

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## DEINTERLACE FILTER

SEMESTRÁLNÍ PROJEKT

SEMESTRAL PROJECT

AUTOR PRÁCE

AUTHOR

BC. TOMÁŠ KUŘINA

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



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

## DEINTERLACE FILTER

DEINTERLACE FILTER

SEMESTRÁLNÍ PROJEKT  
SEMESTRAL PROJECT

AUTOR PRÁCE  
AUTHOR

BC. TOMÁŠ KUŘINA

VEDOUCÍ PRÁCE  
SUPERVISOR

ING. IGOR POTÚČEK, PH.D.

BRNO 2007

# Deinterlace filter

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Igora Potůčka, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Tomáš Kuřina  
3.1.2008

## Poděkování

Rád bych poděkoval za pomoc při implementaci algoritmu a tvorbě tohoto dokumentu Ing. Igoru Potůčkovi, Ph.D.

© Tomáš Kuřina, 2007.

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

## **Abstrakt**

Dokument se věnuje problematice prokládání video obrazu a jeho odstraňování. Je zde popsáno samotné prokládání, jeho historie a důvody pro jeho používání. Dále dokument uvádí důvody pro nutnost odstraňování prokládání a základní metody, které se pro to používají. Shrnuje také dosavadní stav mé implementace základní verze algoritmu pro deinterlacing.

## **Klíčová slova**

video, prokládání, interlacing, deinterlacing, porovnávání bloků

## **Abstract**

This documents elaborates on the subject of video interlacing and deinterlacing. It describes the interlacing of video, its history and reasons that led to its use. The document also explains why is it necessary to remove interlacing a basic methods that are used for that. It also recaps the current state of my basic version of deinterlacing algorithm implementation.

## **Keywords**

video, interlacing, deinterlacing, block matching

# Obsah

Obsah .....	1
1 Úvod.....	2
2 Interlacing .....	3
2.1 Historie .....	3
2.2 Popis .....	4
3 Metody deinterlacingu .....	6
3.1 Weaving .....	6
3.2 Blending .....	7
3.3 Pouze jeden pulsínek.....	7
3.4 Line doubling .....	8
3.5 Adaptivní deinterlacing .....	9
3.6 Kompenzace pohybu .....	9
4 Návrh algoritmu .....	10
4.1 Porovnávání bloků.....	10
4.2 Kompenzace .....	11
5 Současný stav .....	12
5.1 Prostředí .....	12
5.2 Implementace .....	12
5.3 Výsledky.....	13
5.4 Možná vylepšení .....	15
5.4.1 Zvětšování pulsímků .....	15
5.4.2 Využití předchozích snímků .....	15
5.4.3 Jiné prostředí.....	15
5.5 Závěr .....	16
Literatura .....	17

# 1 Úvod

Problematika prokládání video obrazu (tzv. *interlacingu*), kdy se obraz nepřenáší po celých snímcích, ale po půlsnímčích s lichými a sudými řádky, je stejně stará jako samotné televizní vysílání, tedy přes 70 let. Samotné prokládání tehdy ničemu nevadilo a naopak bylo výhodou. Umožnilo přenos televizního obrazu v na svou dobu vysoké kvalitě co se týče plynulosti i rozlišení, při zachování technicky zvládnutelných nároků na přenosovou kapacitu.

V posledních desetiletích ale začíná nabývat na významu odstraňování prokládání (tzv. *deinterlacing*). Souvisí to s rozvojem počítačů, jejichž monitory nedokáží prokládaný obraz zobrazovat. Objevily se sice pokusy o počítačové monitory, které zobrazovaly prokládaný obraz, ale delší práce s nimi byla ergonomicky nepříjemná a vedlo to k zavržení této technologie pro počítačové monitory (a také k zavedení nálepky Non-Interlaced na monitorech pro uklidnění zákazníků). V současné době také zažívají velký rozvoj LCD a plasmové televize (a také samozřejmě LCD monitory pro počítače), pro které platí to samé – nedokáží zobrazovat obraz prokládaně, pouze celé snímky najednou. Je tedy nutnost mít k dispozici rychlé a přitom kvalitní postupy pro převedení prokládaného obrazu na neprokládaný. Základním problémem při takovém převodu je fakt, že jednotlivé půlsnímky jsou každý pořízeny v jiný časový okamžik a případné pohybující se objekty se potom na každém půlsnímku nachází jinde.

