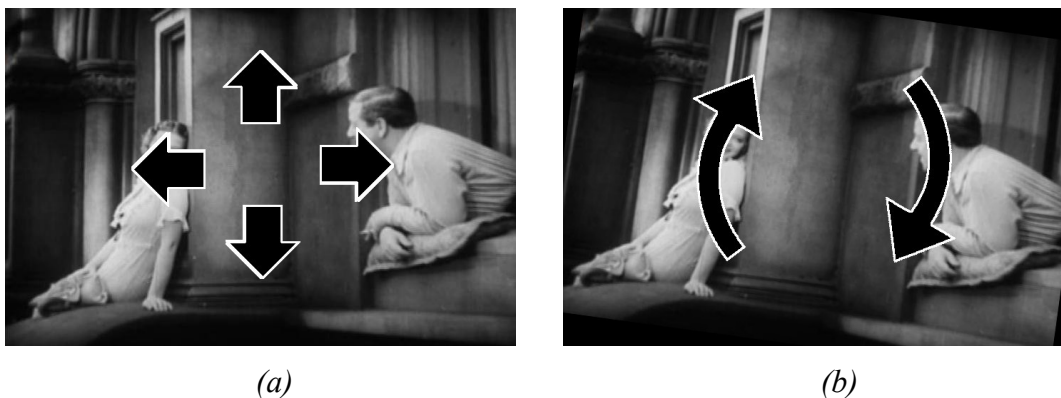
MAD	Váha
$MAD < 5$	0
$5 < MAD < 15$	1
$15 < MAD < 25$	2
$25 < MAD < 40$	3
$40 < MAD$	4

tabulka 12.1: Upravená váha bloku pro určení globálního vektoru pohybu podle MAD

Druhou úpravou, kterou se mi podařilo zvednout počet nulových vektorů, bylo odstranění zón pozadí a popředí. Počítal jsem jeden globální vektor z celého snímku. Tím se efektivita metody pro implementaci na filmové stopě „The peck boys“ zvýšila přibližně o 10%, jak je vidět na obrázku 12.13. Na výsledku je také vidět, kde bylo zapotřebí stabilizovat obraz. Tam, kde je vidět velký nárůst nulových vektorů, docházelo mezi sousedními snímky k nechtěnému globálnímu pohybu, který byl stabilizován, čímž se dosáhlo velkého příbytku nulových vektorů. Naopak tam, kde byly sousedící snímky vpořádku, nebo jejich posunutí bylo nepatrné, nedochází téměř k žádnému příbytku nulových vektorů. Na grafu jsou také vidět místa, kde metoda naopak pohyb do obrazu přidala. To je způsobeno tím, že nestabilita v obrazu se neprojevuje jenom posunem obrazu po osách x a y (viz obrázek 12.14(a)), ale také rotací obrazu (obrázek 12.14(b)) či jeho nakosením. V takovém případě může tato metoda založená na blokových vektorech pohybu detekovat globální pohyb obrazu a pokusí se ho opravit posunem po osách x a y , což nevede ke správnému řešení. Tato skutečnost může být vidět na grafu nulových vektorů, ale opticky není postřehnutelný rozdíl. Snímek se i v originálním podání jeví jako špatný a po korekci touto metodou nevypadá ani hůře ani lépe. Naopak většina nechtěného pohybu v obraze podle os x a y byla bez problému odstraněna. Metoda má tedy velké uplatnění a její výsledky jsou velmi slibné.



obrázek 12.13: Nárůst počtu nulových vektorů po použití upravené metody

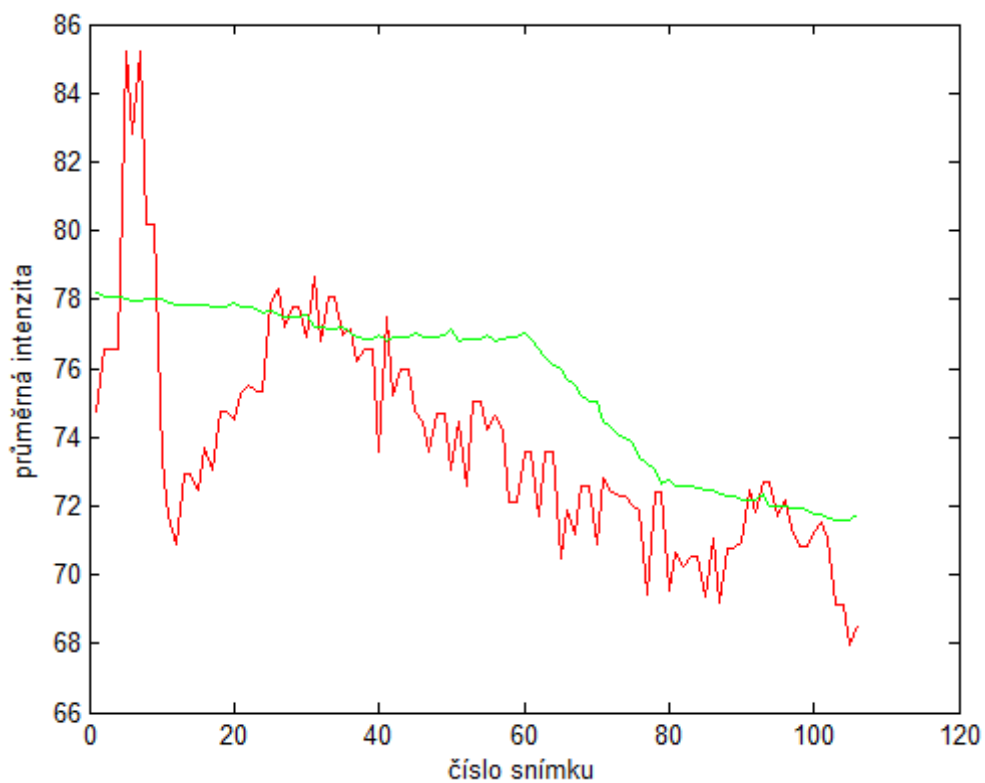


obrázek 12.14: Typy nechtěného pohybu v obraze (a) posun po osách x,y ; (b) rotační pohyb snímku

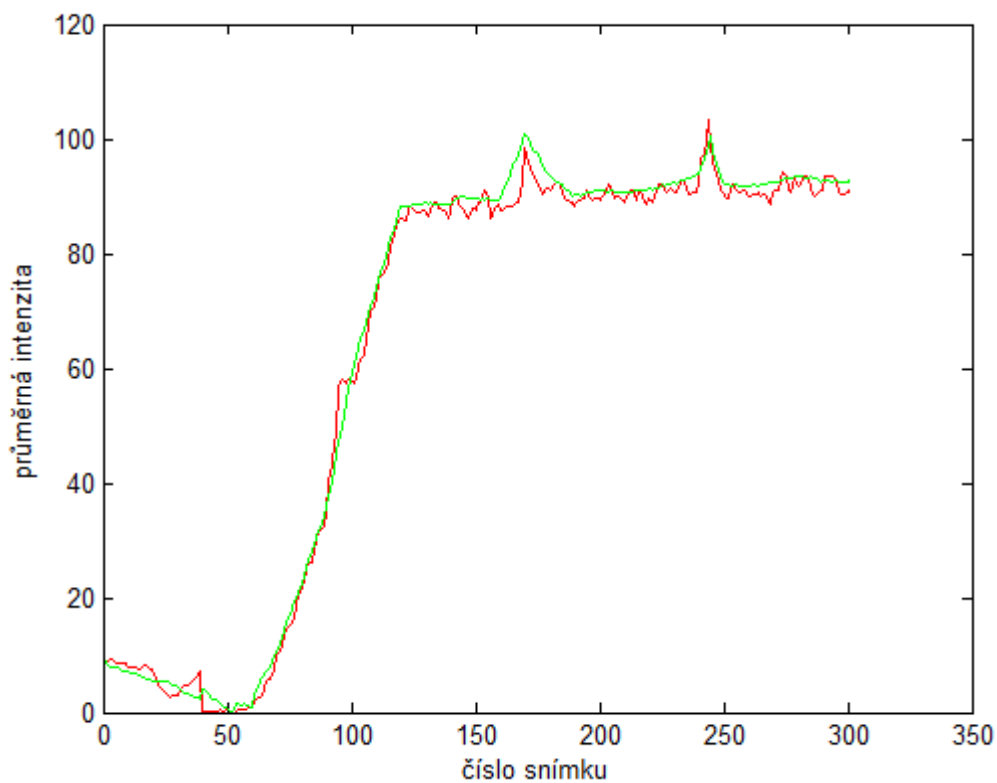
12.3 Implementace korekce blikavého šumu pomocí ekvalizace histogramu

