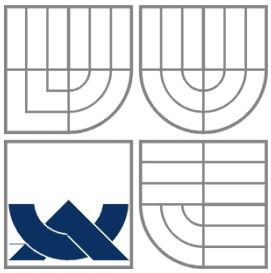
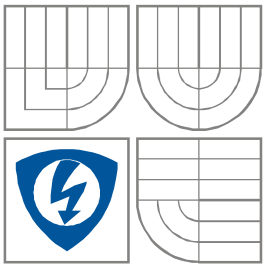


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

Program pro odstranění prokladu ve videosekvencích různých formátů
Program for deinterlacing in video sequences of different formats

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE — Bc. Pavel Jirků
AUTHOR

VEDOUCÍ PRÁCE — Ing. Martin Slanina, Ph.D.
SUPERVISOR

BRNO, 2009

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení:	Bc. Pavel Jirků
Bytem:	Opatov 48, 675 28 Opatov
Narozen/a (datum a místo):	16.ledna 1984 v Třebíči

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 53, Brno, 602 00
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Dr. Ing. Zbyněk Raida, předseda rady oboru Elektronika a sdělovací technika
(dále jen „nabyvatel“)

Čl. 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

   ☐ disertační práce
   ☒ diplomová práce
   ☐ bakalářská práce
   ☐ jiná práce, jejíž druh je specifikován jako .....
   (dále jen VŠKP nebo dílo)

Název VŠKP:	Program pro odstranění prokladu ve videosekvencích různých formátů
Vedoucí/ školitel VŠKP:	Ing. Martin Slanina, Ph.D.
Ústav:	Ústav radioelektroniky
Datum obhajoby VŠKP:	_____

   VŠKP odevzdal autor nabyvateli*:

   ☒ v tištěné formě – počet exemplářů: 2
   ☒ v elektronické formě – počet exemplářů: 2

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.

3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.

4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

* hodící se zaškrtněte

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizovaní výpisů, opisů a rozmnoženin.

5. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.

6. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti

   ☒ ihned po uzavření této smlouvy
   ☐ 1 rok po uzavření této smlouvy
   ☐ 3 roky po uzavření této smlouvy
   ☐ 5 let po uzavření této smlouvy
   ☐ 10 let po uzavření této smlouvy (z důvodu utajení v něm obsažených informací)

7. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.

2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.

3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.

4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 29. května 2009

…….. ……

Nabyvatel Autor

ABSTRAKT

Cílem této diplomové práce je nastudovat současné možnosti využití progresivních algoritmů pro odstranění prokládání ve videosekvencích. První část práce je věnována teorii, základním poznatkům a vlastnostem zpracování s využitím multimediálních dat. Druhá část je věnována zobrazování signálu na různých zobrazovačích při použití prokládaného i progresivního řádkování. Další část je zaměřena na metody konverzí mezi prokládaným řádkováním a progresivním. Poslední část se věnuje implementaci nastudovaných metod. Algoritmus je implementován v jazyce C++, který poskytuje dostatečně rychlé zpracování algoritmů. Závěr práce je věnován otestování a ověření nastudovaných algoritmů.

KLÍČOVÁ SLOVA

Videosekvence, pohyb, detekce pohybu, detekce objektů, obraz, snímek

ABSTRACT

The aim of this thesis is a study of advanced algorithms for removing interlacing in digital video sequences. The first part of the work is devoted to the theory, basic knowledge and processing characteristics using multimedia data. The second part is devoted to displaying the signal using interlaced and progressive spacing. The following part is focused on methods of conversion between the interlaced line spacing and progressive spacing. The last part deals with implementation of the proposed methods. The algorithm is implemented in C++ language, which provides sufficiently fast processing algorithms. The conclusion of work is focused in testing and verification of the implemented algorithms.

KEYWORDS

Video sequences, motion, motion detection, detection of objels, image, frame

Prohlášení

Prohlašuji, že svoji diplomovou práci na téma Program pro odstranění prokladu ve videosekvencích různých formátů jsem vypracoval samostatně pod vedením vedoucího práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 29.května 2009 .....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Martinu Slaninovi Ph.D za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mého diplomové práce.

V Brně dne 29. května 2009 .....
podpis autora

Bibliografická citace

JIRKŮ, P. Program pro odstranění prokladu ve videosekvencích různých formátů. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 52 s. Vedoucí diplomové práce Ing. Martin Slanina Ph.D.

OBSAH

1. Úvod

V současné době jsme svědky dynamického rozvoje v oboru sdělovací techniky. Každý vlastníme televizor, počítač, mobilní telefon, případně digitální kameru či fotoaparát. Tato zařízení mají společné vlastnosti zpracování, využívají multimediální data. Dochází k digitalizaci televizního signálu zejména proto, že je mnohem jednoduší pracovat s digitálními daty než s analogovými daty.

Dnes můžeme denně sledovat několik programů šířených pozemními vysílači či několik desítek programů distribuovaných pomocí satelitů. Co vše stálo za touto cestou?

Prvním krokem byla obrazovka nazývaná "katodová trubice" (CRT – Cathod Ray Tube), která zobrazovala záběry založené na prokládání (monitory, televize). Drobná prodleva mezi obnovením sudých a lichých řádků vedla k efektu “ rozmazání” na LCD monitorech a je to dáno tím, že se LCD monitory zobrazují celé snímky. Při zobrazení sudých a lichých řádků najednou se obnovila jen polovina řádků spolu s pohybujícími objekty, druhá polovina čeká na obnovení a tím vzniká “rozmazání”.

S nástupem LCD monitorů (Liquid Crystal Display), DVD média a digitální kamery se metoda přenosu obrazu ubírala směrem na progresivní skenování a to směrem digitalizací. Aby bylo možno zobrazovat na displejích (na monitorech) televizní analogová data a lépe zpracovávat tato data je potřeba je digitalizovat a překonvertovat na tyto zobrazovací displeje tedy vytvořit neprokládaný obraz.

Metody používané pro odstranění prokladu (deinterlacing) se stále vyvíjí. Pro různé obrazy nebo spíše pohyby v obraze jsou používané různé výpočetní metody. Pro statické obrazy se využívají jednoduché výpočetní metody, kdy nedochází při výpočtech k velkým výpočetním chybám ale jsou méně časově náročné. Na rozdíl od rychlých změn ve scéně, kdy by došlo k velkým výpočetním chybám se využívají složitější metody pro výpočet neprokládaného videa. Metody jsou výpočetně složitější, ale nedochází k artefaktům ve scéně.

Cílem diplomové práce je navrhnout algoritmus pro odstranění prokládání videosekvencí. Při návrhu bylo využito programového prostředí Matlab a C++, pro rychlejší zpracování složitých výpočtů odstranění prokladu. V jazyce C++ byla vytvořená konzola pro ověření výpočtů.

První kapitola se zabývá teorii základních poznatků a vlastností zpracování multimediálních dat.

Druhá část je věnována zobrazování signálu na různých typech obrazovek a to progresivní a prokládané zobrazení.

Třetí část je zaměřena na metody konverzí mezi prokládaným a progresivním řádkování a naopak.

Poslední část se věnuje implementaci a otestování nastudovaných metod v jazyce Matlab a C++ pro různé typy videosekvecí..

2. Základy televizní techniky

2.1 CRT zobrazovače a prokládané řádkování

Principem je elektronový paprsek dopadající na luminofor na stínítku obrazovky. Paprsek je magnetickým polem vychylován řádek po řádku pro celý snímek. Luminofor má poměrně krátkou dobu dosvitu a krátce za paprskem pohasíná. Pozorovatel tak při nízkých kmitočtech vnímá obraz jako nepříjemné blikání.

Z důvodů šířky pásma nebylo možné jednoduše zvýšit snímkovou frekvenci, proto bylo zavedeno prokládané řádkování. Televizní vysílání bylo pomocí prokládaného řádkování jedinou možností jak přenášet obraz relativně úzkým frekvenčním pásmem. Obraz je snímán, přenášen a zobrazován po tzv. půlsnímcích (liché řádky, pak sudé). Tím se snímková frekvence zdvojnásobí, celkový počet řádků na snímek (a tím celkové rozlišení ve svislém směru) se však nemění.

Pro dojem plynulého pohybu obrazu je zapotřebí dosáhnout určité hodnoty snímkové frekvence - všeobecně uznávaná hodnota je 24-25 snímků (norma PAL) nebo 30 snímků norma (SECAM) za sekundu. Pokud je obraz promítán způsobem běžným (snímek je jistou dobu promítán, pak se při zatemnění film posune a promítá se další snímek), závisí úspěch převážně na setrvačnosti sítnice lidského oka, zde výše zmiňovaná hodnota stačí.

Obr. 1: Prokládané řádkování. [5]

Na obrázku 1 [5] je vidět složení obrazu ze dvou půlsnímků. Prokládané řádkování slouží a sloužilo analogovým kamerám, televizím a světu VHS po mnoho let a v některých případech patří ještě k “nejlepším“, možnostem.

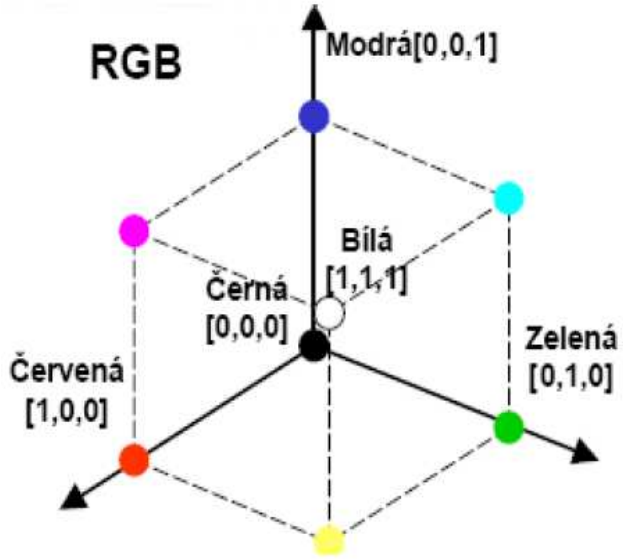
2.2 Úpravy dat

Formát RGB

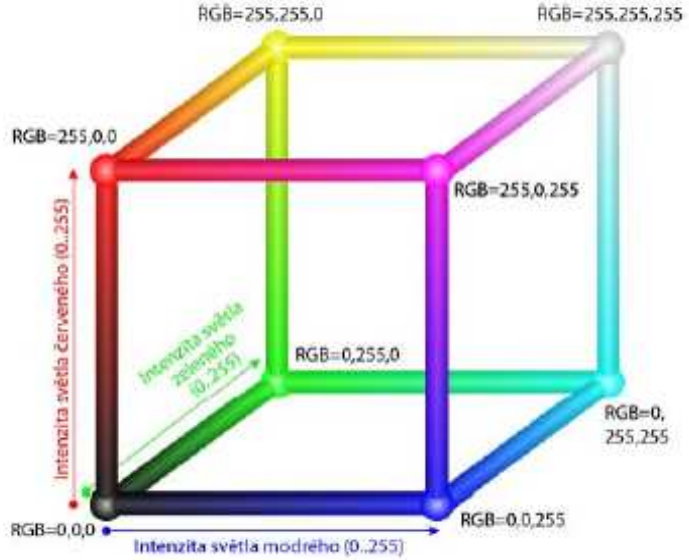
Každý obrazový bod je vyjádřen třemi hodnotami v barevném modelu RGB (červená, zelená, modrá) množstvím jednotlivých barevných složek světla dopadajícího na senzory přístroje.

$$I = 0{,}299\ R + 0{,}587\ G + 0.114\ B \qquad (1)$$

Zobrazovač vytváří obraz na základě metody aditivního míchání světel – smícháme-li světla dvou barev, nová barva vznikne na základě sloučení jejich spektra (smícháním všech barev pak vznikne bílá). Protože lidské oko nevnímá všechny složky barevného obrazu stejně, ale je nejvíce citlivé na zelenou a nejméně na modrou barvu. Intenzita složek barevného spektra je vyjádřena jako: (1) [1]



Obr. 2: Jednotková krychle pro barevný model RGB. [1]



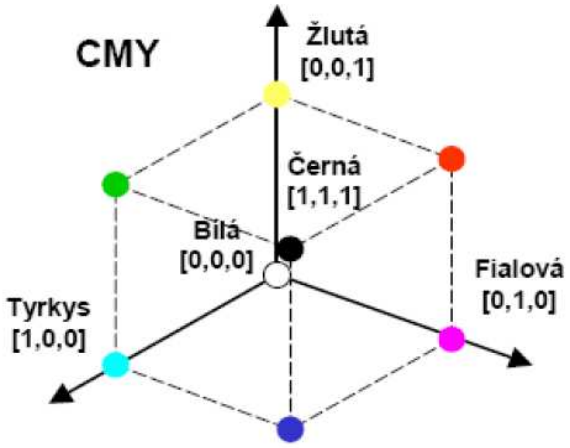
Obr. 3: Barevný model RGB. [2]

Formát CMYK

CMYK je barevný model založený na subtraktivním míchání barev. CMYK se používá především u reprodukčních zařízení, která barvy tvoří mícháním pigmentů (např. inkoustová tiskárna). Model obsahuje čtyři základní barvy:

Tyrkysová, fialová, žlutá. Jde o doplněk k formátu RGB proto lze provádět mezi formáty převody podle vzorce [1].

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2}$$

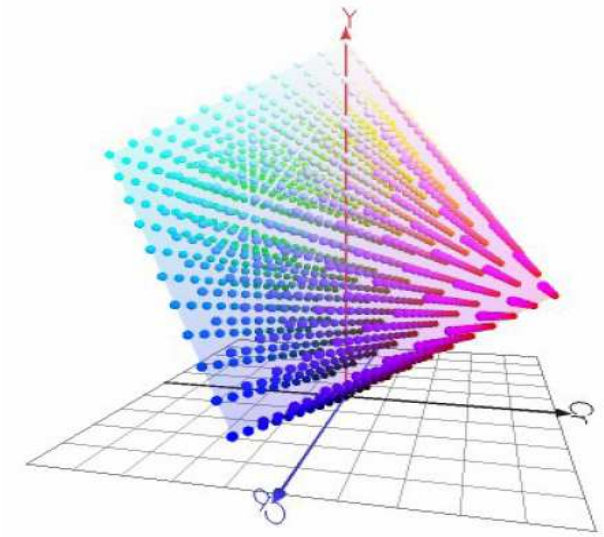


Obr. 4 :Jednotková krychle pro barevný model CMYK. [1]

Formát YUV

YUV je barevný model používaný v televizním vysílání v normě PAL i HDTV.

Model k popisu barvy používá tříprvkový vektor [Y,U,V] (někdy používané Y,Cb,Cr), kde Y je jasová složka a U a V jsou barevné složky. U je také někdy označováno jako B-Y a V odpovídá R-Y. Barevné složky se používají v rozsahu od -0.5 do +0.5, jasová složka má rozsah od 0 do 1.



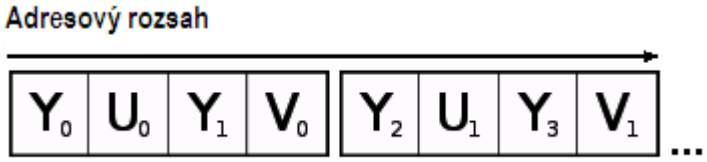
Obr. 5: Barevný model YCbCr. [2]

Bylo zjištěno, že člověk vnímá mnohem více jas, než barvu. Toho se vzhledem k oddělení jasové a barevné informace do složek dá lehce využít ke zmenšení datového toku při zachování vysoké obrazové kvality. Jasová složka je potom uložena v plném rozlišení, zatímco barevné složky U a V jsou uloženy v rozlišení menším a při zobrazení se potom musí zvětšovat. To způsobuje rozmazání barev v oblastech s prudkými barevnými přechody (hrany objektů, ...) jež je však většinou znatelné jen při bližším zkoumání.