Existuje několik základních typů algoritmů, které převod na neprokládaný obraz provádí. Navzájem se od sebe liší rychlostí a s ní související náročností hardwarová implementace, která je nutná hlavně v profesionálních studiích a televizních společnostech. Dále každá z metod poskytuje odlišné výsledky a výběr vhodné metody tak záleží hlavně na oblasti použití, resp. typu daného video obrazu.

Tento dokument popisuje problematiku prokládání, jeho historii a důvody, které vedly k jeho zavedení. Dále se věnuje stávajícím metodám, které se používají pro jeho odstranění. Nakonec jsou zde uvedeny moje zatím dosažené výsledky prováděné v rámci Semestrálního projektu a projektu do předmětu Počítačové vidění.

## 2 Interlacing

Interlacing neboli prokládání je technika pro zlepšení kvality video obrazu při zachování přijatelných nároků na šířku pásma přenosového média.

### 2.1 Historie

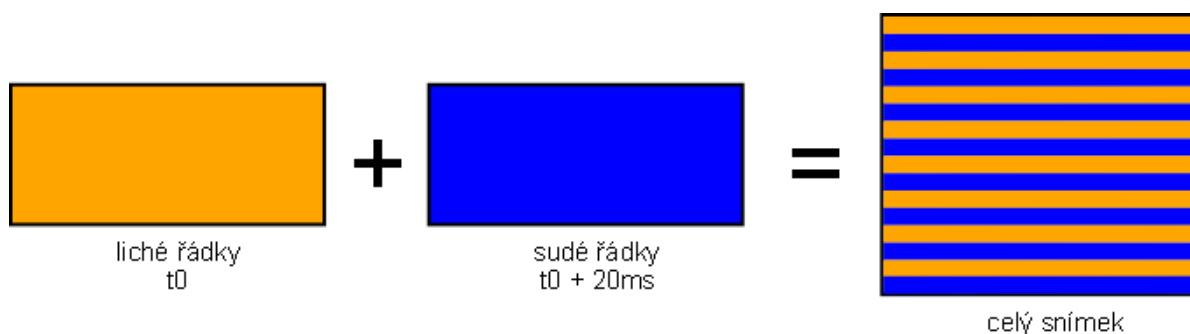
Když se na přelomu 19. a 20. začal rozvíjet film, zjistilo se, že pro plynulé zobrazení bez známek blikání je potřeba promítací plochu osvětlit minimálně 40-krát za vteřinu. Tato minimální nutná hodnota se navíc zvětšuje s přibývajícím světlem v místnosti a rozměry promítací plochy. První filmy se točily rychlostí 16 snímků za sekundu a řešením tedy bylo promítat každý snímek 3-krát za sebou přes trojitou clonu. Výsledkem tak bylo osvětlování promítací plochy s dostačující frekvencí 48 Hz. Později se přešlo na natáčení filmů o 24 snímcích za sekundu (tato hodnota přetrvala dodnes, neboť je pro subjektivní plynulost ve většině případů dostačující) a to umožnilo použití pouze dvojité clony pro dosažení stejné výsledné frekvence osvitů plochy.

Takové řešení ale nebylo použitelné pro televizi, protože by vícenásobné zobrazení jednoho snímku na obrazovce televizoru vyžadovalo uchování celého snímku v nějaké paměti (dnešní *frame buffer*), což bylo v době nástupu televize (30. léta 20. století) technicky nerealizovatelné. Přenášet 40 a více plných snímků zase nebylo možné z důvodu příliš velké náročnosti na šířku přenosového pásma. Posledním limitujícím faktorem byla technologie katodové trubice (CRT) televizí, která kvůli omezení rušení vyžadovala zobrazovací frekvenci shodnou s frekvencí střídavého napětí v elektrické síti, což je 50Hz v Evropě a 60Hz v USA.