Tato metoda byla detailně popsána v kapitole 4.2. Jedná se o metodu potlačující blikavý šum (tedy výkyvy v intenzitě) pomocí ekvalizace histogramu. Metoda by měla zajistit kvalitní odstranění jak lokální vady blikavého šumu, tak globální, tedy postihující celý snímek. Nevýhodou této metody je nutnost manuálního nastavení referenčních snímků, které budou použity pro restauraci a s kterými bude počítáno jako se snímky, které nejsou postiženy blikavým šumem. V této první fázi je zapotřebí zvolit správně referenční snímky. Osvědčený způsob je mít video zobrazeno dvakrát vedle sebe. V prvním videu je zobrazen aktuální referenční snímek (popřípadě první snímek, pokud začínáme), v druhém videu potom hledáme další vhodný referenční snímek a porovnáváme rozložení intenzity jasu ve snímcích z obou videí. Cílem je najít snímek, který má podobný obsah a rozložení intenzity. Takový můžeme označit za referenční. Dále ho můžeme nastavit na levém videopřehrávači a na pravém hledat další referenční snímek a tímto způsobem dojít až do konce videa a najít všechny potřebné referenční snímky. Tento proces hledání referenčních snímků je jednoduchý a efektivní. V případě, že se v sousedních snímcích mění obsah scény, musíme oba snímky označit za referenční, jinak bychom dostali výrazně zkreslený obraz.

Tuto metodu jsem testoval na dvou 10 vteřinových videosekvencích. Jedna byla z filmu „Street Scene“, která byla postižena globálním blikavým šumem. Druhá byla z filmu „Peck bad boy“ a byla postižena pouze lokálním blikavým šumem. Referenční snímky se mi dařilo nacházet přibližně 6-9 snímků od sebe, což plně vyhovuje podmínce z teoretické části (5-20 snímků od sebe). V místech, kde byl střih scény a záběr na jinou postavu, jsem musel označit oba snímky na tomto rozhraní jako referenční. Výsledky byly vizuálně velice dobré. V upravených videosekvencích nebyl blikavý šum přítomen vůbec. To je vidět na porovnání tří po sobě následujících snímků ve filmové sekvenci poškozené blikavým šumem. Na obrázku 12.17 je sekvence před korekcí blikavého šumu a na obrázku 12.18 po korekci. Na obrázcích 12.15 a 12.16 je vidět průměrná intenzita snímků ve videosekvencích. Červenou barvou je vyznačena průměrná intenzita jasu originální sekvence, zelenou pak sekvence upravené. Z těchto grafů je vidět že výkyvy v intenzitě byly odstraněny. Tato metoda je tedy efektivní a vhodná pro úpravu archivních filmů.



obrázek 12.15: Průměrná hodnota intenzity ve videosekvenci 1 z filmu *Street Scene*, červená barva náleží originální sekvenci, zelená opravené verzi



obrázek 12.16: Průměrná hodnota intenzity ve videosekvenci z filmu *Peck bad boy*, červená barva náleží originální sekvenci, zelená opravené verzi



obrázek 12.17: Tři po sobě následující snímky postižené vadou blikavého šumu z filmu „Street Scene“



obrázek 12.18: Tři po sobě následující snímky postižené vadou blikavého šumu z filmu „Street Scene“, po korekci pomocí ekvalizace histogramu

12.4 Implementace detekce vady škrábnutí mediánovým filtrem a Houghovou transformací

Tato metoda je založena na dvou fázích, které jsou popsány v teoretické části v kapitole 6.2.2. První fáze je detekce všech pixelů, které mohou být postiženy vadou škrábnutí. Druhá fáze je hledání v těchto detekovaných pixelech skupiny pixelů, které odpovídají parametru vady škrábnutí, tedy jsou to úsečky.

První fázi jsem nejprve naimplementoval přesně podle teoretické části. Obraz jsem vertikálně podvzorkoval. Podvzorkování jsem provedl v poměru 1:10 (tedy deset pixelů sousedících vertikálně vždy obsahovalo stejnou hodnotu). Poté jsem na snímek aplikoval horizontální mediánový filtr o šířce 5 pixelů. Nakonec jsem takto vytvořený snímek odečetl od původního snímku podle vzorce (12.4). Prahové hodnoty pro detekci vadných pixelů jsem volil podle doporučení z teoretické části, tedy $s_{med}=3$ a $s_{avg}=20$ a dosadil je do rovnice (12.5). Výsledná detekce nebyla špatná a škrábnutí na ní byla poměrně dobře vidět.

$$e(x, y) = G_s(x, y) - M_s(x, y) \quad (12.4)$$

$$c_1(x, y) : |I_g(x, y) - I_m(x, y)| > s_{med} \quad (12.5)$$

$$c_2(x, y) : |I_t(x, y) - I_r(x, y)| < s_{avg}$$

Přesto jsem se pokusil detekci vylepšit podle doporučení literatury [12]. Zde je uvedena možnost nahrazení vertikálního podvzorkování Gausovským filtrem (3x3), který by měl odstranit šum. Tato úprava mi výsledky detekce nepatrně zhoršila. Zjistil jsem ale, že kombinace vertikálního podvzorkování a Gausovské filtrace detekci ovlivní pozitivně a pro všechny testy, které jsem dále dělal, jsem používal právě kombinaci těchto dvou metod. Po několika testech jsem zjistil, že nejlepších výsledků dosahuji při použití vertikálního podvzorkování 1:6. Druhou změnou parametrů metody byla úprava šířky horizontálního mediánového filtru. Ve filmu, na kterém byla metoda testována, bylo velké množství škrábnutí, která byla 3-5 pixelů široká. Tato škrábnutí byla daleko lépe odstraněna a tedy detekována širším mediánovým filtrem o rozsahu 7 pixelů. Třetí změnou, kterou bylo nutno udělat, bylo zvýšení prahu $s_{avg}=30$. Původní hodnota $s_{avg}=20$ se ukázala jako moc nízká a byly potlačeny i poměrně veliké části jednotlivých škrábnutí. Binární detekce vad po všech těchto úpravách je uvedena na obrázku 12.20. Snímek z videosekvence, na který byla aplikována, je na obrázku 12.19.



obrázek 12.19: Snímek z filmu „Street Scene“, který je postižený vadou škrábnutí



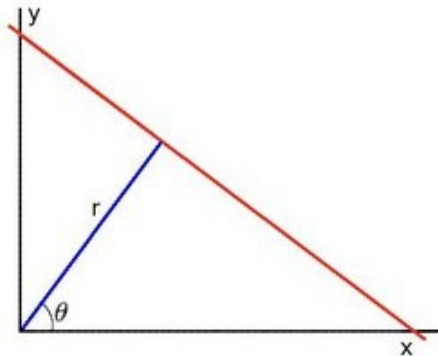
obrázek 12.20: Binární detekce skvrnové vady aplikovaná na snímek z obrázku 12.14

V druhé fázi bylo třeba nastavit Houghovu transformaci tak, aby byla detekce škrábnutí co nejlepší. Abych mohl vysvětlit mnou zvolené nastavení parametrů, nejprve uvedu princip, na kterém pracuje detekce přímek pomocí Houghovy transformace.