Formát YUV se vyskytuje ve třech verzích barevného podvzorkování, a to 444, 422 a 420.

YUV 444 je formát, kde jasová i barevná složka má stejné rozlišení. Jednotlivé složky mohou být uloženy buďto prokládaně, nebo postupně celé roviny. V souborech, dostupných pro testování je použit druhý způsob.

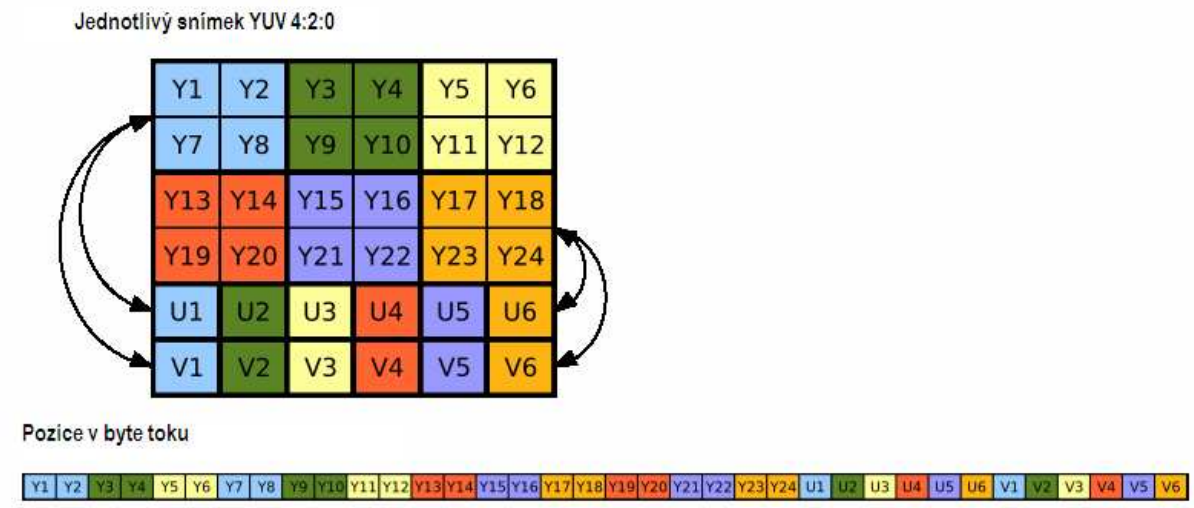
YUV 422 obsahuje barevné složky s poloviční horizontálním rozlišením. Na dva jasové vzorky tak vždy vychází jeden U a jeden V vzorek. Složky jsou standardně ukládány prokládáně, jak je znázorněno na následujícím diagramu:



Obr. 6: Datový tok formátu YUV 4:2:2. [6]

Testovací klipy však tento způsob uložení nedodržují, stejně jako u YUV 444 jsou uloženy postupně všechny vzorky Y, potom všechny U a nakonec všechny V.

Poslední variantou je YUV 420, obsahující barevné složky s poloviční rozlišením (horizontálně i vertikálně). Tentokrát jsou jednotlivé barevné roviny uloženy celé, postupně za sebou, jak znázorňuje následující obrázek:



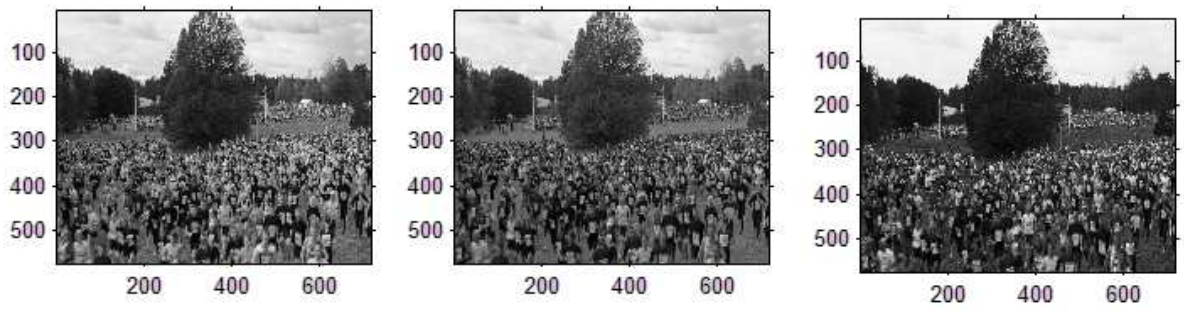
Obr. 7: Datový tok formátu YUV 4:2:0. [6]

.

Pro zpřesnění zpracování videosekvencí je potřeba převést videosekvenci z formátu barev RGB na formát YUV a to proto, že lidské oko je nejcitlivější na jas v obraze a méně citlivé na barvy, proto se mohou složky U a V podvzorkovat aniž by došlo k viditelným změnám v obraze. Způsob zobrazení v RGB a YUV modelu je možné vidět na obrázcích.

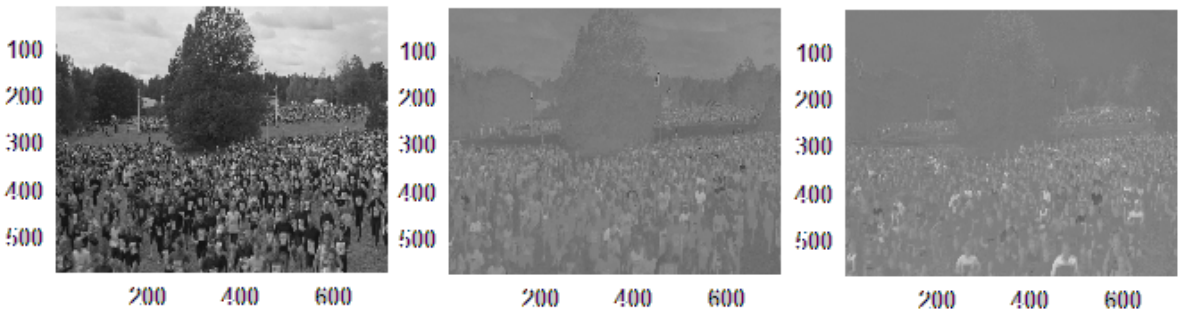
Obr. 8: Snímek RGB .

Na obrázku č.9 je zobrazen snímek rozložený na složky barevného modelu R,G,B. Z obrázku je velmi dobře patrné, jakou obrazovou obrazové informace nese každá složka.



Obr. 9: Snímek RGB rozložený na složky R, G, B.

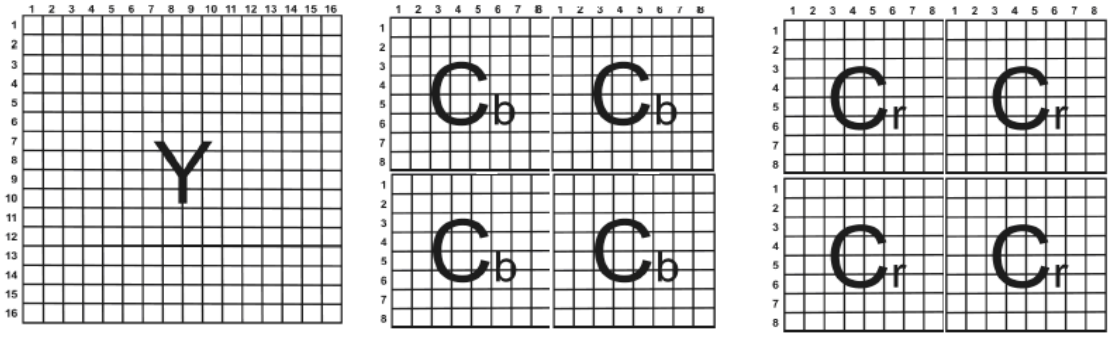
Na obrázku č.10 je zobrazen snímek rozložený na složky barevného modelu YUV. Z obrázku je velmi dobře patrné, jakou obrazovou hodnotu nese každá složka.



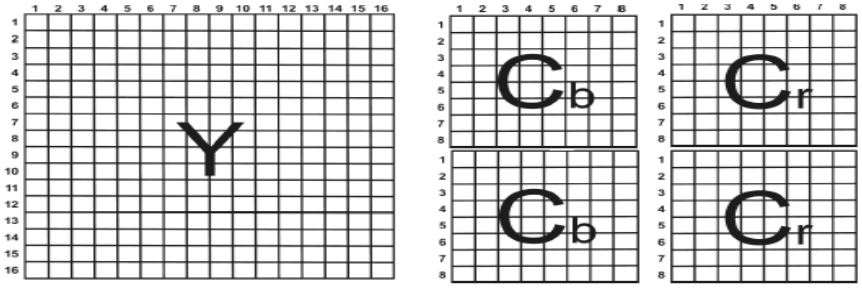
Obr. 10: Snimek YUV rozložený na složky Y, U, V

Formáty vzorkování a podvzorkování barev

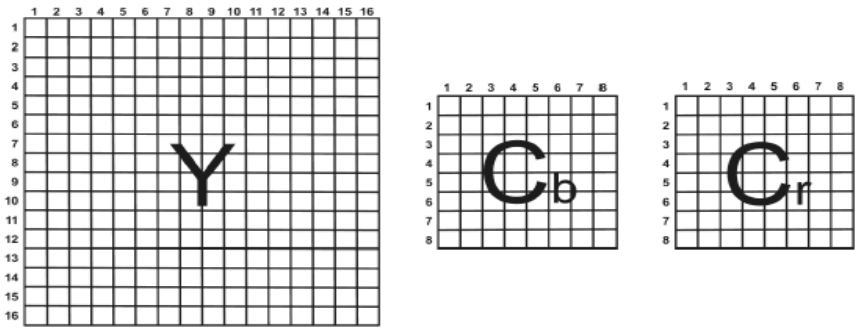
Podvzorkování barev redukuje množství informací u chrominančních složek obrazu. Princip podvzorkování je možné použít díky vlastnostem lidského oka. Na sítnici lidského oka je několik druhů senzorů. Jednotlivé druhy senzorů reagují na rozdílné podněty, některé snímají informace o světelnosti (jasu) obrazu, další jsou citlivé na různé barevné odstíny. Protože senzorů na jasovou složku je více než pro barevné složky, je lidské oko méně citlivé na barevné složky obrazu. Podvzorkování vytváří průměr ze dvou sousedních obrazových bodů na řádku popřípadě v sloupci, nebo ze čtyř sousedních obrazových bodů tvořících čtverec a ukládá se jako jediný pixel, tím dochází ke ztrátě části informace o barevné složce obrázku a zároveň k podstatnému zmenšení objemu dat. Pro snížení výpočtové náročnosti se často místo průměrování pouze vynechají některé obrazových body. Podvzorkování je ztrátová operace, tzn. že podvzorkované barevné signály již nelze obnovit do původní formy. Vzhledem k vlastnostem lidského zraku toto zkreslení nepůsobí rušivě a může být využito k podstatnému zmenšení datového toku.



Obr. 11: Formát vzorkování 4:4:4. [13]



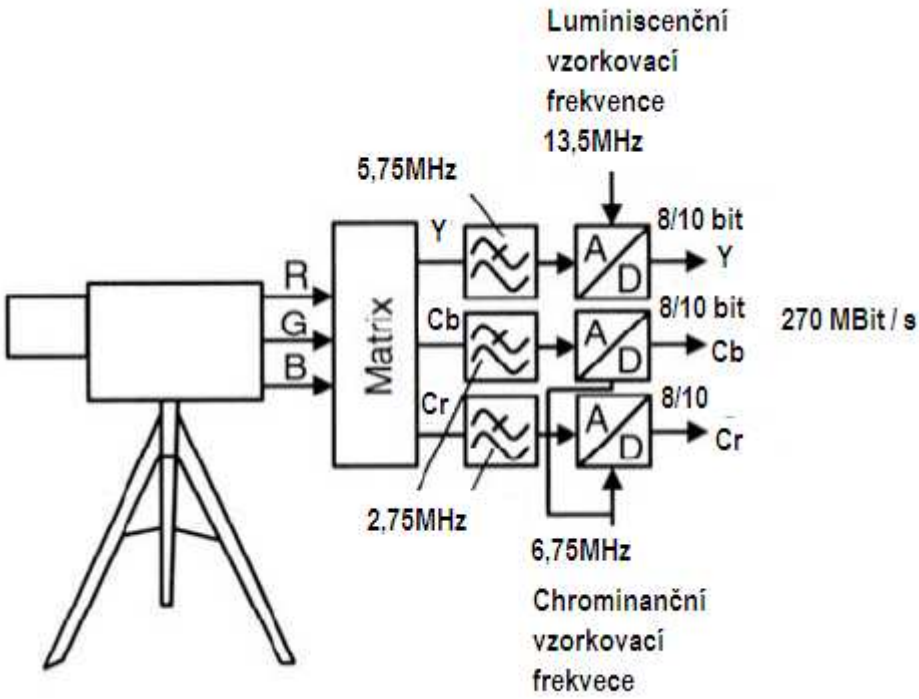
Obr. 12: Formát vzorkování 4:2:2. [13]



Obr. 13: Formát vzorkování 4:2:0. [13]

Digitální videosignál

Analogové R,G,B signály jsou maticovány v televizní kameře a tvoří luminanční (Y) a chrominanční signály (rozdílové barvy Cb a Cr) obr. 14. Signály jsou tvořeny součtem a rozdílem složek R,G,B.



Obr. 14: Digitální videosignál. [11]

Luminační signál je limitován v šířce pásma na 5,75 MHz (TV soustava PAL) a chrominanční signály na 2,75MHz, protože citlivost lidského oka je nižší než u jasové složky (využití u formátu vzorkování).

Převod barev

Před samotným zpracováním obrazu je nutné upravit vstupní obrazová data do předem definovaného tvaru. Zejména je nutné data převést z nejrůznějších barevných prostorů jako jsou RGB a dalších do barevného prostoru YCbCr. Složka Y nese informaci o jasu obrazu, složky Cb a Cr jsou rozdílové chrominanční složky obrazu vzhledem k jasové složce Y. Tento formát vznikl pro přenos video signálu zpětně kompatibilního s černobílým obrazem.