Řešením bylo použití prokládání – nepřenáší se celý snímek, ale dva půlsnímky, jeden z lichých a druhý ze sudých řádků. Tyto půlsnímky jsou pak přeneseny za sebou, celkem 50 (nebo 60) půlsnímků za sekundu. První televizní standard definoval 405 řádků, každý půlsnímek tedy obsahoval 202,5 řádků. Tak bylo dosaženo plynulosti i relativně vysokého vertikálního rozlišení. S rozvojem televizní technologie ve 20. století přišly i nové televizní standardy. V USA se začal používat standard NTSC (525 řádků, 60Hz) a ve většině Evropy PAL (625 řádků, 50Hz). Francie přišla s vlatním systémem SECAM, který začalo používat i Rusko a s ním i celý Sovětský svaz včetně ČSSR. Všechny tyto dodnes užívané standardy používají prokládání ze stejných výše uvedených důvodů, jaké vedly k jeho zavedení ve 30. letech.

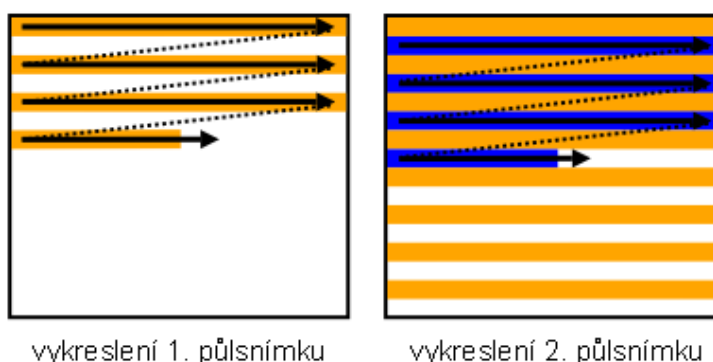
## 2.2 Popis

Při prokládání se nepřenáší celý snímek obrazu, ale dva půlsnímky. Každý půlsnímek má poloviční počet řádků oproti celému snímku a tedy i poloviční výšku. Jeden půlsnímek obsahuje všechny liché řádky původního snímku, zatímco druhý snímek obsahuje všechny sudé řádky. Důležitý je fakt, že každý půlsnímek je kamerou pořízen v jiném časovém okamžiku, s konstantním časovým odstupem. Například v případě standardu PAL jsou od sebe půlsnímky vzdáleny v čase o 20 ms (zobrazovací frekvence PAL je 50 Hz, tedy 50 půlsnímků za 1000 ms):



Obrázek 1 – půlsnímky ve standardu PAL

Na televizní obrazovce nejdřív elektronový paprsek vykreslí na stínítku první půlsnímek, tedy všechny liché řádky. Mezi nimi jsou zatím mezery. Do těch následně vykreslí druhý půlsnímek obsahující zbylé sudé řádky. Fosforové body na stínítku nezhasínají úplně okamžitě, ale trvá jim to nějakou dobu. Tento fakt společně s vlastností lidského zraku, který si každý obraz po krátkou dobu uchovává, umožňuje vnímání takto vykreslovaných půlsnímků jako celků, bez mezer mezi řádky a výraznějšího blikání.