Jak je obecně známo, přímku v prostoru můžeme definovat pomocí dvou parametrů.

a) V Kartézském souřadnicovém systému: parametry: (m, b)

b) V Polárním souřadnicovém systému: parametry (r, Θ)



obrázek 12.21: Parametry přímky v polárním souřadnicovém systému

Pro Houghovu transformaci budeme přímky vyjadřovat v polárním systému. Definici přímky tedy můžeme zapsat jako (12.6). Po úpravě rovnice (12.6) získáme rovnici (12.7).

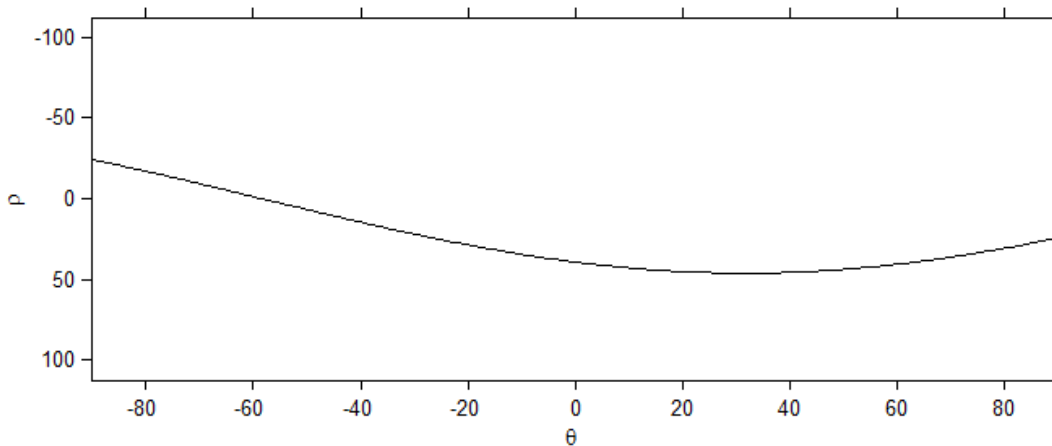
$$y = \frac{-\cos(\theta)}{\sin(\theta)}x + \frac{r}{\sin(\theta)} \quad (12.6)$$

$$r = x \cdot \cos \theta + y \cdot \sin \theta \quad (12.7)$$

Obecně pro každý bod (x_0, y_0) můžeme definovat přímky procházející tímto bodem jako (12.8). To znamená, že každá dvojice (r_θ, Θ) reprezentuje jednu přímku, která prochází bodem (x_0, y_0) .

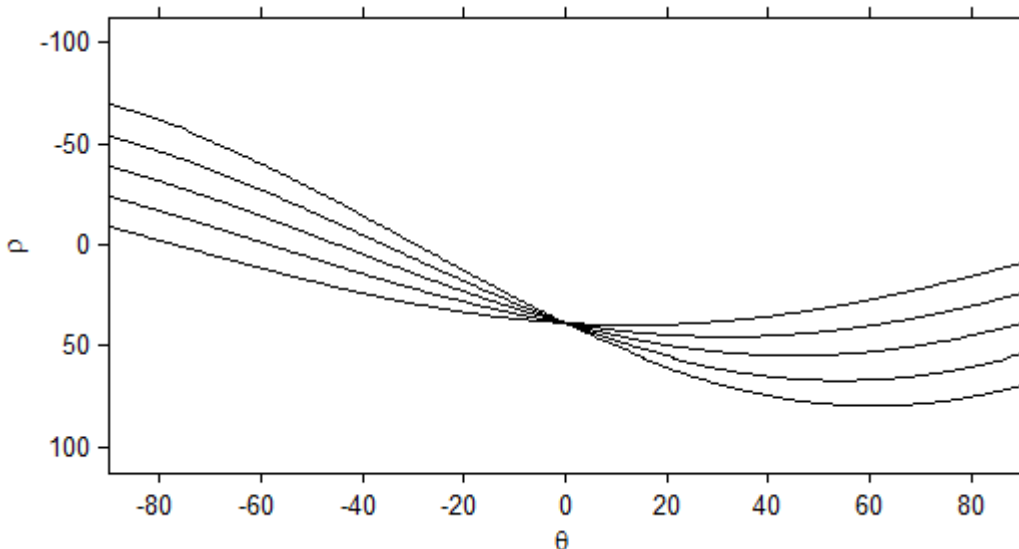
$$r_\theta = x_0 \cdot \cos \theta + y_0 \cdot \sin \theta \quad (12.8)$$

Pro libovolné (x_0, y_0) můžeme zobrazit přímky, které procházejí tímto bodem jako sinusoidu v polárním souřadnicovém systému. Příklad takto vzniklé sinusoidy je v obrázku 12.22.

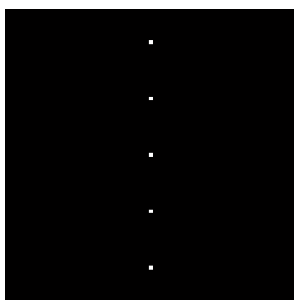


obrázek 12.22: Všechny přímky procházející bodem (25,40) v Polárním systému

Tuto sinusoidu ale nemusíme dělat pouze pro jeden bod, ale můžeme ji udělat pro libovolné množství bodů. Pro každý bod získáme jinou sinusoidu. Na průniku sinusoid leží přímka, která protíná všechny body, jejichž sinusoidy se v tom místě protínají. To je názorně zobrazeno v obrázku 12.23. Na tomto principu je založena Houghova detekce čar v obraze. Předáme jí binární detekci vady škrábnutí a necháme ji najít nejvíc poškozených pixelů ležících na jedné přímce. Takovou přímku označíme jako výsledek detekce.