Transformace barev z RGB do YUV

Pro převod z barevného prostoru RGB (Red Green Blue) do prostoru YCbCr lze použít následující vzorce. Pro rozsah 0 až 255 (pro 8 bitové binární vyjádření). [13]

$$\begin{aligned} Y &= 0{,}2990\ R + 0{,}5870\ G + 0{,}114\ B \\ C_b &= -0{,}1687\ R - 0{,}3313\ G + 0{,}5\ B + 128 \\ C_r &= 0{,}5\ R - 0{,}4187\ G - 0{,}0813\ B + 128 \end{aligned} \tag{3}$$

Opačný převod z Y U V na RGB se provádí pomocí těchto vzorců: [13]

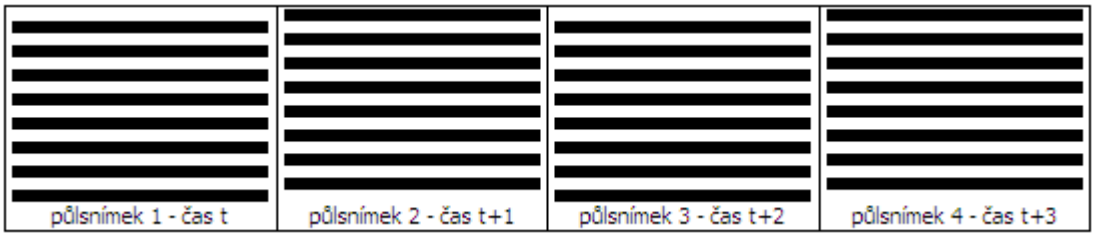
$$\begin{aligned} R &= Y + 1{,}4020\ (C_r - 128) \\ G &= Y - 0{,}3441\ (C_b - 128) - 0{,}71414(C_r - 128) \\ B &= Y + 1{,}772\ (C_b - 128) \end{aligned} \tag{4}$$

Přepočet barevného modelu YcbCr z modelu RGB pomocí matice [13]

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 65{,}481 & 128{,}553 & 24{,}966 \\ -37{,}797 & -74{,}203 & 112 \\ 112 & -93{,}786 & -18{,}214 \end{pmatrix} x \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{5}$$

3. Základy prokládání – interlacing

Prokládání, využívající techniku vyvinutou pro CRT televize tvoří obraz o 576 (soustava PAL) viditelných řádků. Jde o střídání sudých a lichých řádků a střídavé obnovení s frekvencí 50 snímků za sekundu. Prokládání bylo zavedeno kvůli omezení přenosu informace za současného zachování plynulosti videa a jeho rozlišení. Využívá se nedokonalosti lidského oka, které není schopné zachytit malé a rychlé změny v obraze. Systém vznikl při tvorbě TV norem PAL a NTSC. Původně mělo prokládání dva významy. Tím prvním, který už dnes ztratil smysl, bylo zamezení blikání obrazu na starých televizních přijímačích. Druhým důvodem, který se uplatňuje dodnes, je vytváření iluze zobrazování 50 fps (snímků za vteřinu), přestože jich je ve skutečnosti pouze 25. Moderní televizory sice zpracovávají obrazový signál s řádkovým prokladem, ale dva po sobě jdoucí půlsnímky se dají složit do kompletního neprokládaného snímku.



Obr. 15: Vykreslování půlsnímků. [7]

Proč se prokládáním zabývat

Zobrazovat obraz prokládaně dokáží zobrazit jen klasické CRT obrazovky. Plasmové zobrazovače, LCD panely, projektory a počítačové monitory zobrazují vždy celý obraz (snímek) najednou. CRT televize zobrazují obraz způsobem dvou půlsnímků místo jednoho celého (časový rozdíl a to způsobem řádek 1,3,5 – jedno pole a 2,4,6 druhé pole) progresivní řádkování.

Plasmové televize, projektory a počítačové monitory zobrazují vždy celý obraz (snímek) najednou. CRT televize zobrazují obraz způsobem dvou Half – rámců místo jednoho celého (časový rozdíl a to způsobem řádek 1,3,5 – jedno pole a 2,4,6 druhé pole)

V počítači se ale neukládají jednotlivé půlsnímky zvlášť, ale vždy dva půlsnímky se spojí do jednoho celého snímku - progresivní řádkování a tím nenastává zde problém “blikání“ obrazovky. Pro zabezpečovací aplikace může být tato vlastnost progresivního řádkování důležitá, protože umožňuje sledovat detaily (záběr pohybujících se objektů – utíkající osoba). Protože, ale půlsnímky, ze kterých je celý snímek složen, nebyly u prokládaného videa pořízeny v jednom čase, liší se v nich poloha pohybujících se objektů (vznikají artefakty v obraze) zobrazením takovýchto obrazů na plasmových televizích, projektorech a počítačových monitorech.

Obr. 16: Vykreslování půlsnímků. [7]

Obr. 17: Efekt prokládání zobrazením na monitorech. [5]

4. Konverze neprokládaného videa na prokládané

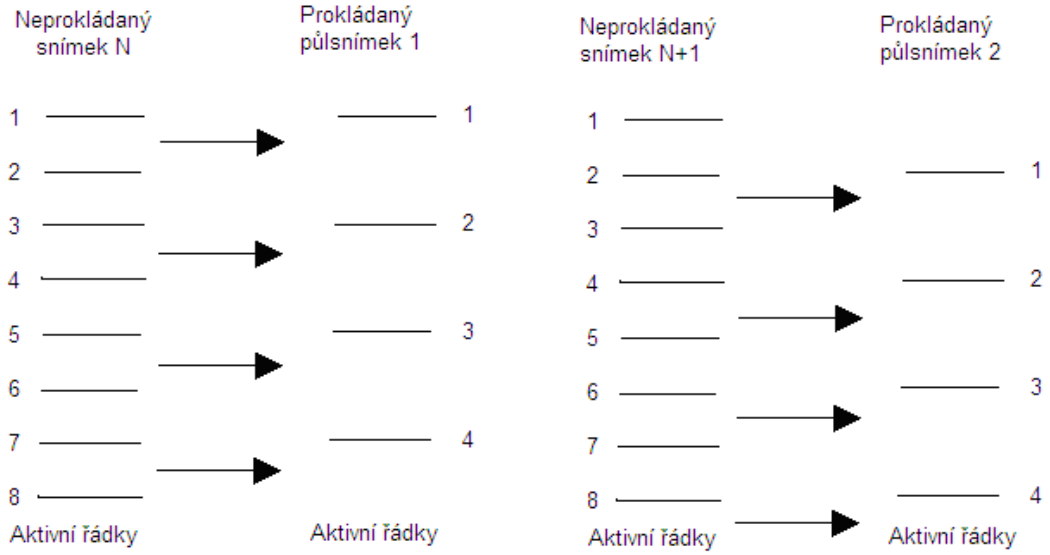
V některých aplikacích je potřeba zobrazovat neprokládané video na displeji, který dokáže zobrazit pouze video prokládané. Proto musíme použít nějakou formu konverze z neprokládaného na prokládané video.

Tato konverze musí být vykonána s videem ve složkovém stavu (RGB, nebo YCbCr). Kompozitní video signál (NTSC nebo PAL) nemůže být přímo zpracován díky přítomnosti fázové informace barevné subnosné, která by byla bezvýznamná po tomto zpracování. Tyto signály musí být převedeny na složkové ještě před konverzí.

Decimace řádků (Scan line decimation)

Tento nejjednodušší postup spočívá ve vypuštění ostatních aktivních řádků z neprokládaného videa. Tento postup však přináší problémy na vršcích a spodcích objektů. Pokud je v obraze ostrý přechod v barvě nebo intenzitě, tak tento postup bude mít za následek třepotání nebo mizení této hrany. Toto třepotání bude mít poloviční frekvenci než je snímková frekvence. Pro příklad: pokud bude hrana silná jeden řádek, tak díky decimaci bude zobrazena pouze na polovině snímků a bude tak blikat. Hrana široká dvě linky se bude třepotat nahoru a dolu.

Jednoduchá decimace také může vnést do obrazu aliasingové artefakty. Tyto nemusejí být viditelné, ale můžou ztížit následné zpracování obrazu.



Obr. 18: Decimace řádků (Scan Line Decimation).

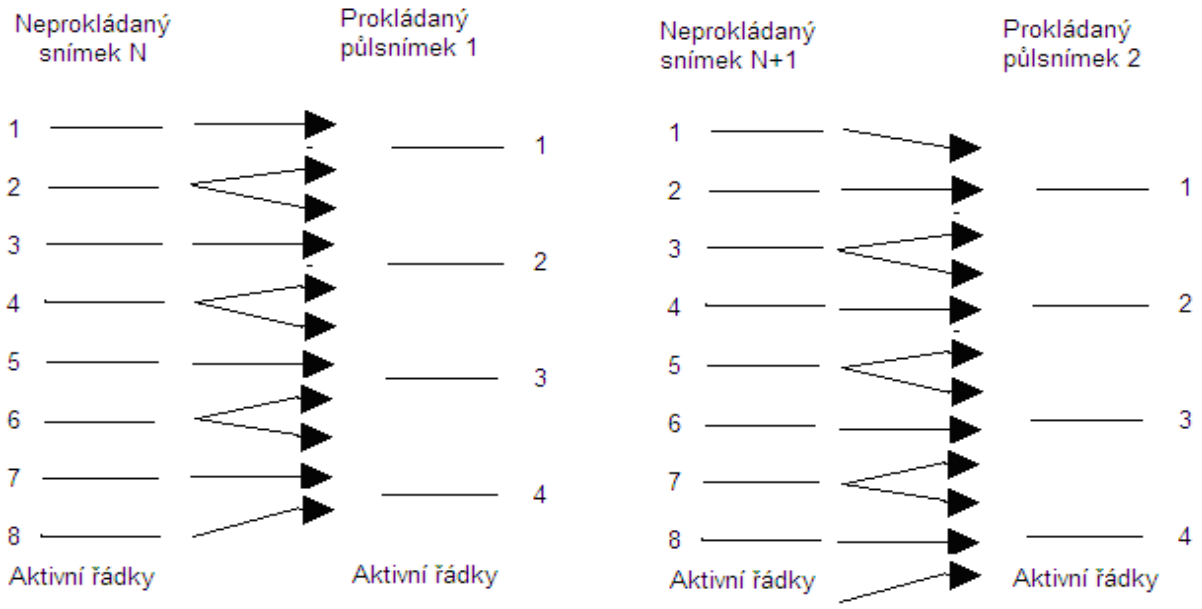
Vertikální filtrace (Vertical filtering)

Lepším řešením je použít dvou (nebo více) řádků neprokládaného videa k vytvoření jednoho řádku videa prokládaného. Rychlý vertikální přechod je pak rozložen do několika prokládaných řádků.

Typicky se výsledný řádek skládá ze 3, které jsou vynásobeny koeficienty 0.25, 0.5, 0.25. Použití více než 3 řádků však vede k rozostření a malý text se stává obtížně čitelným.

Alternativou je použití filtrů IIR spíše než filtrů FIR. Ty k obyčejnému „průměrování“ přidávají redukci zesvětlení kolem objektů a redukují poblikávání.

Nesmíme také zapomenout, že na začátku a konci snímku je pro filtraci k dispozici omezený počet řádků a s tím musíme při filtraci počítat.



Obr. 19: Vertikální filtrace (Vertical filtering).

5. Konverze prokládaného videa na neprokládané

V některých aplikacích je potřeba naopak zobrazovat prokládané video na displeji, který dokáže zobrazit pouze video neprokládané. Tento postup je inverzní k předchozímu a je nazýván Deinterlacing nebo progresivní konverze. Opět platí, že operace musejí probíhat se signálem ve složkovém tvaru (RGB nebo YCbCr) nikoli s kompozitním signálem (NTFS nebo PAL). Každý signál tedy musí být do tohoto složkového tvaru převeden.

Algoritmy vyvíjené pro převod prokládaného videa zpět na neprokládané video, můžeme rozdělit na Intra-Field a Inter-Field. Intra-field využívají vysokou korelaci mezi obrazovými body v jednom snímku. Obecně nevyžadují vysoký hardwarový výkon. Důležité je vybrat tu správnou metodu pro konkrétní scénu, která povede k nejlepším výsledkům.

5.1 Zpracování v rámci půlsnímků (Intra- Field Proccesing)

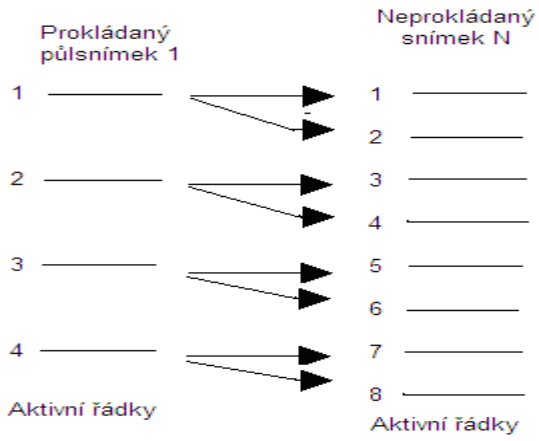
Nejjednodušší metodou pro generování dodatečných řádků při použití původního snímku je nazývána také jako metoda *bob*. Existují dvě techniky jak toho dosáhnout:

Zdvojení řádků (Scan Line Duplication)

Tato technika jednoduše duplikuje předchozí aktivní řádek. Počet aktivních řádků je zdvojnásoben, ale vertikální rozlišení je stejné - **Weave** algoritmus. Ten algoritmus zachovává rozlišení v prostoru, avšak nesedí v čase. Proto není vhodný pro sekvence, ve kterých se vyskytují pohybující se předměty. Je popsán touto rovnicí: [17]

$$f_{OUT}(\vec{x},n) = \begin{cases} f(\vec{x},n), & y \bmod 2 = n \bmod 2 \\ f(\vec{x},n-1), & \text{else} \end{cases} \qquad (6)$$

Opět ovlivňuje liché řádky v lichých snímcích, nebo sudé řádky v sudých snímcích, zatímco řádky s opačnou polaritou kopíruje vždy z předchozího snímku. Vzorec (6) popisuje, že sudé řádky v lichých snímcích necháme bez zpracování a na pozici sudých řádků zkopírujeme liché řádky.



Obr. 20: Zdvojení řádků (Scan Line Duplication).

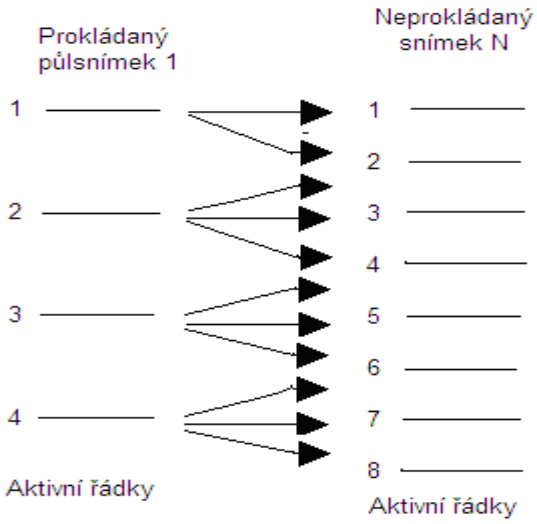
Interpolace řádků (Scan Line Interpolation)

Aktivní řádek je vytvořen interpolací předcházejícího a následujícího řádku. I když počet aktivních řádků je dvojnásobný, rozlišení se nemění. Nejjednodušší způsob je ukázán na obr. 21. [6]

Tato technika se nazvaná **bob**. Ta zachovává rozlišení v čase, avšak snižuje rozlišení v prostoru. Funguje dle následujícího vzorce: [17]

$$f_{OUT}(\vec{x},n)=\left\{\begin{array}{l} f(\vec{x},n), \\ \dfrac{f(\vec{x}-1,n)+f(\vec{x}+1,n)}{2} \end{array}\right\},\begin{array}{l} y \bmod 2 = n \bmod 2 \\ else \end{array} \tag{7}$$

Ovlivňuje tedy liché řádky v lichých snímcích, nebo sudé řádky v sudých snímcích. Řádek, proložený z minulého snímku přitom nahradí průměrem řádku nad ním a pod ním jak popisuje vzorec (7).



Obr. 21: Interpolace řádků (Scan Line Interpolation).

Lepší výsledky dosáhneme pokud k interpolaci použijeme FIR filtr.