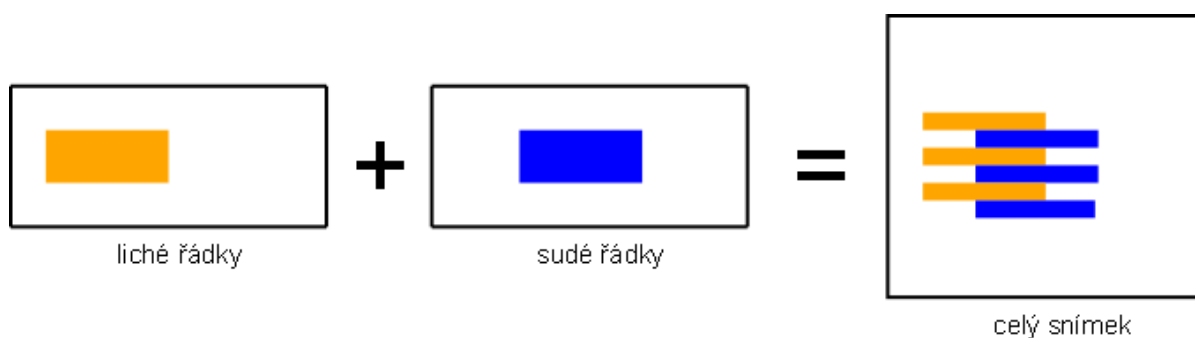
Obrázek 2 - vykreslování půlsnímků na obrazovce

Problém ale vzniká při zobrazení na zařízeních, které zobrazují tzv. progresivní obraz, tedy celé snímky bez prokládání. To jsou typicky počítačové monitory a LCD panely, které v poslední době



začaly výrazně pronikat i do oblasti televizorů v domácnostech a společně s technologií plazmových panelů vytlačují starou technologii CRT. Na žádném z těchto zařízení není možné zobrazit prokládaný obraz, musí zobrazovat celý snímek najednou, a proto je nutné nejdřív provést tzv. *deinterlacing*. Při tomto procesu se půlsnímký převádějí na plné snímky a ty se pak zobrazují.

Tento proces nicméně není dokonalý, a tak může docházet ke ztrátě kvality obrazu, plynulosti nebo ke vzniku artefaktů. Obrázek 3 ukazuje hlavní problém, ke kterému dochází při odstraňování prokládání – tzv. hřebenový artefakt. Ten je způsoben faktem, že jednotlivé půlsnímký nejsou kamerou pořízeny ve stejném časovém okamžiku. Objekt se pak v obraze pohybuje například zleva doprava a na druhém půlsnímký je tedy více vpravo než na prvním. Pokud se tyto půlsnímký zkombinují, budou řádky na obou stranách vysunuté:



**Obrázek 3 - vznik hřebenového artefaktu**

Tento artefakt je hlavním důvodem pro pokročilejší metody odstraňování prokládání. Nepříjemnost tohoto jevu navíc ještě zveličuje vyšší rozlišení obrazovky. V dnešní době dochází k rozmachu HD Ready LCD a plasma televizí, tedy televizí, které dokáží zobrazit minimálně 720 řádků obrazu. Televizní obraz PAL má 576 viditelných řádků a ty se tak musí interpolovat na vyšší rozlišení. Tím ale dochází i ke zvýraznění hřebenového artefaktu na jednotlivých řádcích.

Deinterlacing je tak balancování mezi rychlostí převodu a kvalitou výsledného obrazu. Jednotlivých metodám deinterlacingu se věnuje následující kapitola.

### 3 Metody deinterlacingu

V současné době se pro odstranění prokládání používá několik základních metod, které se liší jak kvalitou obrazu, který produkují, tak rychlostí převodu. Tato kapitola obsahuje jejich popis.

#### 3.1 Weaving

Weaving znamená v angličtině tkaní, a to v podstatě vystihuje princip této metody – jednotlivé řádky z prvního a druhého půlsnímku jsou „spleteny“ dohromady bez jakékoli úpravy. Do finálního snímku se pouze postupně kopírují střídavě liché a sudé řádky. Výhodou takového řešení je samozřejmě rychlost, protože obraz se nemusí nijak zpracovávat, pouze se přesouvají data. Nedochozí také ke ztrátě vertikálního rozlišení, a tak je tato metoda vhodná pro statické scény s minimem pohybu. Pokud se ale v obraze něco rychle pohybuje a pozice objektů se tak v prvním a druhém půlsnímku liší, na plném snímku se to projeví typickým hřebenovým artefaktem:



Obrázek 4 - hřebenový artefakt

Výhody	Nevýhody
<ul style="list-style-type: none"><li>zachovává vertikální rozlišení</li><li>rychlost</li></ul>	<ul style="list-style-type: none"><li>při pohybu v obraze vznik artefaktů</li></ul>