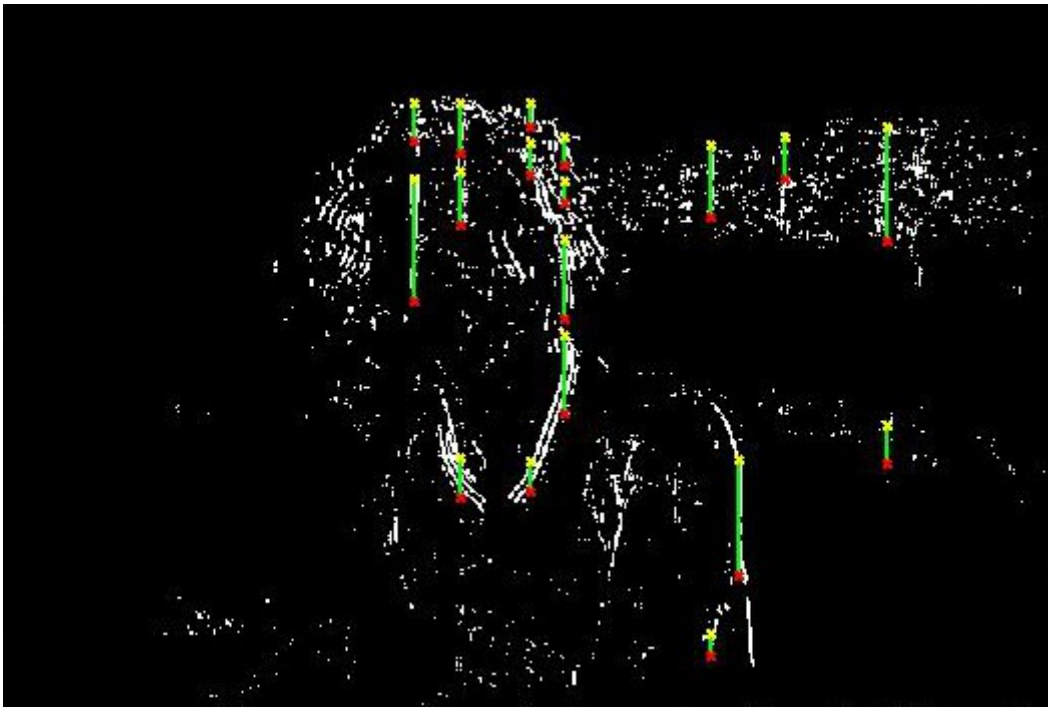


obrázek 12.23: Příklad Houghovy transformace pro pět bodů ležících na jedné přímce



obrázek 12.24: Pět bodů v detekčním prostoru ležící na jedné přímce

Prvním parametrem, který bylo třeba nastavit, bylo Θ . Zvolil jsem rozsah -2 až 2 stupně. To znamená, že detekovány budou jenom přímky, které jsou vychýleny o -2 až 2 stupně od osy y, tedy pouze vertikální škrábnutí. Takovouto detekcí bychom mohli detekovat škrábnutí, která jsou přes celou obrazovku. Většina škrábnutí je ale jen přes část obrazovky a bylo by tedy neefektivní nechat opravit i místa, která nejsou vadou škrábnutí postižena. Proto je Houghova transformace dále upravena tak, aby hledala pouze úsečky o určité minimální délce. Nejlepších výsledků jsem dosahoval při nastavení minimální délky úsečky na 7 pixelů. Posledním parametrem je počet pixelů, který může na přímce chybět, aby stále mohla být určena jako postižená vadou škrábnutí. Tento parametr jsem nastavil na hodnotu 3 pixelů. Při tomto nastavení parametrů jsem dosahoval výsledné detekce uvedené na obrázku 12.24. Abychom zjistili, jak je detekce skutečně přesná a účinná, doporučuji porovnat obrázek 12.19 s obrázek 12.25. Je vidět, že detekce skutečně odhalí většinu pixelů postižených vadou škrábnutí. Jedinou výraznější nevýhodou této metody je nutnost nastavení všech uvedených parametrů, které by mělo být optimalizováno pro každou videosekvenci.



obrázek 12.24: Výsledná detekce vad škrábnutí pomocí Houghovy transformace



Obrázek 12.25: Detekce z obrázku 12.24 promítnutá do původního snímku, černé úsečky představují detekci vady škrábnutí



Obrázek 12.26: Rozšíření detekce z obrázku 12.25 tak, aby došlo k odstranění celé skvrnové chyby

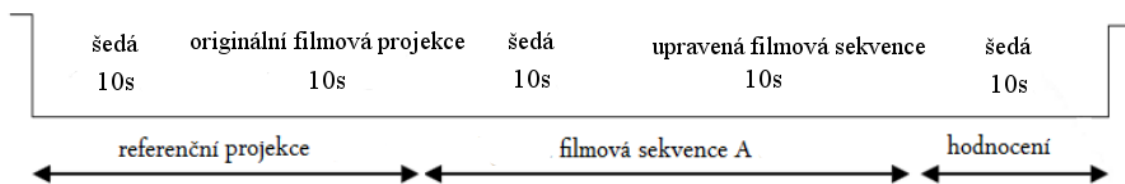
Potlačení vady poškrábáním je implementováno podle algoritmu prezentovaného v kapitole 6.3. Tento algoritmus dosahoval dobrých výsledků a nebylo třeba ho upravovat.

13. Vyhodnocení subjektivních testů

Pro vyhodnocení efektivity implementovaných metod byly provedeny dva subjektivní testy. Oba tyto testy byly popsány v kapitole 10. Jedná se o Test vad s dvojitým podnětem a Test párového porovnání.

13.1 Test vad s dvojitým podnětem

Tento klasický test pro porovnání kvality originální a upravené digitální projekce byl proveden na čtyřech videosekvencích připravených speciálně pro tento test podle návodu z kapitoly 10. Časový průběh testů je zobrazen na obrázku 13.1. Formulář pro hodnocení testů je přiložen jako příloha č. 2. Důležité je poznamenat, že testovanou videosekvencí jsem pozorovatelům promítal pro každou vadu jedenkrát. Při každém spuštění testu se pozorovatelé soustředili a hodnotili pouze jednu konkrétní vadu.



obrázek 13.1: Časový průběh Testu vad s dvojitým podnětem

Výsledky těchto testů jsou uvedeny v tabulkách, které jsou součástí přílohy č. 3. Pro jednoduché porovnání těchto výsledků uvádím tabulku 1 a tabulku 2. Hodnocení u tohoto testu leží v rozsahu 1-velice rušivá vada až 5-nepostřehnutelná vada. U testů je vidět rozdílná efektivita jednotlivých metod. Nejlepšího hodnocení dosáhla metoda pro odstranění skvrnové vady. Naopak redukce šumu pomocí Wienerova filtru nebyla pro pozorovatele dostatečně uspokojivá. Jako nedostatečná se také jeví metoda pro odstranění nestability obrazu. U algoritmu pro redukci vady poškrábáním se dařilo dosáhnout poměrně dobrých výsledků, ale stále je zde prostor pro zdokonalení.

Originální sekvence	Pecks bad boy	Street Scene 1	Street Scene2	Street Scene3
blikavý šum	4,2	4,8	2,8	3,2
šum	4,8	3,8	3	2,4
skvrnová vada	2	1	1,2	2,6
vada poškrábáním	3,4	1,2	1,6	3
nestabilita obrazu	1,6	4,8	2,4	2

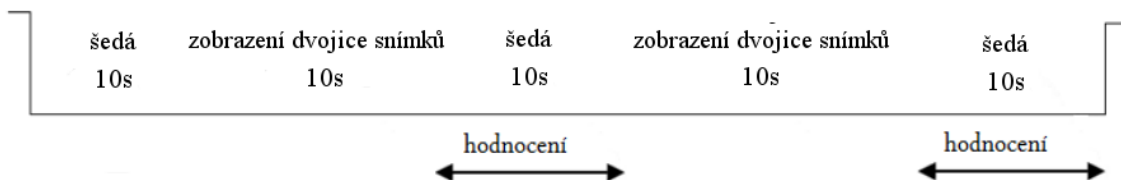
tabulka 1: Průměrné hodnocení snímků ve všech testovaných videosekvencích