Variable Interpolation

V některých (ne příliš častých) případech není periodicita mezi počtem vstupních a výstupních řádků. V tomto případě je, teoreticky, nutný nekonečně velký počet filtrovacích cyklů a koeficientů. V praxi to není možné, ale řešením je použití velkého, ale konečného počtu filtrů. Jejich počet potom určuje přesnost interpolace. Tato technika kombinuje odstranění prokladu a změnu vertikálního rozlišení v jednom procesu.

5.2 Zpracování několika půlsnímků najednou (Inter- Field Proccesing)

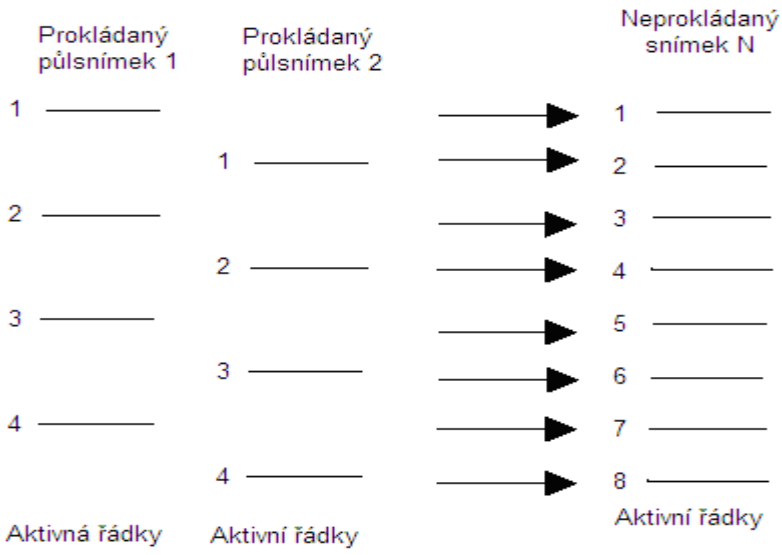
Při této metodě je video-informace získána z více než jednoho snímku. Umožňuje vyšší vertikální rozlišení než při zpracovávání jednoho snímku.

Sloučení půlsnímků (Field Merging)

Tato metoda slučuje dva prokládané snímky do jednoho neprokládaného. Aktivní řádky předcházejícího snímku jsou vloženy do snímku aktuálního. Jednoduchá na aplikaci, tak tato metoda přináší zdvojnásobení rozlišení v oblastech bez pohybu.

Objekty v pohybu budou mít artefakty, zvané také jako *Combing*, díky časovému rozdílu mezi snímky. Pokud spojíte takovéto snímky s posunem, tak každý objekt vytvoří dvojitý obraz.

Je běžné zjemnit obraz ve vertikální rovině kvůli viditelné redukci Combiningu (sníží se vertikální rozlišení a chvění obrazu.



Obr. 22: Sloučení půlsnímků (Field Merging).

Adaptivní odstranění prokladu (Motion Adaptive Deinterlacing)

Dobrým řešením je použití *field merging* pro oblasti bez pohybu a *scan line* pro oblasti s pohybem. Je proto nutné detekovat pohyb v obraze v reálném čase. Nutné je mít k dispozici několik obrazů, které jsou za sebou.

Při kombinaci 2 snímků, je rozlišení zachováno tam, kde je oko nejcitlivější na detaily. Rozdíly v obou snímcích jsou ohodnoceny od 0 (beze změny) až do maxima (např. přechod od bílé k černé). Podle toho se vybere vhodnější způsob.

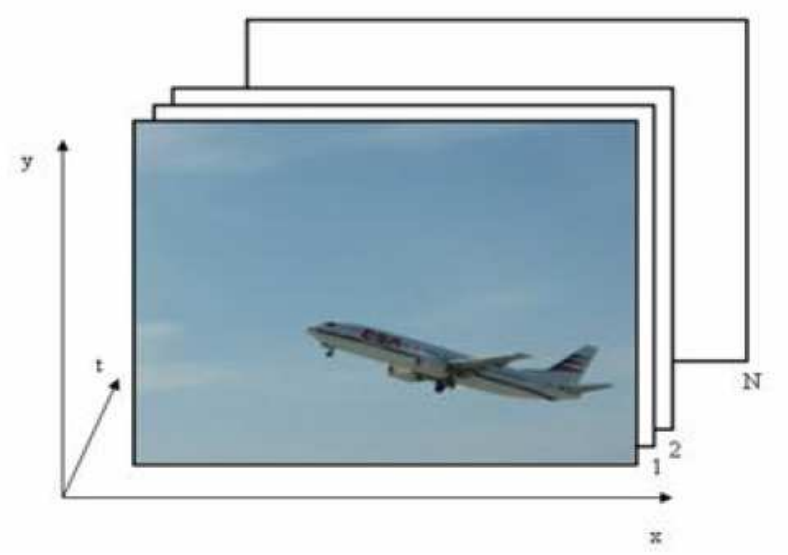
Adaptivní algoritmus, spojující výhody popsaných metod Weave a bob se nazývá VTMF: [17]

$$f_{OUT}(\vec{x},n)=\begin{cases} f(\vec{x},n), & \left.\begin{matrix} \\ \end{matrix}\right\} y \bmod 2 = n \bmod 2 \\ median\left(f(\vec{x}-1,n)+f(\vec{x}+1,n),\, f(\vec{x},n-1)\right) & \left.\begin{matrix} \\ \end{matrix}\right\} else \end{cases} \tag{8}$$

Tenhle způsob je označován jako „per pixel“. Ohodnocují se jednotlivé oblasti a je dost náročný, a tedy i nákladný, na hardware. Proto se používá metoda zvaná „Per Field“, která hodnotí snímek jako celek. I když má tato metoda horší výsledky, tak je pořád lepší než vertikální interpolace.

5.3 Odstranění prokladu založené na pohybujících se objektech

Dynamické obrazy jsou v podstatě posloupnosti statických obrazů. Ze statického obrazu mžeme získat informaci o poloze objektů v obraze, ale nezjistíme nic o směru, rychlosti či zrychlení pohybu. Tento problém se řeší posloupností statických obrazů.



Obr. 23: Dynamický obraz. [4]

Tato metoda se nezaměřujeme na bloky o určité velikosti např. [4x4] , ale na odhadu pohybu objektů - skupinu obrazových bodů, které mají stejný pohybový vektor v daném regionu, které ale nemusejí nutně náležet ke stejnému fyzickému objektu.

Nejdříve musíme segmentovat objekty z pozadí. Obrazové body musejí splňovat 3 parametry: objekt složený z těchto bodů musí být v pohybu, spojené a musejí mít stejný pohybový vektor. Bohužel, ve většině případů není pohybový vektor známý, pokud neprovedeme odhad pohybu „problém podobný jako určení jestli bylo dřív vejce nebo slepice“. Pokud použijeme iterativní (postupnou) metodu, tak objem výpočtů bude příliš veliký (příliš mnoho objektů). Metody založené na příliš velké přesnosti pohybového vektoru vždy vedou na příliš mnoho objektů.

Odstranění prokladu s kompenzací pohybu (Motion - Compensated Deinterlacing)

Motion-compensated je metoda pro odstranění prokladu více komplexnější, než metoda předcházející a často používána v nástrojích pro převod videa.

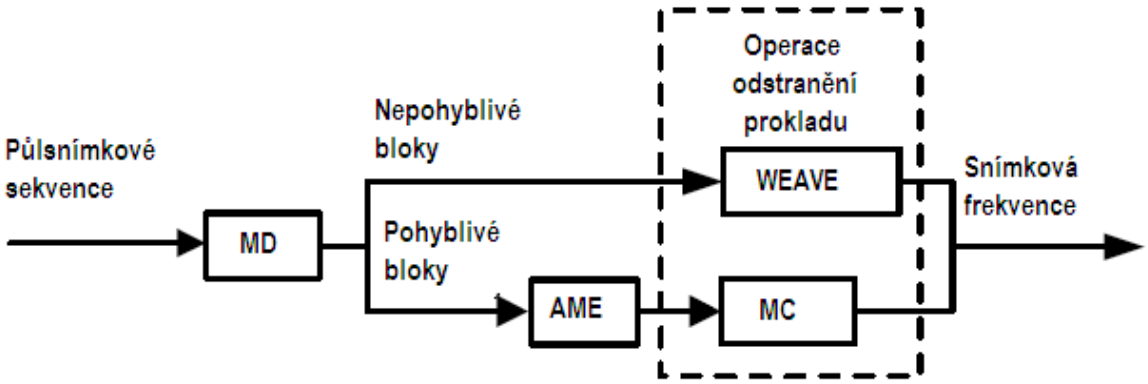
Zpracování při kompenzaci pohybu vyžaduje výpočet pohybového vektoru objektu mezi snímky. Oblast obrázku je pokrývaná objektem tak, jak se přes něj pohybují jednotlivé objekty. Tyto vektory musí také být přesné na úrovni subpixelů a rozlišené ve dvou temporálních směrech mezi snímky.

Chyby v pohybovém vektoru budou mít za následek artefakty v obraze.

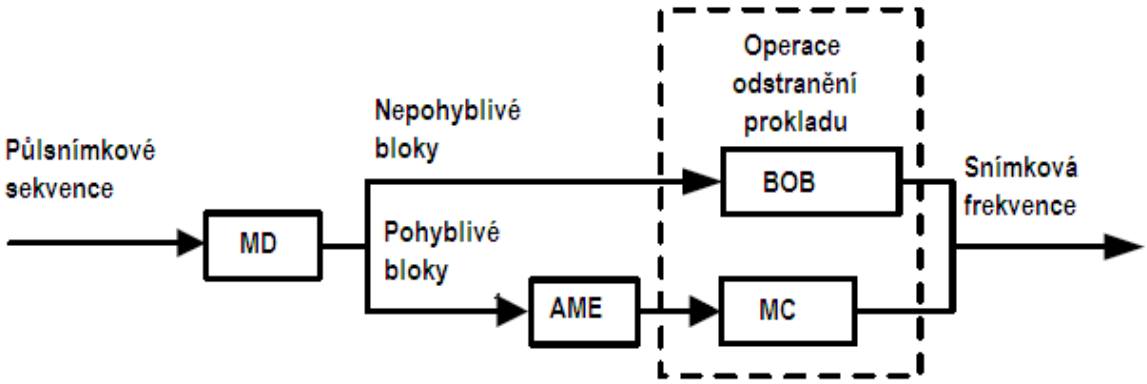
Adaptivní MC algoritmus založený na časoprostorové korelaci *(AMC-STC)*

Ve videosekvencích se vyskytuje spousta statických bloků. Pro statické bloky je metoda Weave nejméně náročná na výpočet metoda ale nezachovává rozlišení v čase. Zatímco metoda bob je na tomto lépe. Pro pohybující se bloky je vhodná metoda MC. To vede k úvaze rozdělit řešení pro statickou a pohybující se část. Je tedy nezbytná detekce pohybu. Adaptivní MC algoritmus založený na časoprostorové korelaci (AMC-STC) obsahuje tři kroky,

- Detekce pohyb bloků(MD)
- Adaptivní odhad pohybového vektoru založený na časoprostorové korelaci (AME)
- Vlastní odstranění prokladu



Obr. 24: Algoritmus pro odstranění prokladu varianta1. [17]

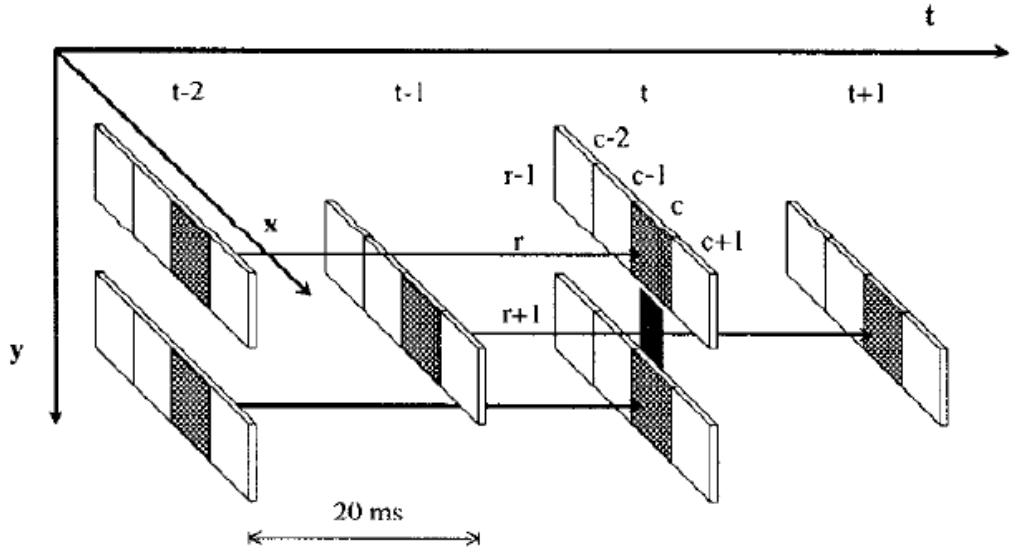


Obr. 25: Algoritmus pro odstranění prokladu varianta2. [17]

Odstranění prokladu pomocí metody – MC (Motion - Compensated)

Jak je vidět z ilustrací obr. 26:, je potřeba mít v paměti čtyři snímky a z jejich diference je možné spočítat, zda pixel odpovídá pohybujícímu se objektu. Funkce F, použitá autory vrací největší z jejích argumentů (maximum). Detekce pohybu jde zpřesnit použitím porovnávání větších bloků namísto obrazových bodů, nebo následným morfologickým zpracováním detekované masky pohybu. [16]

$$d_3(r,c,t) = F\left\{\begin{array}{l} |p(r-1,c,t) - p(r-1,c,t-2)|, \\ |p(r,c,t+1) - p(r,c,t-1)|, \\ |p(r+1,c,t) - p(r+1,c,t-2)| \end{array}\right\}, \tag{9}$$



Obr. 26: Zpracování snímků metodou MC. [16]

Pixely, pro něž byl detekován pohyb jsou dále zpracovávány, je třeba zjistit vektor pohybu. K tomu se používá porovnávání bloků. Ve snímku v čase t je pixel, pro který potřebujeme zjistit vektor pohybu. Porovnáváme blok intenzit okolo daného obrazového bodu s bloky v obraze z času t-1. Neprohledáváme celý obraz, ale jen malé okolí. Zde je možné použít predikci pohybu, založenou na předchozích snímcích, nebo na souslednosti pohybu blízkých obrazových bodů. Ta však není použita a je tedy nutné prohledávat všechny relevantní pozice okna, což je pomalé, avšak k demonstračním účelům to postačí. Metrika, použitá pro porovnávání bloků se nazývá Mean Pixel Difference Classification (MPCD): [17]

$$MPDC(\vec{X},n)=\frac{1}{WxH}\sum_{W=1}^{W}\sum_{H=1}^{H}ord(\left|I_X(w,h,n)-I_X(w,h,n-1)\right|\le\alpha) \tag{10}$$

Funkce ord je definována jako 1 pro pravdivý argument a 0 pro nepravdivý. Pokud se podaří nalézt posunutý blok, je možné zkopírovat obrazové body pro liché / sudé řádky a odstranit tak prokládání. Pokud se blok najít nepodaří, je potřeba použít jiný algoritmus, například bob. Celý proces je shrnut jako: [17]

$$f_{OUT}(\vec{X},n)=F\left\{\begin{array}{l} f(\vec{X},n), \quad y\bmod 2=n\bmod 2 \\ f(\vec{X},n-1),(d(\vec{X},n))\,a \quad y\bmod 2\neq n\bmod 2 \\ f\left(\vec{X}+\left(\begin{matrix}0\\1\end{matrix}\right)_{WxH},n\right),(d(\vec{X},n)=0)\,a \quad y\bmod 2\neq n\bmod 2 \quad a \quad match(\vec{X},n)=0, \\ median\left\{\begin{array}{l} f\left(\vec{X}+\left(\begin{matrix}0\\1\end{matrix}\right)_{WxH},n\right) \\ f\left(\vec{X}-\left(\begin{matrix}0\\1\end{matrix}\right)_{WxH},n\right) \\ f\left((\vec{X}-v(\vec{X},n),n-1\right) \end{array}\right., else \end{array}\right\}, \tag{11}$$

Detekce pohybu objektů (Motion Object Detection (MOD))

Dokud neodhadneme pohybové vektory, není možné určit, které obrazové body patří ke konkrétnímu regionu. Předpokládáme, že pohybující se objekty jsou homogenní. Potom můžeme použít metody pro prostorovou segmentaci.