## 3.2 Blending

Této metodě se také říká průměrování. Oba půlsnímký se umístí přes sebe, hodnoty pixelů se zprůměrují a půlsnímký se zvětší na plnou výšku snímku. Tak sice zamezíme vzniku hřebenových artefaktů, dochází ale ke ztrátě vertikálního rozlišení (obraz je rozmazáván) a také vzniká typický efektů duchů (*ghosting*) kolem pohybujících se objektů, jak ukazuje následující obrázek:



Obrázek 5 - efekt duchů

Výhody	Nevýhody
<ul style="list-style-type: none"><li>• zamaskuje artefakty</li><li>• zdánlivě plynulejší</li></ul>	<ul style="list-style-type: none"><li>• snížení vertikálního rozlišení</li><li>• rozmazává i oblasti, které to nepotřebují</li><li>• efekt duchů</li></ul>

## 3.3 Pouze jeden půlsnímek

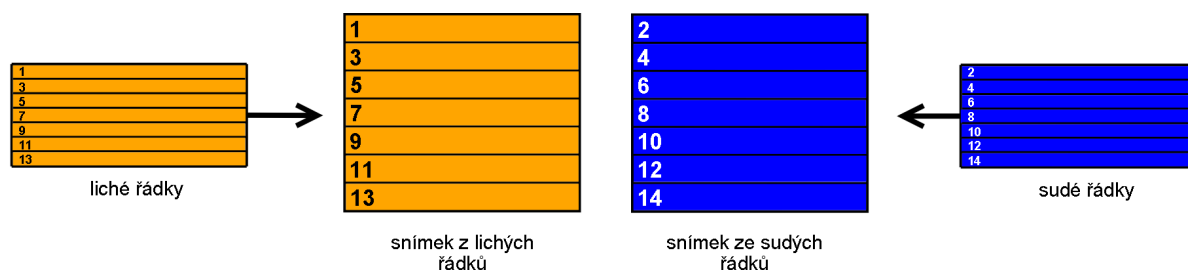
Anglicky taky *discard* (zahazování) nebo *duplicate* (duplikování). Vybere se vždy první nebo druhý půlsnímek, každý jeho řádek se použije dvakrát (zvětší se jeho výška na dvojnásobek) a druhý snímek se zahodí. Obraz tak ale ztratí polovinu svého vertikálního rozlišení. Navíc dochází ke ztrátě plynulosti, protože s polovinou obrazové informace ztrácíme i polovinu časové informace.

Výhody	Nevýhody
<ul style="list-style-type: none"> <li>• rychlost</li> <li>• žádné artefakty</li> </ul>	<ul style="list-style-type: none"> <li>• ztráta plynulosti</li> <li>• snížení vertikálního rozlišení</li> </ul>

### 3.4 Line doubling

Metoda je známá také pod názvem *bob* (poskakovat), kvůli charakteristickému artefaktu, který produkuje. Řádky každého pulsnímků jsou zdvojeny a tím vzniká plný snímek, který je zobrazen. Na rozdíl od předchozí metody se nezahazuje jeden ze dvou pulsnímků. Je tedy přehráváno 50 nebo 60 plných snímků za sekundu a tak je dosaženo původní plynulosti.

Nicméně, liché a sudé řádky si neodpovídají co se vertikální pozice týče. Např. první lichý a první sudý řádek jsou v původním plném snímku samozřejmě pod sebou, při této metodě ale budou ve výsledku zobrazeny každý ve stejné vertikální pozici, totiž na prvním řádku svého plného snímku. Ilustruje to Obrázek 6:



Obrázek 6 - tvorba plných snímků z pulsnímků

Je vidět, že páry řádků 1 a 2, 3 a 4 atd. budou ve svých pulsnímcích na stejných vertikálních pozicích, i když původně byly sudé řádky níž. Obraz tak bude „poskakovat“ nahoru a dolů, od toho tedy anglický název *bob*.