Upravená sekvence	Pecks bad boy	Street Scene 1	Street Scene2	Street Scene3
blikavý šum	4,6	4,8	3,6	4,4
šum	4,8	4	3	3
skvrnová vada	3,4	4,2	3	3,6
vada poškrábáním	3,8	3	2,6	3,8
nestabilita obrazu	2,4	4,8	3,2	2,4

tabulka 2: Průměrné hodnocení snímků ve všech testovaných videosekvencích

13.2 Test párového porovnání

Tento test je určený k porovnání rozdílu mezi originální verzí videosekvence a její digitálně upravenou verzí pomocí statických snímků, které jsou zobrazeny vedle sebe. Formulář pro hodnocení tohoto testu uvádím jako přílohu č. 1. Tento test jsem prováděl na třech videosekvencích připravených specificky pro tento test a každá tato sekvence obsahovala 10 párů snímků. Časový průběh testu je vidět na obrázku 13.2. s tím rozdílem, že zde jsou zobrazeny pouze dva páry snímků, ve skutečnosti tedy test trval 5x déle.



obrázek 13.2: Časový průběh Testu párového porovnání

Výsledky těchto testů jsou uvedeny v tabulkách, které jsou součástí přílohy č. 4. Testy dopadly velice pozitivně. Jako shrnutí všech naměřených výsledků uvádím tabulku 3, kde jsou shrnuty průměrné výsledky na jednotlivých videosekvencích. Pro úplnost uvádím, že hodnocení 3 znamená, že v obrázcích není rozdíl, hodnocení 4 znamená, že upravené obrázky byly v lepší kvalitě než originální a hodnocení 5 znamená, že upravený obrázek je ve výrazně lepší kvalitě než originální. U všech sekvencí tedy došlo k výraznému zlepšení vizuálního dojmu.

	Pecks bad boy	Street Scene 1	Street Scene2
Průměrné hodnocení	4,04	3,96	4,4

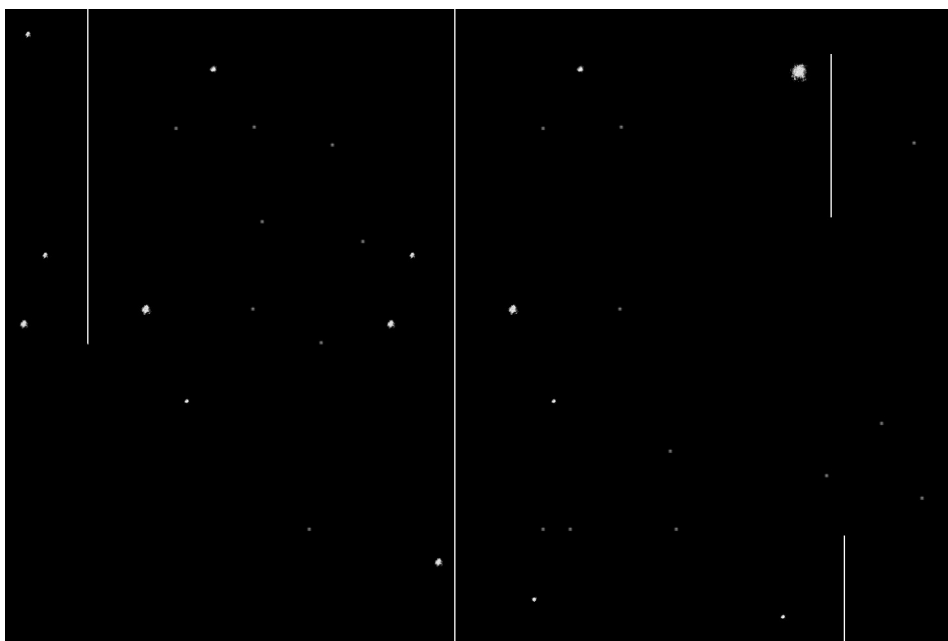
tabulka 3: Průměrné hodnocení snímků ve všech testovaných videosekvencích

14. Vyhodnocení objektivních testů

Jak bylo vysvětleno v kapitole 9 a 11, pro vyhodnocení objektivních testů potřebujeme nejenom poškozenou filmovou sekvenci, ale i originální, nepoškozenou. Na poškozenou sekvenci aplikujeme opravné algoritmy a poté ji porovnáme s originální nepoškozenou stopou.

Bohužel málokdy je k dispozici nepoškozená filmová sekvence i její poškozená kopie. Proto si pro potřeby testování musíme uměle pomoci a do originální, nepoškozené sekvence vady vygenerovat uměle. Tím získáme originální, nepoškozenou sekvenci, uměle poškozenou sekvenci a sekvenci, která vznikla opravením poškozené sekvence. V takovém případě už můžeme provádět objektivní testy. Jako nepoškozenou videosekvenci jsem použil scénu z filmu *Street Scene*, kde již byla skvrnová chyba a vada poškrábáním odstraněna.

Dalším krokem bylo vytvoření poškozené kopie. To jsem provedl podle návodu v kapitole 9 a to tak, že jsem nejdříve vytvořil plochu, která byla 4x větší, než je plocha snímku ve videosekvenci. Tedy pro videosekvence s rozměry 720x480 pixelů měla tato plocha 1440x960 pixelů. Do této plochy jsem přidal vadu poškrábáním a skvrnovou vadu. Některé skvrny jsem vytvořil uměle, jiné jsem zkopíroval z filmů, které je obsahovaly. Výsledná plocha je uvedena v obrázku 14.1.