Transformace Watershed je pro toto použití velmi vhodná. Watershed má však tendenci k přesegmentování, proto je nutný Post-Procesing.

Na dva snímky za sebou se stejnou paritou použijeme Wateshed a provádí se korelace jednotlivých objektů z prvního snímku s objekty daného regionu z druhého snímku. Místo s největší korelací je nová pozice objektu. Pokud se liší od původní pozice v prvním snímku tak můžeme vypočítat pohybový vektor. Pak se na oblasti kde nebyl zjištěn pohyb použije metoda Bob a na oblasti s pohybem se použije metoda Weave.

Predikce pohybu (Motion Estimation (ME))

Motion Estimation je časová predikce používaná v technice MPEG video a je založena na odhadu pohybu. Základní artefaktem v pohybovém odhadu je, že ve většině případů jsou budoucí snímky videa podobné s předchozími, s výjimkou změn vyvolaných objekty pohybující se v rámci snímků. V jednoduchém případě pokud by byl nulový pohyb mezi snímky, je snadné účinně předvídat aktuální snímek jako duplikát predikce snímku. Touto predikcí se účinně urychlí výpočty v rámci odstranění prokladu videa.

5.4 Způsob vyhodnocení kvality obrazu

V případě použití metod pro konverzi prokládaného na neprokládané video dochází k nevratné ztrátě části původní informace, především při použití formátu vzorkování 4:2:2 a 4:2:0. Tuto ztrátu lze kvantifikovat několika způsoby.

Střední kvadratická chyba MSE(Mean square error)

Jedná se o velmi často používanou metodu popisující změnu obrazu oproti obrazu originálnímu. Není nejvýhodnější z hlediska subjektivní kvality obrazu, kdy se i obraz s velkou MSE může jevit stejně kvalitně jako originál.

MSE je dána vztahem

$$MSE_{[-]} = \frac{1}{M \cdot N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \left(x(m,n) - \hat{x}(m,n) \right)^2 \tag{12}$$

Kde

$\hat{x}$ - jsou obrazové body (pixely) původního obrazu

$\hat{x}$(m,n) jsou obrazové body rekonstruovaného (výstupního) obrazu

M,N jsou rozměry obrazu

Špičkový poměr signál – šum – PSNR (Peak Signál to Noise Ratio)

Vychází ze střední kvadratické chyby MSE a je :

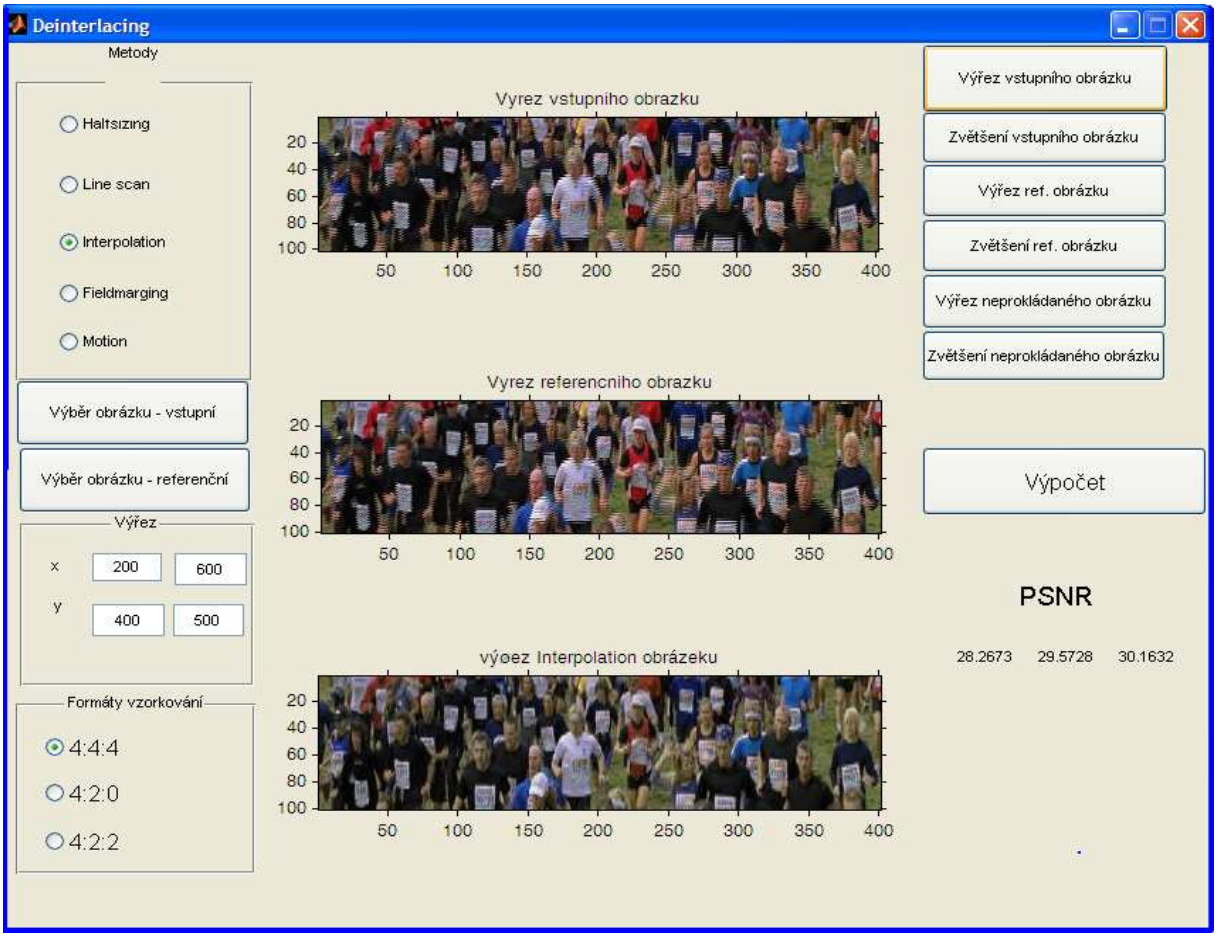
$$PSNR_{[dB]} = 10 \cdot \log_{10} \left(\frac{255^2}{MSE} \right) \tag{13}$$

Subjektivní hodnocení

K posouzení kvality se využívá i metod subjektivních, jejich konkrétní realizace vychází z hodnocení výsledného obrazu statisticky dosti velkou skupinou lidí.

6. Implementace algoritmu v prostředí MATLAB

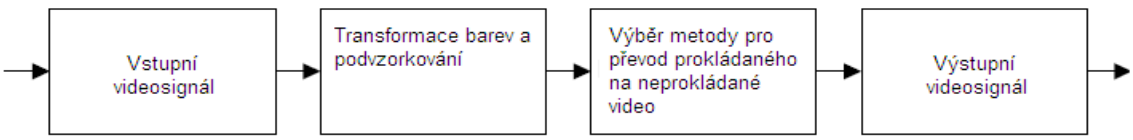
6.1. Funkce programu



Obr. 27: Úvodní menu v Matlabu.

Pro ověření platnosti vztahů, pro odstranění prokladu byla vytvořena jednoduchá aplikace v prostředí Matlab v grafickém prostředí GUI. Jsou načítány už vytvořené videosekvence (snímek složen ze dvou půlsnímků). U jednoduchých algoritmů využívám načítání jen jedné videosekvence [14] a u algoritmů Field Merging (sloučení půlsnímků) a Motion Compensation (pohybová kompenzace) využívám dvě videosekvence.

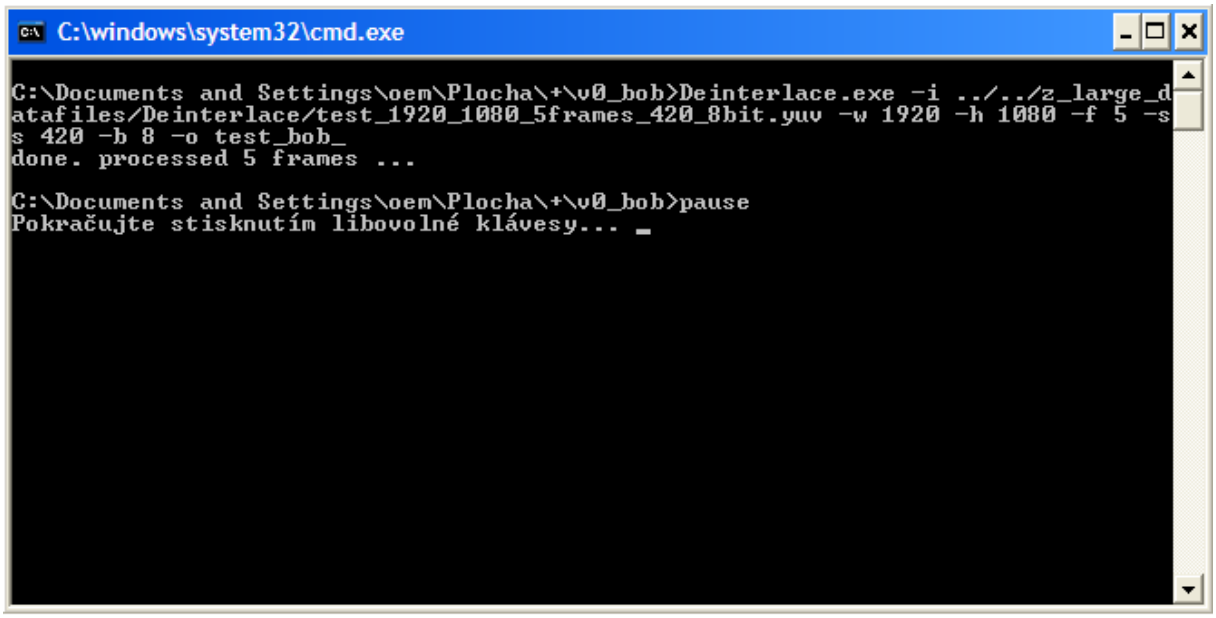
Nejprve se vybere metoda konverze prokládaného na neprokládané video. Poté se načte vstupní videosekvence u algoritmů Field Merging a Motion compensation načítám dvě vstupní videosekvence s referenčním. Vstupní obraz se převede do barevného prostoru YCbCr, pro lepší zpracování, protože hodnoty Cb a Cr složky nemají takové rozestupy jako R,G,B složky. Výstupem je neprokládané video a vyhodnocení kvality signálu. Možnost je i zvětšení části vstupního, referenčního a výstupního obrazu.



Obr. 28: Blokové schéma.

7. Implementace algoritmu v prostředí C

7.1. Funkce programu



Obr. 29: Ukázka aplikace v jazyce C++.

Program demonstruje čtyři jednoduché algoritmy pro odstranění prokládání obrazu a to metody BOB WEAVE, VTFM a odstranění prokladu pomocí MC (Motion - Compensated).

Navržené algoritmy pro odstranění prokladu byly otestované pomocí jazyka C, kde byla vytvořená konzola pro odstranění prokladu videosekvencí.

Vstupní videosekvence byly použity v rozlišení 720x576 a 1920x 1080. [12] Tento velký rozdíl v rozlišení zaručil kvalitnější subjektivní vyhodnocení výsledného progresivního obrazu a ukázky rozdílu.

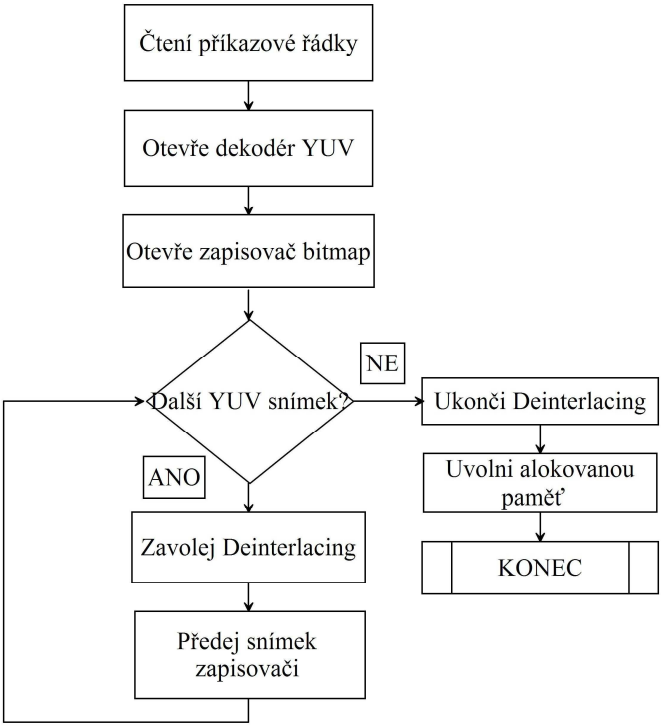
Program načítá prokládané snímky z videosekvence ve formátu yuv a ukládá filtrované snímky ve formátu bmp. Formát yuv neobsahuje žádné informace o velikosti, či počtu snímků a tak se tyto informace musí programu zadávat. Program se volá z příkazové řádky s následující syntaxí:

deinterlace -i <vstupni-soubor> -w <sirka> -h <vyska>
-f <pocet-snimku> -ss <subsampling>
-b <bitova-hloubka> -o <vystupni-soubor>

Kde sirka, vyska jsou rozměry obrazu v pixelech, subsampling je buďto 444, 422 nebo 420 (vysvětleno dále) a bitova-hloubka je buď 8 nebo 16. Program potom zpracovává jednotlivé snímky, na obrazovku se tiskne kolikátý snímek je právě zpracováván.

Hlavní program main prochází příkazovou řádku a porovnává jména parametrů v příkazové řádce, jestli jsou vstupní data správně zadaná. Dále kontroluje jestli jsme zadali vstupní a výstupní soubor a zavolá YUV dekodér a prochází všechny snímky a ukládá je do bitmapy. Poté program demonstruje čtyři jednoduché algoritmy pro odstranění prokládání obrazu, kde výsledné filtrované snímky ukládá ve formátu bmp.

Program main je demonstrován ve vývojovém diagramu níže.



Obr. 30: Vývojový diagram hlavního programu main .