Výhody	Nevýhody
<ul style="list-style-type: none"> <li>• žádné artefakty</li> <li>• plynulost</li> </ul>	<ul style="list-style-type: none"> <li>• „poskakování“ obrazu mezi snímky</li> <li>• snížení vertikálního rozlišení</li> </ul>

### 3.5 Adaptivní deinterlacing

Tato metoda analyzuje související páry půlsnímků a hledá mezi nimi změnu. Tam, kde nedošlo mezi půlsnímky k pohybu, je použita metoda weaving, neboť zachovává maximální rozlišení a k hřebenovému artefaktu vzhledem k absenci pohybu nedochází. V oblastech, kde došlo ke změně a kde by tedy byl vidět hřebenový artefakt se použije metoda blending. Tím se artefakt rozmázne a nebude tak nápadný, nicméně tím opět vzniká efekt duchů. Tato metoda je poměrně rychlá a poskytuje relativně slušné výsledky.

Výhody	Nevýhody
<ul style="list-style-type: none"><li>• lepší výsledky než samotný weaving nebo blending</li><li>• rychlost</li></ul>	<ul style="list-style-type: none"><li>• výsledek závislý na pohybu ve scéně – rychlé scény blending, statické weaving</li></ul>

### 3.6 Kompenzace pohybu

Kompenzace pohybu je nejpokročilejší ze všech metod deinterlacingu. Sleduje pohyb v obraze mezi půlsnímky nebo celými snímky a snaží se tento pohyb ve výsledném plném snímku kompenzovat. To znamená, že pokud je např. ve dvou souvisejících půlsnímcích nadetekován stejný objekt, jen s určitým posunutím, tak algoritmus posune pixely objektu z jednoho půlsnímku do druhého o vzdálenost, kterou mezi půlsnímky objekt urazil. Zachová se tak vertikální rozlišení a zmizí hřebenový artefakt.

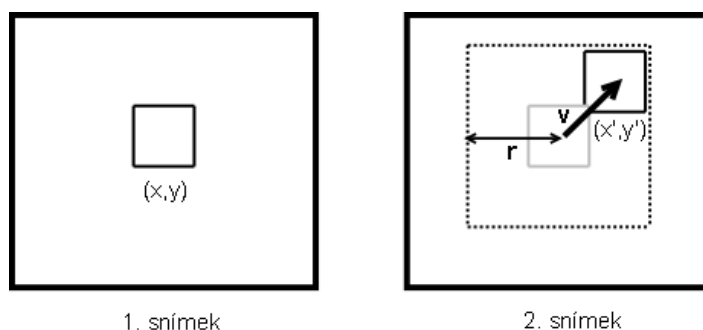
Výhody	Nevýhody
<ul style="list-style-type: none"><li>• nejlepší výsledky ze všech metod</li><li>• zachovává vertikální rozlišení</li></ul>	<ul style="list-style-type: none"><li>• rychlost</li><li>• složitá a drahá HW implementace</li></ul>

## 4 Návrh algoritmu

Algoritmus pro odstranění prokládání bude založen na metodě kompenzace pohybu. Následuje popis hlavních částí implementace:

### 4.1 Porovnávání bloků

Algoritmus porovnávání bloků je jedním ze základních a často používaných algoritmů pro zjištění pohybu mezi dvěma snímky. Princip ilustruje Obrázek 7:



Obrázek 7 - porovnávání bloků

První snímek je rozdělen na bloky zvolené velikosti. Pro každý blok se pak provádí hledání nejlepší shody stejně velkého bloku v druhém snímku. Hledání se provádí do dané vzdálenosti od původní pozice bloku  $r$ . V tomto okolí se pak porovnávají všechny bloky s původním. Po nalezení bloku, který se s ním nejvíce shoduje, vrací algoritmus na výstupu pro tento konkrétní blok prvního snímku vektor pohybu  $v$ .

Zjišťování pohybu mezi půlsnímky nebo celými snímky bude prováděna s využitím metody porovnávání bloků pomocí funkce `cvCalcOpticalFlowBM` volně šiřitelné knihovny Intel OpenCV. Funkce dostává na vstup dva obrázky, několik hodnot pro nastavení a výstupní matice vektorů.