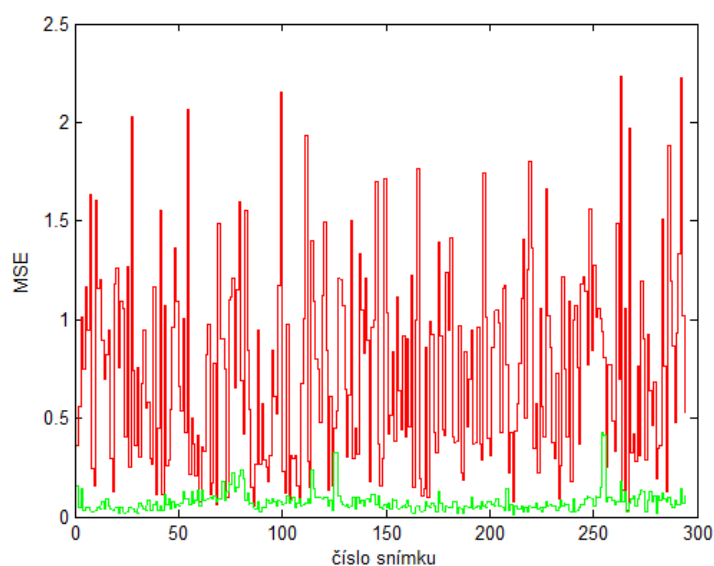


obrázek 14.1: Plocha obsahující skvrnovou chybu a vadu poškrábáním

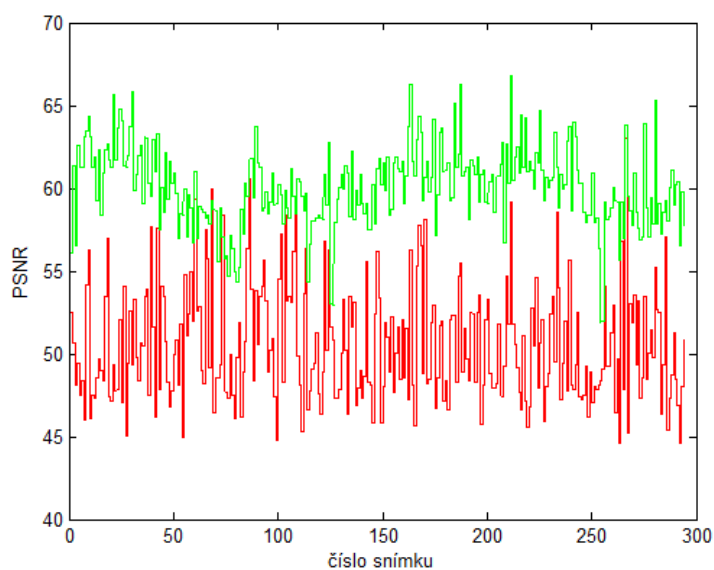
Po vytvoření této plochy obsahující vady ji bylo nutno promítnout do každého snímku ve videosekvenci. To jsem provedl také podle návodu z kapitoly 9. Nejdříve jsem náhodně vygeneroval souřadnice $x \in (1,480)$, $y \in (1,720)$. Od těchto souřadnic jsem „vystříhl“ z plochy v obrázku 14.1 plochu o velikosti snímku ve videu (720x480 pixelů). Poté jsem vygeneroval další náhodné číslo, které programu říká, jestli má vady k obrazu přičítat, nebo odečítat. Ve výsledku to znamená, zda vady mají být reprezentovány bílými (přičítání), nebo černými (odečítání) pixely. Poslední náhodně vygenerované číslo reprezentuje intenzitu vady a leží v rozsahu 0,2 – 1. Tím jsem docílil i méně kontrastních vad. Takto upravený snímek z videosekvence se může označovat za poškozený a připravený pro objektivní testování.

Provedl jsem dva testy uvedené v kapitole 11. A to test MSE a PSNR. Jejich výsledky jsou vidět na obrázcích 14.2 a 14.3. Jak lze z těchto objektivních testů vidět, metody pro odstranění skvrnové chyby a vady poškrábáním skutečně zvyšují kvalitu obrazu. Důležité je podotknout, že MSE a tedy i PSNR (PSNR je odvozeno z MSE) vycházejí v hodnotách, které by odpovídaly kvalitním videostopám. To je způsobeno tím, že do obrazu jsem nepřidával žádný jiný typ šumu, takže filmové sekvence byly přibližně na 95-98% plochy totožné. Jiný šum jsem do obrazu nepřidával, především proto, aby výsledky byly přímo úměrné kvalitě algoritmů na odstranění skvrnové chyby a vady poškrábáním.

Pokud bychom chtěli testovat ostatní vady, postupovali bychom obdobně. Výjimku by tvořila vada blikavého šumu. Tam by plocha obsahující vadu nevypadala ani vzdáleně jako ta na obrázku 14.1, ale byla by to bílá plocha, která by se multiplikativně aplikovala na filmovou sekvenci s koeficientem $k \in (0,0.7)$.



obrázek 14.2: Výhodnocení testu MSE pro 300 snímků sekvence *Street Scene*



obrázek 14.3: Výhodnocení testu PSNR pro 300 snímků sekvence *Street Scene*

15. Závěr

V této diplomové práci jsem se věnoval restauraci digitalizovaných archivních filmových děl.

Nejprve jsem se soustředil na teoretickou studii všech typů vad, které se v archivních filmech vyskytují. Když jsem získal přehled o těchto vadách a jejich vlastnostech, věnoval jsem se metodám, které mohou konkrétní typy vad detekovat a potlačit, právě na základě jejich specifických vlastností.

Praktická část diplomové práce spočívala v implementaci algoritmů navržených v teoretické studii. Tyto algoritmy byly implementovány v programovém prostředí Matlab. Toto prostředí mělo oproti ostatním programovacím jazykům výhodu v podpoře pro velké množství funkcí, které jsou v úpravě digitálního obrazu často využívány, což zjednodušilo tvorbu programů. Zvolil jsem jej zejména z důvodu nutnosti naimplementování většího množství metod a jejich porovnání. Zde vidím příležitost v navázání na mou práci implementací metod, které byly vyhodnoceny jako nejefektivnější v jazyku Java, aby byly dostupné pro veřejnost, která nemá zaplacenou licenci na Matlab.

Při implementaci metod navržených v teoretické části a jejich aplikaci na testovací videosekvence jsem ve většině případů musel algoritmy upravovat, často i výrazně změnit tak, abych dosahoval skutečně co nejlepších výsledků.

Tyto optimalizované algoritmy byly všechny aplikovány na testovací videosekvence a poté byly provedeny subjektivní testy kvality. Výsledky těchto testů byly jednoznačné a metody se ukázaly jako efektivní pro odstranění vad, snad s výjimkou metody pro odstranění nestability obrazu, kde se mi podařilo dosáhnout jen mírného zlepšení. Naopak u adaptivního algoritmu pro odstranění skvrnové vady jsem dosáhl podle testů velice výrazného zlepšení, a proto by tato metoda určitě mohla být kvalitním základem pro metody, které se budou aplikovat v českých i zahraničních filmových archivech.

Archivní filmová díla představují kulturní dědictví, které by nikdy nemělo být zapomenuto. To si v posledních letech uvědomují lidé po celé Evropě a velké množství finančních prostředků je věnováno právě na restauraci historických filmových děl, aby je mohly jednou vidět generace po nás a filmy na ně zapůsobily stejně tak, jako zapůsobily na generace před námi. Moje práce, stejně jako práce navazující, doufám, tomuto vznešenému cíli napomohou.

16. Seznam použitých videosekvencí

Filmová sekvence **Street Scene** (1931) je černobílé drama produkované Samuelem Goldwynandem a režírované Kingem Vidorem. Film se odehrává na ulicích New York City od večera jednoho dne do odpoledne druhého. S výjimkou jediné scény, natočené uvnitř taxi, byly všechny ostatní scény točeny venku a zachycují postupně všechny domy z poloviny čtvrti.

Film obsahoval velké množství vad, především skvrnové vady a vadu blikavého šumu. V menší míře se vyskytovala i nestabilita obrazu a poškrábání. Šum byl většinou přítomen v zanedbatelné míře. Tento film se mi tedy jevil jako dobrý testovací zdroj pro naimplementované metody na potlačení výše zmíněných vad.

- [I] Street Scene, sekvence 1, "Street_Scene_1"
- [II] Street Scene, sekvence 2, "Street_Scene_2"
- [III] Street Scene, sekvence 3, "Street_Scene_3"
- [IV] Street Scene, sekvence 1 po rekonstrukci, "Street_Scene_1_restored"
- [V] Street Scene, sekvence 2 po rekonstrukci, "Street_Scene_2_restored"
- [VI] Street Scene, sekvence 3 po rekonstrukci, "Street_Scene_3_restored"

Filmová sekvence **Peck's Bad Boy** (1934) je černobílé drama režírované Edwardem F. Clinem. Toto drama bylo založeno na sérii knih Geoge W. Pecka.

Film byl v horším stavu než film Street Scene. Obsahoval velké množství všech vad, obzvláště rušivou chybou zde byla nestabilita obrazu a vada poškrábáním. Z filmu jsem si vybral hned úvodní scénu, ve které se postupně zobrazí název filmu a je postižena právě výraznou nestabilitou obrazu.

- [VII] Pecks bad boy, sekvence 1, "Pecks_bad_boy_1"
- [VIII] Pecks bad boy, sekvence 1, "Pecks_bad_boy_1_restored"

Další videosekvence, se kterými jsem ve své práci pracoval, byly **videosekvence upravené pro potřeby testování**. Jedná se o kombinace výše uvedených filmových sekvencí (ve dvojicích originální verze – opravená verze) upravených vždy pro potřeby testování, jak je uvedeno v kapitole 12.