7.2. Úprava a převod vstupní videosekvence

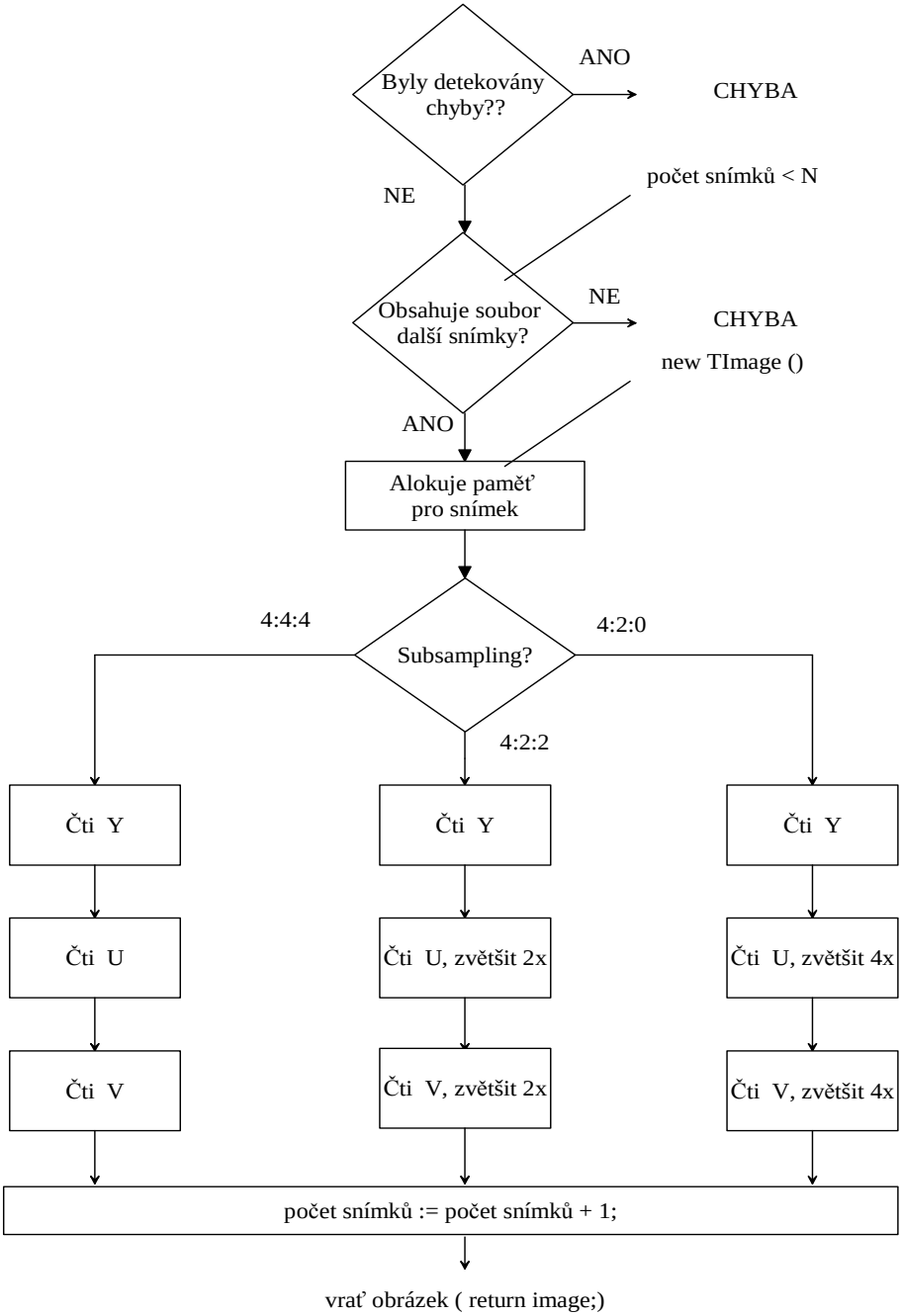
Program pro vytvoření YUV formátu z videosekvencí formátu sgi byl použit program pro tuto konverzi a to sgi2yuv.exe. [22]

Konfigurace programu sgi2yuv:

```
FramesToBeConverted = 5          # Počet snímků které mohou být konvertovány
SourceWidth         = 720        # počet pixelů na šířku
SourceHeight        = 576        # počet pixelů na výšku
InputFilePrefix     = ""         # předpona  vstupního souboru
InputFileSuffix     = ".sgi"     # přípona vstupního souboru
InputFileFirstFrame = 0355       # číslo prvního snímku
Count_digits        = 5          # Počet výstupních sekvencí
OutputFile          ="test_720_576_5frames_420_8bit"
                                 # Název výstupního souboru
Color_subsampling   = 2          # 0: 4:4:4, 1: 4:2:2, 2: 4:2:0
YUV_Format          = 0          # 0: (jeden soubor na sekvenci)
BitDepth_out        = 8          # počet bitů
Stuffing16bit       = 1
```

YUV dekodér

Další krok programu je YUV dekodér, který byl vytvořen pro vytvoření obrazu k jeho odstranění prokladu. YUV dekodér načte všechny snímky, prochází je a čte jednotlivé složky formátu YUV a ukládá je do image. Pro odstranění prokladu je použit formát RGB a proto je potřeba překonvertovat formát YUV na RGB viz kapitola 2.2.



Obr. 31: Vývojový diagram programu pro YUV dekodér

7.3. Použité algoritmy

Barevný formát je YUV 4:2:0, Y je jasová složka a U,V barvonosné složky. Barevná hloubka je 8 bitů na pixel pro každou složku. Navzorkování v poměru 4:2:0 znamená, že barevné složky jsou vůči jasové v poměru 4:1 a barevná složka tedy obsahuje je čtvrtinu bodů vůči jasové složce (na 4 jasové složky připadá jediný barevný).

Odstranění prokladu metodou – BOB

Navržený algoritmus využívá jednoduchého průměrování předchozího a následujícího řádku. Na začátku algoritmu se počítá kolikátý máme snímek a zkontroluje se zda máme správný formát. Program postupuje po řádcích a to buď po sudých nebo po lichých (záleží na snímku jestli je lichý nebo sudý).

Vybere se vrchní a spodní řádek a prochází se celý půlsnímek a vypočítává se průměr z těchto řádků.

```
const uint8_t *p_upper = p_image->p_data + (y - 1) * w * a;
const uint8_t *p_lower = p_image->p_data + (y + 1) * w * a;
```
// vezme si ukazatele na vrchni a spodni radek

```
 while(p_dest != p_end)
```
// postupuje po pixelech od začátku do konce
```
            p_dest[0] = (p_upper[0] + p_lower[0]) / 2;
            p_dest[1] = (p_upper[1] + p_lower[1]) / 2;
            p_dest[2] = (p_upper[2] + p_lower[2]) / 2;

if(p_image->n_format == fmt_RGBA8)
            p_dest[3] = (p_upper[3] + p_lower[3]) / 2;
```
// vypočítá průměr ze spodního a vrchního řádku, což je vystup

Obr. 32: Výřez načteného prokládaného videa.

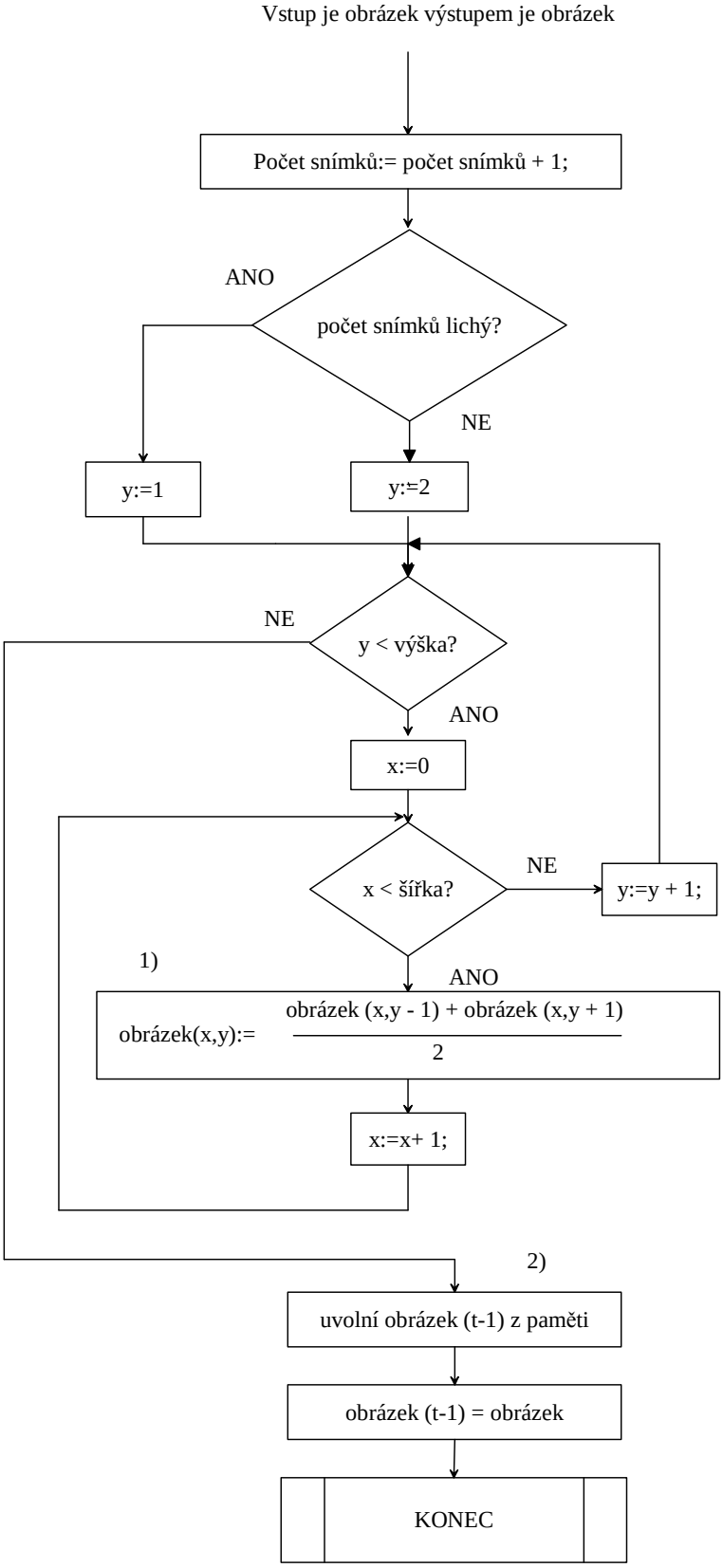
Na obr. 32 lze vidět, že obraz je prokládaný, protože detaily jsou subjektivně vnímané jako čáry (posunutí předchozího půlsnímku rozmazání). Jde o artefakt posunutí v obraze, protože celý snímek je složen z prokládaného videa pořízeny v jednom čase, liší se v nich poloha pohybujících objektů.

Na obr. 33 je vidět výřez obrazu po odstranění prokladu pomocí metody BOB. Jak je vidět obraz jasný, mírně rozmazaný a méně ostrý (rozmazání a ostrost je způsobeno malým rozlišením). Potvrdilo se, že je rozlišení v čase zachováno ale v prostoru je rozlišení zhoršeno (zmíněné rozmazání). V rychlém sledu se "video" subjektivně jeví jako jasné a ostré.

Obr. 33: Výřez neprokládaného videa ve formátu 444.

Pokud se využije vyššího rozlišení (1920,1080 oproti 720,576) je obraz ostřejší a metoda BOB je ohledně subjektivního vyhodnocení, kvalitní. Jak je vidět na obr. 34.

Obr. 34: Výřez neprokládaného videa ve formátu 420 a rozlišení 1920, 1080.



Obr. 35: Vývojový diagram.

Porovnání formátu vzorkování

Tím, že sítnice lidského oka je na různé barevné odstíny méně citlivá a více na jas, můžeme tyto barevné složky podvzorkovat. Podvzorkování vytváří průměr ze dvou sousedních obrazových bodů na řádku popřípadě v sloupci, nebo ze čtyř sousedních pixelů tvořících čtverec a ukládá se jako jediný pixel, tím dochází ke ztrátě části informace o barevné složce obrázku a zároveň k podstatnému zmenšení objemu dat. Pro snížení výpočtové náročnosti se často místo průměrování pouze vynechají některé obrazové body.

Pokud porovnáváme různé formáty vzorkování, jde vidět podle obr. 36, 37 a 38, že rozdíly nejsou pouhým okem viditelné. Pokud si zvětšíme detaily, vidíme, že barvy u a jednotlivé odstíny jsou u nižšího i formátu vzorkování méně kontrastní a někdy dochází k "vloudění" jiného odstínu barvy. Podvzorkování chrominančních barev redukuje množství informací u chrominančních složek obrazu.

Obr. 36: Výřez ve formátu 444.

Obr. 37: Výřez ve formátu 422.

Obr. 38: Výřez ve formátu 420.

Odstranění prokladu - metoda WEAVE

Navržený algoritmus využívá jednoduchého kopírování předchozího řádku. Na začátku algoritmu se počítá kolikátý máme snímek a zkontroluje se zda máme správný formát. Program postupuje po řádcích a to buď po sudých nebo po lichých (opět ovlivňuje liché řádky v lichých snímcích, nebo sudé řádky v sudých snímcích, zatímco řádky s opačnou polaritou kopíruje vždy z předchozího snímku).

Ten algoritmus zachovává rozlišení v prostoru, avšak nesedí v čase. Proto není vhodný pro sekvence, ve kterých se vyskytují pohybující se předměty.

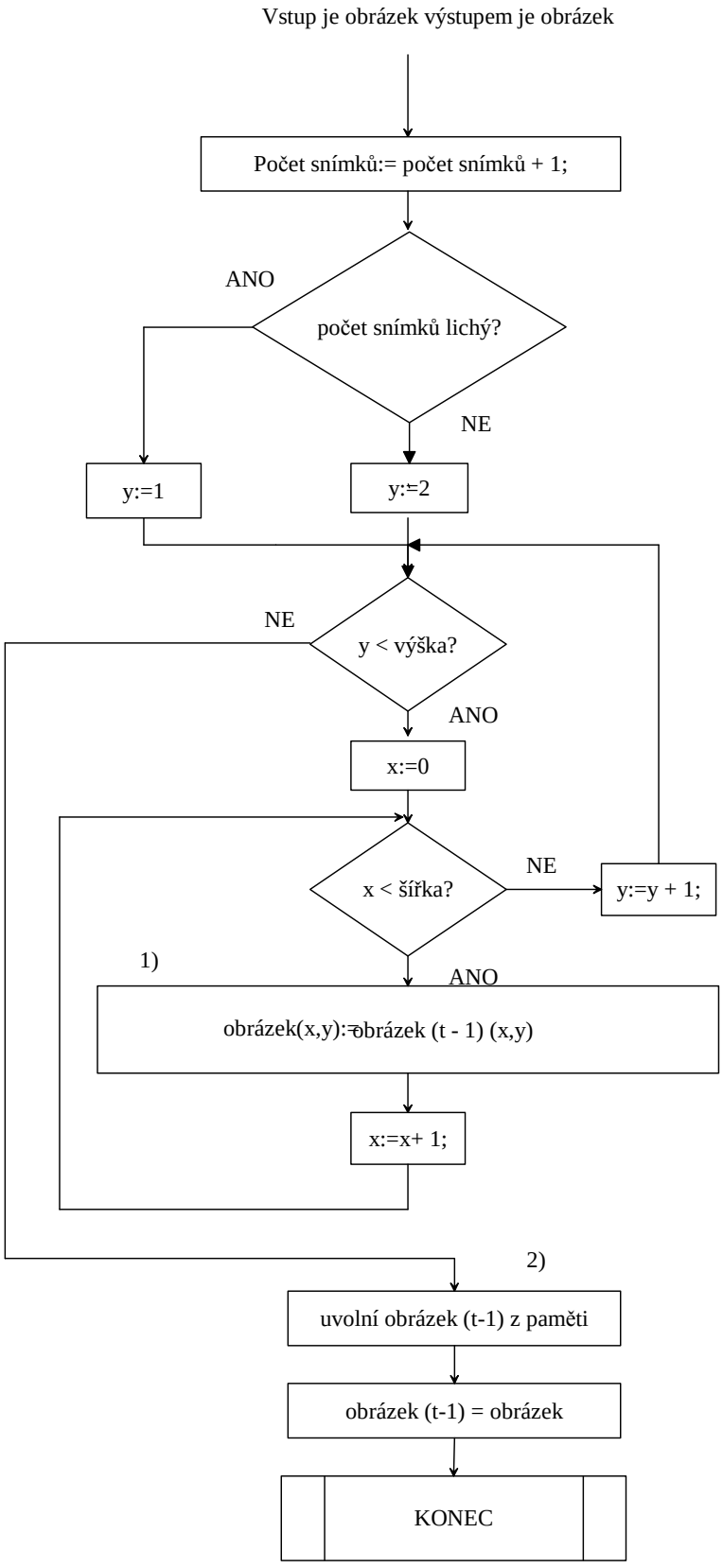
```
const uint8_t *p_src = p_prev_frame->p_data + y * w * a;
```
// spočítá ukazatel na řádek v předchozím snímku

```
memcpy(p_dest, p_src, w * a);
```
// zkopíruje data celého řádku

Obr. 39: Výřez vstupního prokládaného obrazu.