Výhodou takové přístupu je větší přesnost, daná právě použitím bloků. Ostatní funkce pro výpočet optického toku počítají tok po pixelech a výsledky jsou pak poměrně nepřesné, hlavně vzhledem k omezení na šedotónový obraz. U porovnávání bloků je nicméně potřeba najít určitý kompromis mezi přesností a velikostí bloků. Čím větší blok, tím je větší šance, že bude ve druhém obrázku správně nalezen. Na druhou stranu ale větší bloky mohou způsobit viditelné bloky obrazových dat ve výsledném obraze.

## 4.2 Kompenzace

V momentě, kdy dostaneme matici vektorů udávající změny mezi dvěma půlsnímky, můžeme daný pohyb kompenzovat. Vpodstatě to znamená, že si vybereme který půlsnímek bude základní a do něho pak přesunujeme obrazová data z druhého půlsnímku. Ilustruje to Obrázek 4: buď prohlásíme první půlsnímek jako základní a v celém snímku pak posuneme žluté řádky do modrých, nebo vezmeme druhý půlsnímek jako základní, žluté řádky se pak nepohnou a do nich se posunou modré.



Obrázek 8 - ukázka pohybu mezi půlsnímky

Dejme tomu, že si zvolíme jako základní půlsnímek ten s lichými řádky. Podle matice pohybových vektorů potom do mezer mezi lichými řádky přesouváme pixely (resp. jejich celé bloky, shodné s velikostmi bloků použitých ve funkci *cvCalcOpticalFlowBM*) ze sudých řádků. V oblastech, kde nedošlo k žádnému pohybu, budou hodnoty složek vektorů nulové. To znamená, že pixely ze sudých řádků ani nezmění svou pozici ve finálním snímku. Tam, kde došlo k pohybu, získáme pixely z místa, kam se objekt na druhém půlsnímku posunul (tedy posunujeme podle získaného pohybového vektoru daného bloku). Tyto pixely svoji pozici změní.

Nakonec tak dostáváme jeden plný deinterlacovaný snímek vztažený k jednomu z půlsnímků, s daty doplněnými z druhého půlsnímku.

## 5 Současný stav

V rámci projektu do předmětu Počítačové vidění, který jsem si vybral na stejné téma, jsem implementoval základní verzi algoritmu. Následuje jeho popis.

### 5.1 Prostředí

Algoritmus je implementován jako filtr pro populární program pro úpravy videa VirtualDub. Filtry pro tento program se píší v jazyce C/C++ a jedná se v podstatě o DLL soubory, které mají pouze změněnou příponu na VDF. Uživatel pak přes menu Video – Filters – Add pouze vybere tento soubor a filtr se pak použije pro výstup videa.

Pro práci s obrazem je využita knihovna Intel OpenCV.

### 5.2 Implementace

VirtualDub při zpracování videa postupně předává filtru jednotlivé snímky. Ty jsou plné, ale jednoduchým vykopírováním lichých a sudých řádků do snímků poloviční výšky dostáváme potřebné půlsnímky. Tyto půlsnímky jsou převedeny do šedotónové reprezentace a následně předány do funkce *cvCalcOpticalFlowBM*, která pomocí block matchingu vypočítá matice vektorů pohybu mezi půlsnímky. Výsledkem jsou dvě matice, jedna pro osu x a druhá pro osu y. Tyto matice následně procházíme a hledáme očividně špatně vypočítané vektory podle průměru okolních vektorů. Takové vektory si v matici označíme „magickým číslem“, abychom je později mohli vyfiltrovat.