- [IX] Street Scene, sekvence 1, Test porovnáním, „Street_Scene_1_Test2“
- [X] Street Scene, sekvence 1, Test celkové kvality, „Street_Scene_1_Test1“
- [XI] Street Scene, sekvence 2, Test porovnáním, „Street_Scene_1_Test2“
- [XII] Street Scene, sekvence 2, Test celkové kvality, „Street_Scene_1_Test1“
- [XIII] Street Scene, sekvence 3, Test celkové kvality, „Street_Scene_1_Test1“
- [XIV] Pecks bad boy, sekvence 1, Test porovnáním, „Street_Scene_1_Test2“
- [XV] Pecks bad boy, sekvence 1, Test celkové kvality „Street_Scene_1_Test1“

17. Přehled implementací a jejich parametry

I. Blotch_adaptive

Vstupní parametry: video (video určené pro opravu ve formátu avi)
Výstupní parametry: video_detekce (detekce blotch chyby)
video_detekce_s_přesahem (detekce blotch chyby s
přesahem)
video_opravene (opravené video)
video (původní video, neopravené)

II. Blotch_SD1a

Vstupní parametry: video (video určené pro opravu ve formátu avi)
treshold (proměnná určující prahovou hodnotu)
Výstupní parametry: video_detekce (detekce blotch chyby)
video_detekce_s_přesahem (detekce blotch chyby s
přesahem)
video_opravene (opravené video)
video (původní video, neopravené)

III. Blotch_sROD

Vstupní parametry: video (video určené pro opravu ve formátu avi)
treshold (proměnná určující prahovou hodnotu)
Výstupní parametry: video_detekce (detekce blotch chyby)
video_detekce_s_přesahem (detekce blotch chyby s
přesahem)
video_opravene (opravené video)
video (původní video, neopravené)

IV. Scratch_Hough_Transform

Vstupní parametry: video (video určené pro opravu ve formátu avi)
max_vypln (maximální počet chybějících pixelů v úsečce
představující škrábnutí)
min_delka (minimální délka úseček poškrábání, v pixelech)
treshold (prahová hodnota pro detekci škrábnutí)
prah_okoli (prahová hodnota pro zamítnutí detekce)
Výstupní parametry: video_detekce (detekce blotch chyby)
video_opravene (opravené video)
video (původní video, neopravené)

V. Flicker_Hist_equalization

Vstupní parametry: video (video určené pro opravu ve formátu avi)
Výstupní parametry: video_opravene (opravené video)
video (původní video, neopravené)

VI. Unsteadiness

Vstupní parametry: video (video určené pro opravu ve formátu avi)
threshold (proměnná určující prahovou hodnotu)

Výstupní parametry: video_zmensene (opravené video, zmenšené o okraje)
video_opravene (opravené video)
video (původní video, neopravené)

VII. Average_intensity

Vstupní parametry: video (video určené pro výpočet průměrné intenzity)
video2 (video určené pro výpočet průměrné intenzity)

Výstupní parametry: prumer_skutecny (hodnoty průměrné intenzity videa)
prumer_skutecny2 (hodnoty průměrné intenzity videa2)

VIII. Counter_null_vectors

Vstupní parametry: video (video určené pro výpočet počtu nulových vektorů)
video2 (video určené pro výpočet počtu nulových vektorů)

Výstupní parametry: pocitadlo (počty nulových blokových vektorů videa)
pocitadlo2 (počty nulových blokových vektorů videa2)

18. Použité zdroje

- [1] Buadès, A., Delon, J., Gousseau, Y., Masnou, S.: Adaptive blotches detection for film restoration, 17th IEEE Int. Conf. on Image Proc. (ICIP), pp. 3317 - 3320, 2010
- [2] Kyung-tai Kim, Eun yi Kim, "Automatic Film Line Scratch Removal System based on Spatial Inforamtion", 2007
- [3] Peter Michael Bruce Van Roosmalen, "Restoration of Archived Film and Video" 1999
- [4] Kevin Shaw. Color restoration for film [online], dostupné z: <http://www.finalcolor.com/acrobat/ColorRestorfc.pdf>
- [5] Zhang xiaona, Qi guoqing, Xu rong, Zhang Tao, "An Improved Approach of Detection and Restoration Blotches in Archived Films", 2009
- [6] Ranran Zhang, Jiong Wu, Youdong Ding, Pengyi Hao, "The Correction of Intensity Flicker in Archived Film" 2009
- [7] Vittoria Bruni, Domenico Vitulano, "A Ganeralized Model for Scratch Detection", 2004
- [8] D.L. Donoho, "De-noising by Soft-Thresholding", IEEE Trans. on Information Theory, Vol 41, No. 3, pp. 613-627, 1995
- [9] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable Multiscale Transforms", IEEE Trans. on Information Theory, Vol. 38, No. 2, pp. 587-909, 1992.
- [10] M. Vetterli and J. Kovacevic, "Wavelets and Subband Coding", Prentice Hall, USA, 1995.
- [11] Alain Trémeau, "Perception of color and image quality of films projected in cinema", 2007.
- [12] Alasdair Newson, Patrick Pérez, "Adaptive Line Scratch Detection in Degraded Films"
- [13] Zhou Wang, Ligang Lu, Alan C. Bovik, "Video Quality Assessment Using Structural Distortion Measurement"
- [14] A. Kuiper, M. Sigmund, "Simulating of Authentic Movie Faults", Serbia & Montenegro, Belgrade, November 2005
- [15] Yi-Chong Zeng, Si-Yao-Lin, Yi-Ping Shih, Hong-Yuan Mark Liao, "Intensity Flicker Removal in Digitized Old Films Using Global-Local Histogram Transform", Berlin, 2009

[16] Filippo Vella, Alfio Castorina, Massimo Mancuso, Giuseppe Messina, "Digital Image Stabilization by Adaptive Block Motion Vectors Filtering", IEEE Transactions on Consumer Electronics, 2002

[17] Pracovní skupina při Filmové radě pro Ministerstvo kultury "Digitalizace českých filmových děl, návrh koncepce" [online], leden 2010, Dostupné z: <http://www.ceskam.cz/cz/technicke-novinky/digitalizace-ceskych-filmovych-del---navrh-koncepce>

[18] Todd Veldhuizen "Measures of image quality" [online], leden 1998, dostupné z: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/VELDHUIZEN/node18.html

[19] Shyamprasad Chikkerur, Vijay Sundaram, Martin Reisslein, Lina J. Karam, "Objective Video Quality Assessment Methods: A Classification, Review, and Performance Comparison", IEEE Transactions on broadcasting, červen 2011

[20] Peter Chng, "An Overview of Image QA methods" [online], dostupné z: <http://unitstep.net/downloads/school/image-qa-presentation/>

[21] Jaroslav Reichl, Martin Všetická "Systém telecine" [online], dostupné z: <http://fyzika.jreichl.com/main.article/view/1528-system-telecine>

19. Přílohy

Příloha č. 1: Formulář pro subjektivní Test párového porovnání

Název filmové sekvence:
Použité algoritmy:
Hodnotitel:

Pořadové číslo snímku	Hodnocení				
Snímek 1	1	2	3	4	5
Snímek 2	1	2	3	4	5
Snímek 3	1	2	3	4	5
Snímek 4	1	2	3	4	5
Snímek 5	1	2	3	4	5
Snímek 6	1	2	3	4	5
Snímek 7	1	2	3	4	5
Snímek 8	1	2	3	4	5
Snímek 9	1	2	3	4	5
Snímek 10	1	2	3	4	5

1	Výrazně horší kvalita
2	Horší kvalita
3	Stejná kvalita
4	Lepší kvalita
5	Výrazně lepší kvalita

*odpovídající zakroužkujte

_____ datum testu

_____ podpis hodnotitele

Příloha č. 2: Formulář pro subjektivní Test vad s dvojitým podnětem

Název filmové sekvence:
Použité algoritmy:
Hodnotitel:

Typ vady	Sekvence A	Sekvence B
Blikavý šum		
Šum		
Skvrnová vada		
Vady poškrábáním		
Nestabilita obrazu		

1	Velice rušivá vada
2	Rušivá vada
3	Mírně rušivá vada
4	Těžko postřehnutelná vada
5	Nepostřehnutelná vada

*ohodnotte odpovídající známkou

Celkový dojem ze Sekvence A	
Celkový dojem ze Sekvence B	

*ohodnotte známkou 1 (nejhorší) – 5 (nejlepší)

_____ datum testu

_____ podpis hodnotitele

Příloha č. 3: Výsledky Testů vad s dvojitým podnětem:

Street Scene 3						
Originální sekvence	1. pozorovatel	2. pozorovatel	3. pozorovatel	4. pozorovatel	5. pozorovatel	průměr
blikavý šum	3	4	5	4	5	4,2
šum	5	4	5	5	5	4,8
skvmová vada	2	2	2	2	2	2
vada poškrábáním	5	4	2	4	2	3,4
nestabilita obrazu	1	1	3	1	2	1,6

Upravená sekvence						
1. pozorovatel	2. pozorovatel	3. pozorovatel	4. pozorovatel	5. pozorovatel	průměr	
blikavý šum	4	5	5	4	5	4,6
šum	5	4	5	5	5	4,8
skvmová vada	4	4	3	3	3	3,4
vada poškrábáním	5	4	3	4	3	3,8
nestabilita obrazu	3	1	3	2	3	2,4

tabulka 1: Výsledky testu na videosekvenci Pecks bad boy

Street Scene 3						
Originální sekvence	1. pozorovatel	2. pozorovatel	3. pozorovatel	4. pozorovatel	5. pozorovatel	průměr
blikavý šum	5	4	5	5	5	4,8
šum	4	4	4	4	3	3,8
skvmová vada	1	1	1	1	1	1
vada poškrábáním	1	2	1	1	1	1,2
nestabilita obrazu	5	4	5	5	5	4,8

Upravená sekvence						
1. pozorovatel	2. pozorovatel	3. pozorovatel	4. pozorovatel	5. pozorovatel	průměr	
blikavý šum	5	4	5	5	5	4,8
šum	4	4	5	4	3	4
skvmová vada	4	4	5	4	4	4,2
vada poškrábáním	3	3	3	3	3	3
nestabilita obrazu	5	4	5	5	5	4,8

tabulka 2: Výsledky testu na videosekvenci Street Scene, sekvence 1

Street Scene 3						
Originální sekvence	1. pozorovatel	2. pozorovatel	3. pozorovatel	4. pozorovatel	5. pozorovatel	průměr
blikavý šum	4	3	2	2	3	2,8
šum	4	3	3	3	2	3
skvmová vada	2	1	1	1	1	1,2
vada poškrábáním	3	2	1	1	1	1,6
nestabilita obrazu	1	3	3	3	2	2,4

Upravená sekvence						
1. pozorovatel	2. pozorovatel	3. pozorovatel	4. pozorovatel	5. pozorovatel	průměr	
blikavý šum	5	4	3	3	3	3,6
šum	4	3	3	3	2	3
skvmová vada	4	3	3	2	3	3
vada poškrábáním	4	2	2	2	3	2,6
nestabilita obrazu	2	4	3	3	4	3,2

tabulka 3: Výsledky testu na videosekvenci Street Scene, sekvence 2

Street Scene 3						
Originální sekvence	1. pozorovatel	2. pozorovatel	3. pozorovatel	4. pozorovatel	5. pozorovatel	průměr
blikavý šum	3	3	3	3	4	3,2
šum	2	2	3	3	2	2,4
skvmová vada	2	2	3	4	2	2,6
vada poškrábáním	2	3	4	4	2	3
nestabilita obrazu	2	2	2	2	2	2

Upravená sekvence						
	1. pozorovatel	2. pozorovatel	3. pozorovatel	4. pozorovatel	5. pozorovatel	průměr
blikavý šum	4	5	5	4	4	4,4
šum	3	3	3	4	2	3
skvmová vada	4	3	4	4	3	3,6
vada poškrábáním	3	4	4	4	4	3,8
nestabilita obrazu	2	2	3	2	3	2,4

tabulka 4: Výsledky testu na videosekvenci Street Scene, sekvence 3

Příloha č. 4: Výsledky Testů párového porovnání:

Pecks bad boy	1. pozorovatel	2. pozorovatel	3. pozorovatel	4. pozorovatel	5. pozorovatel	průměr
1. snímek	5	4	5	5	5	4,8
2. snímek	4	4	4	3	4	3,8
3. snímek	5	3	3	3	4	3,6
4. snímek	4	4	4	5	4	4,2
5. snímek	4	4	4	4	4	4
6. snímek	4	4	5	5	4	4,4
7. snímek	5	4	4	4	5	4,4
8. snímek	5	4	5	5	3	4,4
9. snímek	3	3	3	3	3	3
10. snímek	4	4	3	4	4	3,8

tabulka 5: Výsledky testu na videosekvenci Pecks bad boy

Street Scene 1	1. pozorovatel	2. pozorovatel	3. pozorovatel	4. pozorovatel	5. pozorovatel	průměr
1. snímek	5	5	5	5	5	5
2. snímek	4	5	4	4	5	4,4
3. snímek	3	3	4	3	4	3,4
4. snímek	3	3	4	3	4	3,4
5. snímek	4	3	4	4	5	4
6. snímek	4	4	4	4	5	4,2
7. snímek	4	4	5	4	5	4,4
8. snímek	3	3	3	3	3	3
9. snímek	4	4	4	4	4	4
10. snímek	4	4	4	4	3	3,8

tabulka 6: Výsledky testu na videosekvenci Street Scene, sekvence 1

Street Scene 2	1. pozorovatel	2. pozorovatel	3. pozorovatel	4. pozorovatel	5. pozorovatel	průměr
1. snímek	5	4	4	4	5	4,4
2. snímek	4	4	4	3	4	3,8
3. snímek	4	5	4	4	5	4,4
4. snímek	5	5	5	4	5	4,8
5. snímek	5	4	4	4	5	4,4
6. snímek	4	4	3	4	4	3,8
7. snímek	5	4	5	5	5	4,8
8. snímek	5	4	5	5	4	4,6
9. snímek	5	4	5	5	5	4,8
10. snímek	5	4	4	3	5	4,2

tabulka 7: Výsledky testu na videosekvenci Street Scene, sekvence 2

Příloha č. 5: Obsah příloženého DVD:

- Metody rekonstrukce archivních videozáznamů.pdf
- Programy
 - Blotch_adaptive
 - Blotch_ROD
 - Blotch_SDIA
 - Counter_null_vectors
 - Flicker_Hist_equalization
 - Average_intensity
 - Unsteadiness
 - Scratch_Hough_Transform
- Videosekvence
 - Pecks bad boy
 - Peck_bad_boy1
 - Peck_bad_boy1_restored
 - Peck_bad_boy1_Test1
 - Peck_bad_boy1_Test2
 - Street Scene 1
 - Street_Scene_1
 - Street_Scene_1_restored
 - Street_Scene1_Test1
 - Street_Scene1_Test2
 - Street Scene 2
 - Street_Scene_2
 - Street_Scene_2_restored
 - Street_Scene2_Test1
 - Street_Scene2_Test2
 - Street Scene 3
 - Street_Scene_3
 - Street_Scene_3_restored
 - Street_Scene3_Test1