Na obr. 40 je vidět výřez obrazu po odstranění prokladu pomocí metody WEAVE. Jak je vidět obraz ještě více rozmazaný, je to dáno tím, že se kopírují řádky v jiném čase a dávají se pod sebe (sudé řádky zůstávají beze změn a liché jsou kopírovány o čas t-1) a to vytváří rozmazání a efekt tzv. “duchu“ v obraze. Potvrdilo se, že je rozlišení v čase nezachováno.

Obr. 40: Výřez neprokládaného videa.



Obr. 41: Vývojový diagram metody WEAVE.

Odstranění prokladu - metoda VTMF

Adaptivní algoritmus, spojující výhody popsaných metod Weave a bob.

```
const uint8_t *p_upper = p_image->p_data + (y - 1) * w * a;
const uint8_t *p_lower = p_image->p_data + (y + 1) * w * a;
const uint8_t *p_previous = p_prev_frame->p_data + y * w * a;
```
// najde ukazatele na horní a spodní řádek v tomto snímku a na stejný řádek v předchozím snímku

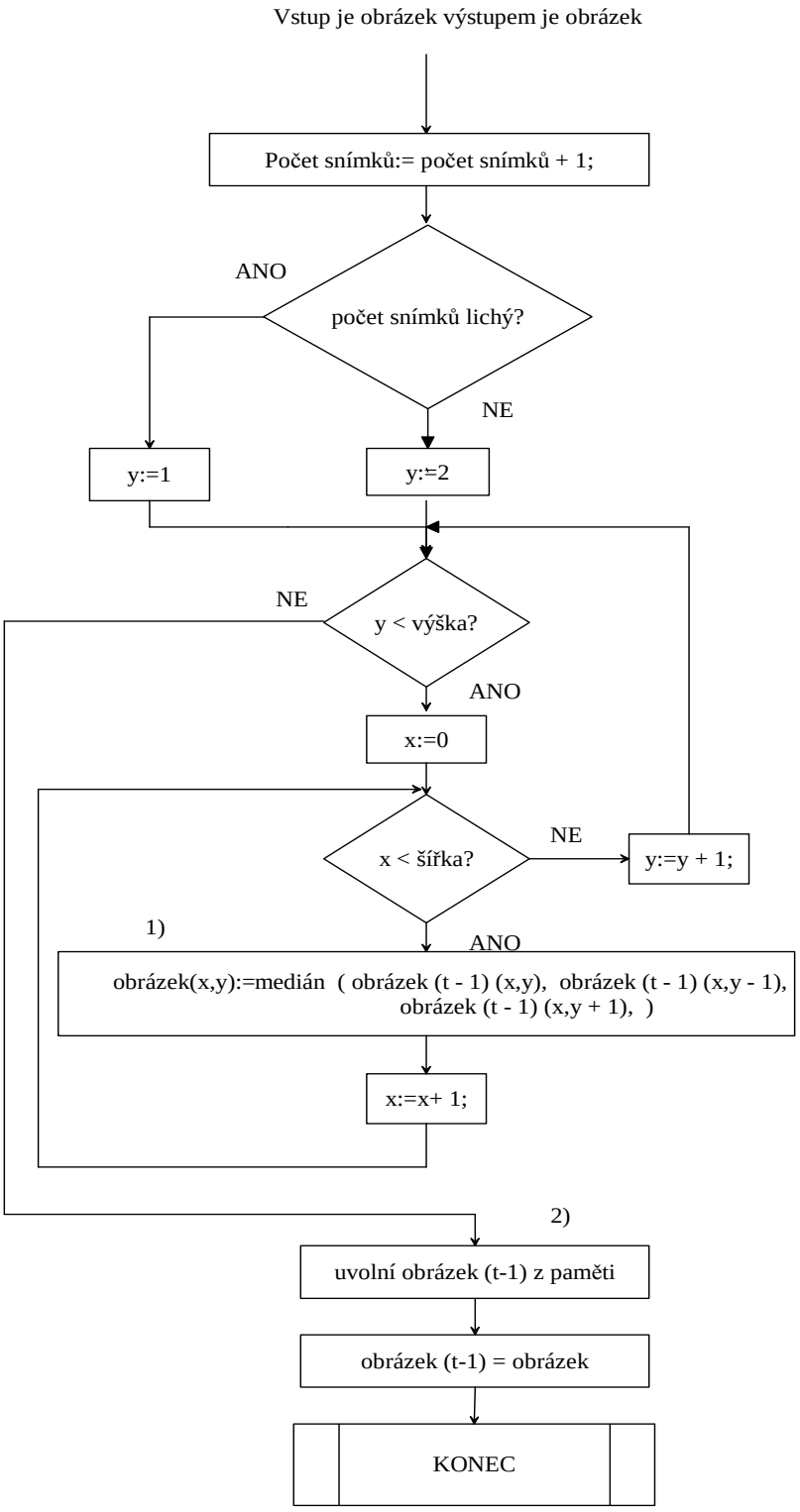
```
while(p_dest != p_end)
```
// projde všechny pixely v řádku
```
 p_dest[0]= n_med3(p_upper[0], p_lower[0], p_previous[0]);
 p_dest[1] = n_med3(p_upper[1], p_lower[1], p_previous[1]);
 p_dest[2] = n_med3(p_upper[2], p_lower[2], p_previous[2]);

      if(p_image->n_format == fmt_RGBA8)
 p_dest[3] = n_med3(p_upper[3], p_lower[3], p_previous[3]);
```

// spočítá medián z vrchního, spodního a předchozího pixelu a to je výsledek
```
            p_upper += a;
            p_lower += a;
            p_previous += a;
            p_dest += a;
```
// posune se na další pixel

Obr. 42: Výřez načteného prokládaného videa.

Obr. 43: Výřez neprokládaného videa.

Na obr. 43 je vidět výřez obrazu po odstranění prokladu pomocí metody WTMF. Jak je vidět je obraz mírně rozostřený ze subjektivního pohledu, mírně rozmazaný a méně ostrý (rozmazání a ostrost je způsobeno malým rozlišením). Tato metody by byla vhodnější na statické obrazy. V rychlém sledu se “video” subjektivně jeví jako méně jasné a ostré.



Obr. 44: Vývojový diagram metody VTMF.

Toto jsou všechno velmi jednoduché algoritmy. Složitější algoritmy využívají faktu, že prokládání je viditelné na pohybujících se objektech a jsou tedy založené na detekci a kompenzaci pohybu. Detekce pohybu pouze určuje, které obrazové body patří pohybujícím se objektům a které ne. Části obrazu bez pohybu jsou poté obvykle zpracovány weave nebo BOB algoritmem, protože dávají pro nepohyblivý obraz nejlepší výsledky. Pro části s pohybem je potřeba zjistit směr pohybu a pro liché, respektive sudé řádky pohyb kompenzovat (kompenzovat časový rozdíl lichých řádků oproti sudým). Program obsahuje algoritmus z této rodiny.

Odstranění prokladu - metoda MC

Na obr. 46 je vidět výřez obrazu po odstranění prokladu pomocí metody MC. Na první 4 snímky se využívá v prvním případě metoda WEAVE. Tím, že se obraz více rozmaže (viz kapitola Odstranění prokladu metodou WEAVE), dochází při výpočtu mírné zkreslení (obraz je mírně rozmazán (jako v metodě WEAVE) v místech kde nedošlo k pohybu (kde byl pohybový vektor malý). Potvrdilo se, že v místech pohybu je obraz ostrý rozlišení v čase zachováno. V rychlém sledu se “video” subjektivně jeví jako jasné a ostré.

Obr. 45: Výřez načteného prokládaného videa.

Obr. 46: Výřez neprokládaného videa.

.

Na obr. 49 je vidět výřez obrazu po odstranění prokladu pomocí metody MC. Na první 4 snímky se využívá metoda BOB. Potvrdilo se, že v místech pohybu je obraz ostrý, rozlišení v čase zachováno. V rychlém sledu se "video" subjektivně jeví jako jasné a ostré. Výsledný obraz se subjektivně jeví stejně jako výsledný obraz zpracovaný v předchozí metodě.

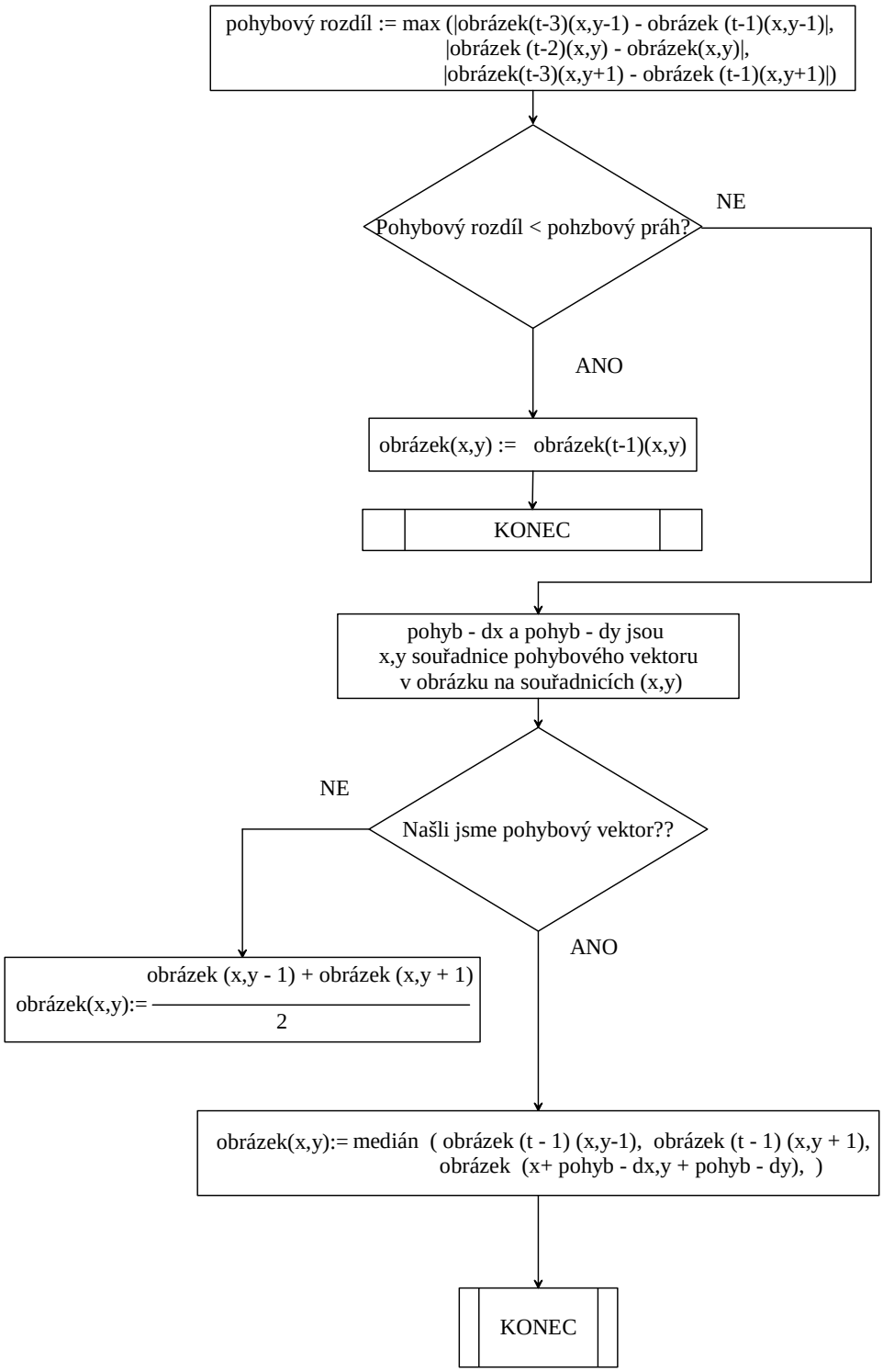
Obr. 47: Výřez načteného prokládaného videa.

Obr. 48: Výřez neprokládaného videa.

Výpočet detekce pohybu v jazyce C++ podle vzorce 9:

```
int n_motion_difference =
n_Max(abs(n_Grey(p_t3_t_pixel)     -     n_Grey(p_t1_t_pixel)),
n_Max(abs(n_Grey(p_t2_pixel)       -       n_Grey(p_t0_pixel)),
abs(n_Grey(p_t3_b_pixel) - n_Grey(p_t1_b_pixel))));
// vypočítá detekci pohybu
```

Program postupuje po řádcích (sudých / lichých) a najde ukazatele na různé obrazové body, posunuté v čase a prostoru a ty použije ve vzorci (9).



Obr. 49: Vývojový diagram programu.

7.4. Vyhodnocení kvality obrazu u jednotlivých metod

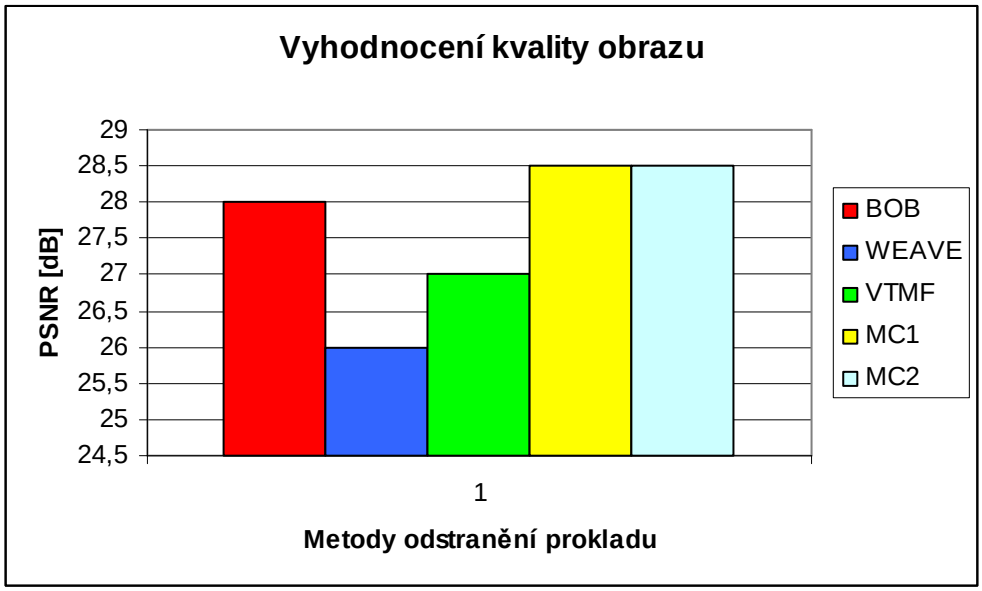
Vyhodnocení kvality obrazu bylo hodnoceno pomocí subjektivního pohledu a pomocí špičkového poměru signál šum PSNR. Nejméně kvalitní metoda se jak z pohledu subjektivního tak i výpočtově se jevila metoda WEAVE, bylo to dáno tím, že jsme ovlivňovali řádky, které nebyly časově korelovány a tím vznikl nepestrý obraz.

Lepší metodou se zdála metoda VTMF, která by byla vhodná na statické obrazy.

Třetí metoda zvaná BOB se z pohledu subjektivního hodnocení zdála efektivní metoda. Obraz v rychlém sledu byl ostrý a jasný, nepatrné rozmazání v místě pohybu nebyly okem postřehnutelné. Tento nedostatek odstraňovala metoda MC, kde byly použity dva pohledy.

První možností bylo, že první 4 snímky byly zpracovány metodou WEAVE (zhoršení obrazu) a pak použita metoda MC. Výsledný obraz je ostrý (jen místy je vidět mírné rozostření, tyto artefakty by šli odstranit filtrací detailů). Další možností bylo využít na první 4 snímky metodu BOB. Výsledný obraz byl naprosto stejný jako při využití metody WEAVE.

Tyto poslední dvě metody byly velmi náročné na čas, protože algoritmus procházel velmi velké množství dat.



Obr. 50: Vyhodnocení kvality obrazu.

7.5. Uložení výsledného snímku

Pojem bitmapa pochází z počítačového terminologie, a znamená to mapa bitů, některých kontextech, termín bitmapa znamená jeden bit na pixel. Bitmapový formát souboru, obvykle s názvem přípony BMP je typ paměti používaný pro ukládání digitálních fotografii.

Nekomprimované bitmapy, obrazové body jsou obvykle uloženy s barevnou hloubku 1, 4, 8, 16, 24, 32, 48 nebo 64 bitů na pixel. Pixel na 8 bitů a méně může představovat buď odstínech šedi nebo indexovaných barev.

Bity zastupující bitmapu obrazových bodů, které můžeme uložit jako byte. V závislosti na barevné hloubce, což je pixel v obrázku bude zabírat nejméně n / 8 bytů, kde n je bitová hloubka. Přibližnou velikost (délka řádků) 2^n barev souboru v bytech lze vypočítat pomocí. [15]

$$\text{rowsize} = 4 \cdot \left\lfloor \frac{(n \cdot \text{width}) + 31}{32} \right\rfloor, \tag{14}$$

Typickým BMP soubor obvykle obsahuje tyto bloky dat:

BMP souboru Header	Obchody obecné informace o BMP souboru.
Informace Bitmap (DIB hlavička)	Obchody podrobné informace o bitmapový obrázek.
Barevná paleta	Obchody definici barev bude použito pro indexované barvy bitmapy.
Bitmap Data	Obchody skutečný obraz, pixel by pixel.

Obr. 51: Bloky dat BMP souboru. [15]

Informace Bitmap
Tento blok bytů popisuje podrobné informace o obrázku, který bude použit k zobrazení obrázku na displeji.

BITMAPCOREHEADER

Tato struktura obsahuje informace o rozměrech a barevné podobě bitmapy (DIB).

```
typedef struct (tagBITMAPCOREHEADER
   DWORD bcSize;
   WORD bcWidth;
   WORD bcHeight;
   WORD bcPlanes;
   WORD bcBitCount;
 BITMAPCOREHEADER);
```

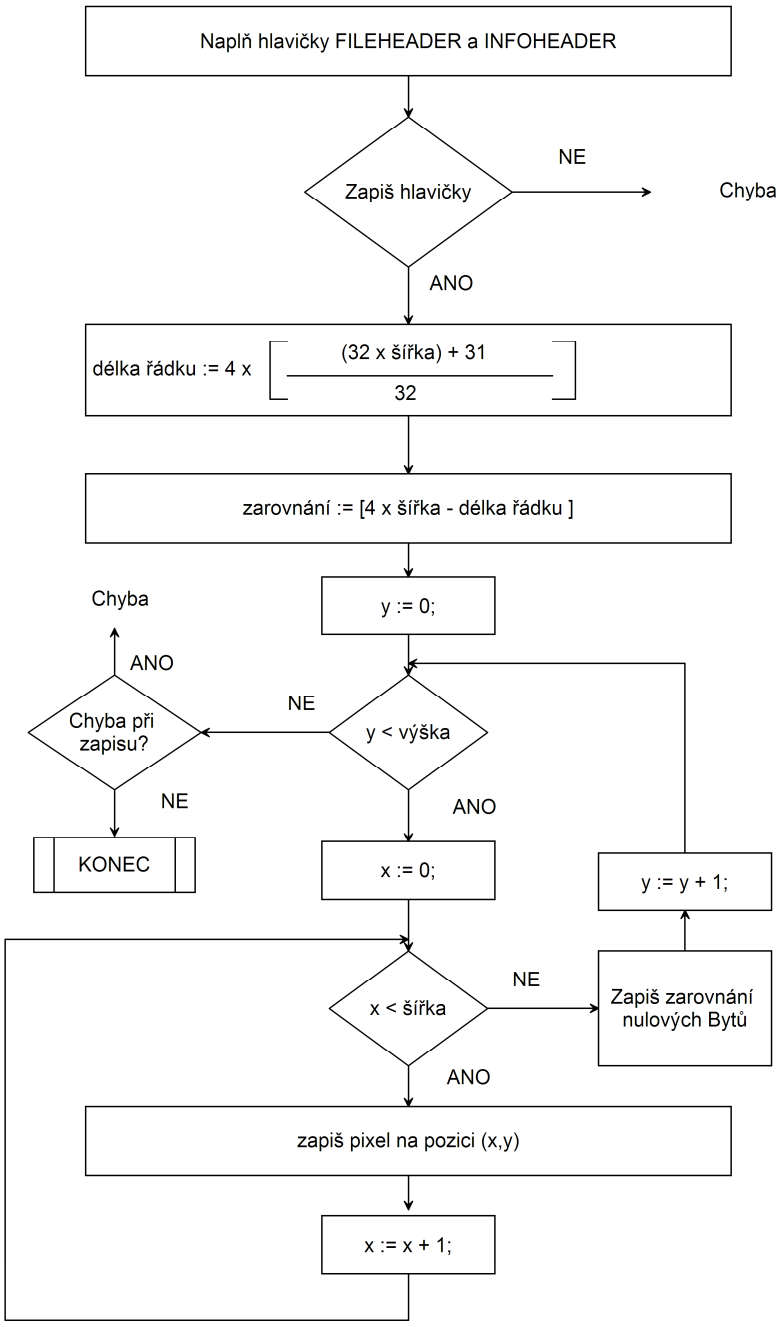
V **BITMAPCOREINFO** konstrukce kombinuje BITMAPCOREHEADER struktury a barvy, tabulky poskytovat kompletní definici DIB rozměrů a barev.
bcSize -Určuje počet bajtů požadované struktury
bcWidth -Určuje šířku bitmapového v pixelech
bcHeight -Určuje výšku bitmapového v pixelech
bcPlanes -Určuje počet letadel pro cílové zařízení. Tato hodnota musí být 1.
bcBitCount -Určuje počet bitů na pixel. Tato hodnota musí být 1, 2, 4, 8 nebo 24.

Zapisání hlavicky souboru

```
BITMAPFILEHEADER t_file_header;
  t_file_header.bfType = 'B' | ('M' << 8);
  t_file_header.bfSize = sizeof(BITMAPFILEHEADER) +
  n_image_size + sizeof(BITMAPINFOHEADER);
  t_file_header.bfReserved1 = 0;
  t_file_header.bfReserved2 = 0;
  t_file_header.bfOffBits = sizeof(BITMAPFILEHEADER) +
  sizeof(BITMAPINFOHEADER);
  fwrite(&t_file_header, sizeof(BITMAPFILEHEADER), 1, p_fw);
```

Zapisání informační hlavičky

```
BITMAPINFOHEADER t_info_header;
        t_info_header.biSize = sizeof(BITMAPINFOHEADER);
        t_info_header.biWidth = p_image->n_width;
        t_info_header.biHeight = p_image->n_height;
        t_info_header.biPlanes = 1;
        t_info_header.biBitCount = 24;
        t_info_header.biCompression = BI_RGB;
        t_info_header.biSizeImage = n_image_size;
        t_info_header.biXPelsPerMeter = 96;
        t_info_header.biYPelsPerMeter = 96;
        t_info_header.biClrUsed = 0;
        t_info_header.biClrImportant = 0;
        fwrite(&t_info_header, sizeof(BITMAPINFOHEADER), 1, p_fw
```



Obr. 52: Vývojový diagram programu bitmap.

8. Závěr

Metody pro odstranění prokladu přináší nové možnosti jak nejefektivněji vytvořit neprokládaný obraz, využívající jak pohybu v obraze tak i metody pro statické obrazy. Výborných výsledků dosahují metody založené na kompenzaci pohybu, jen jsou velmi časově a výpočtově náročné a proto se hledají metody, jak výpočtovou náročnost snížit, proto je potřeba využívat i metody, které nejsou výpočtově ani časově tak složité. Důsledkem bývá omezení parametrů (prostorové rozlišení), které neovlivní výsledný obraz z pohledu diváka.

První část diplomové práce je věnována teorii, která je zaměřena na základy televizní techniky a základy odstranění prokladu.

Další kapitoly se zabývají konverzí mezi prokládaným a progresivním obrazem a naopak. Jedna z jednodušších metod je metoda BOB, která dosahuje dobrých výsledků v čase i v prostoru a nedochází k rozmlžení obrazu. Nejméně kvalitní metoda se jevila metoda WEAVE, bylo to dáno tím, že jsme ovlivňovali řádky, které nebyly časově korelovány a tím vznikl nepestrý obraz. Lepší metodou se zdála metoda VTMF, která by byla vhodná na statické obrazy.

Nejlepší metodou, z pohledu výsledku je metoda MC (kompenzace pohybu). Tuto metodu jsem zkoušel ze dvou pohledů. První byl, že první 4 snímky byly zpracovány metodou WEAVE (zhoršení obrazu) a pak použita metoda MC. Výsledný obraz je ostrý (jen místy je vidět mírné rozostření, tyto artefakty by šly odstranit filtrací detailů). Další možností bylo využít na první 4 snímky metodu BOB. Výsledný obraz byl naprosto stejný jako při využití metody WEAVE.

Tyto poslední dvě metody byly velmi náročné na čas, protože algoritmus procházel velmi velké množství dat. Pro zrychlení výpočtu by se dala využít metoda Motion Estimation (pohybová predikce) je to metoda s časovou predikce video a je založena na odhadu pohybu v následujících snímcích. Základní artefaktem v pohybovém odhadu je, že ve většině případů jsou budoucí snímky videa podobné s předchozími. Tento algoritmus by urychlil náročnost výpočtů a odstranění prokladu by bylo časově méně náročné.

Možné rozšíření navržených algoritmů by bylo jak, urychlit časovou náročnost výpočtů. Možným řešením by bylo využít již zmíněné časové predikce, založené na odhadu pohybu v následujících snímcích, jiné možnosti vyhledávání jako např. logaritmický vyhledávání (TDL - Logarithmic Search), ortogonální vyhledávací algoritmus, (OSA - Orthogonal Search Algorithm), a tak dále. Další možností je využít morfologické operace k zjištění pohybových vektorů a následného výpočtu neprokládaného obrazu (k zjištění pohybu by se dala využít Watershed transformace (povodí)). Nejlepší možností by bylo využít fuzzy logiky, používání přesných popisů a odstranění některých podstatných nevýhod metod MC.

Při výzkumu odstranění prokladu (deinterlacingu) je hlavním tématem sladění poměru mezi účinností a výkonem a proto je zde široký prostor pro diskuzi mezi těmito vlastnostmi a je zde veliký potenciál v jejich vývoji.

SEZNAM OBRÁZKŮ

LITERATURA

[1] MIŠTA, P. : *Hledání objektů v obraze*, Semestrální projekt, FEKT VUT BRNO, 2008

[2] PLÉHA, D. : *Rozpoznávání obličeje*, Bakalářská práce, FEKT VUT BRNO, 2008

[3] LE GALL, D., MPEG: A *Video Compression Standard for Multimedia Applications*, .New York, USA: ACM, 1991. ISSN: 0001-0782

[4] ZÍTKA, M. : *Detekce pohybu v obraze*, Bakalářská práce, FEKT VUT BRNO, 2008

[5] NETCAM. Internetové stránky: [cit. leden 2009], Dostupné na WWW: <http://www.netcam.cz/encyklopedie-ip-zabezpeceni/progresivni-skenovani.php >

[6] Wikipedia. Internetové stránky: [cit. leden 2009], Dostupné na WWW: <http://wikipedia.infostar.cz/y/yu/yuv.html>

[7] TVFREAK. Internetové stránky: [cit. leden 2009], Dostupné na WWW: <http://www.tvfreak.cz/art_doc-08E03CF47AF18760C125727C005926C4.html >

[8] Základy počítačové grafiky, *Barevné modely*, FIT VUT BRNO

[9] Winkler, S. *Digital Video Quality: Vision Models and Metrics*

[10] Internetové stránky: [cit. leden 2009], Dostupné na WWW: <http://www.100fps.com/>

[11] KRATOCHVÍL, T. : *Digitální televizní systémy*, Ústav radioelektroniky FEKT VUT BRNO, 2007

[12] Internetové stránky: [cit. leden 2009], Dostupné na WWW: <http://www.ebu.ch/en/technical/hdtv/test_sequences.php >

[13] VÍT, V. *Televizní technika, přenosové barevné soustavy*. Praha: BEN technická literatura, 1997. 719 stran. ISBN 80-86056-04-X.

[14] Internetové stránky: [cit. leden 2009], Dostupné na WWW: <ftp://vqeg.its.bldrdoc.gov/HDTV/SVT_MultiFormat/>

[15] Internetové stránky: [cit. leden 2009], Dostupné na WWW: <http://en.wikipedia.org/wiki/BMP_file_format>

[16] Koivunen, T., Digest of Technical Papers, *Motion Detection of an Interlaced Video Signal*, 1994, [cit. leden 2009]

[17] XINBO, G., JUXIA, G., JIE, L.: Consumer Electronics, IEEE Transactions on, *De-interlacing Algorithms Based on Motion Compensation*, 2005, [cit. leden 2009]

[18] ZAPLATÍLEK,K., DOŇAR, B.: *Matlab – tvorba uživatelských aplikací*, Praha: BEN – Technická literatura, [cit. leden 2009]

[19] ŘÍČNÝ, V., KRATOCHVÍL. T., *Základy televizní techniky* – skripta, Ústav radioelektroniky, BRNO 2008

[20] ALEXANDRESCU, A. : *Moderní programování v C++ Šablony*, Praha: BEN – Technická literatura, [cit. leden 2009]

[21] ECKEL, B. : *Myslíme v jazyku C++*, Praha: BEN – Technická literatura, [cit. leden 2009]

[22] Internetové stránky: [cit. leden 2009], Dostupné na WWW: <http://www.ldv.ei.tum.de/Members/tobias/videotools/sgi2yuv.zip/view?set_language=en>