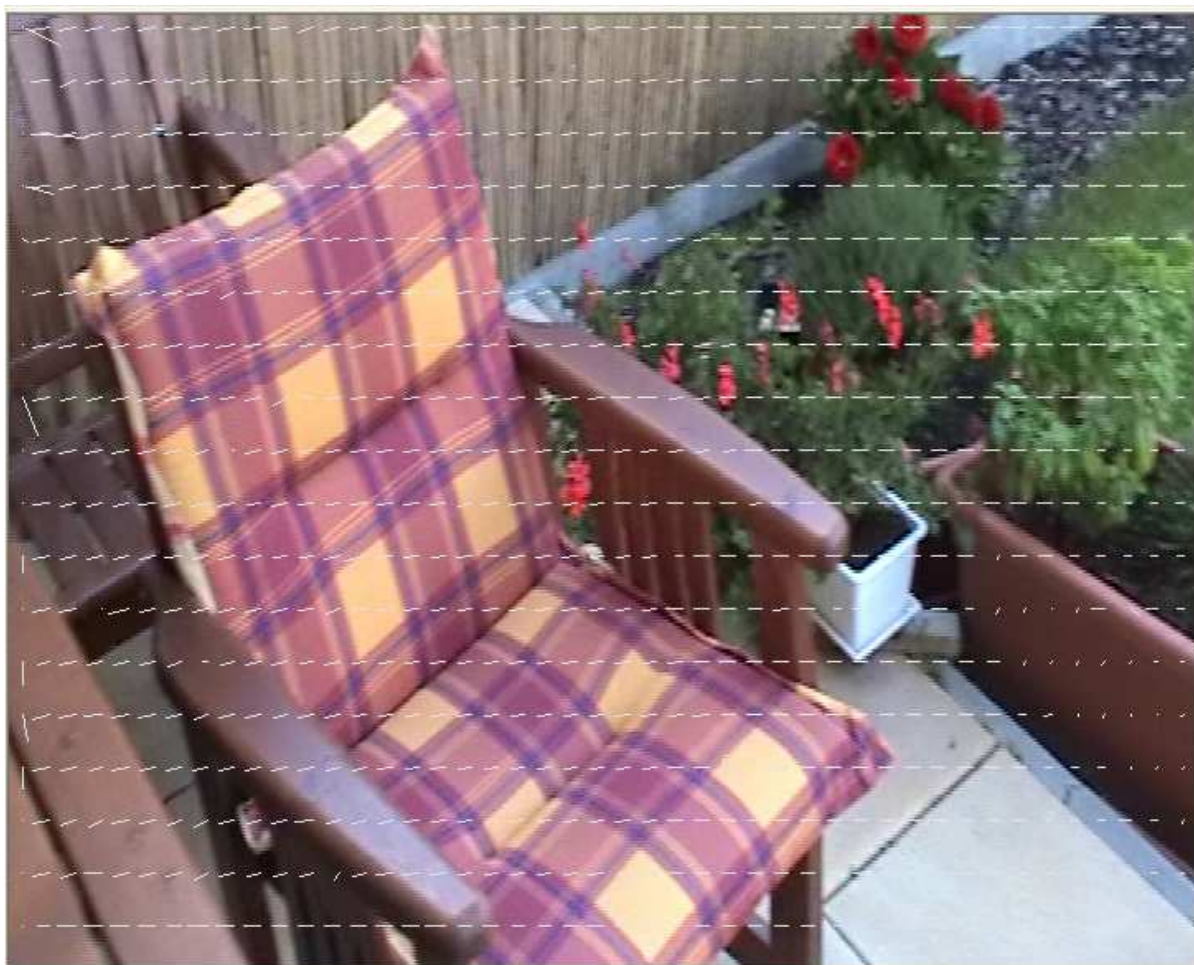
Následně vytvoříme výstupní plný snímek: sudé řádky jsou pouze zkopírovány ze svého půlsnímku na sudé řádky plného snímku. Hodnotu pixelu na lichém řádku zjistíme následovně: vezmeme aktuální pozici pixelu, k ní přičteme příslušnou hodnotu vektoru a z této pozice dostaneme hodnotu pixelu. Tak je vlastně realizován posun lichých řádků do sudých. Posun se neprovede pokud máme vektor označen jako špatný, pak se pouze zkopírují sudé řádky daného bloku (line doubling).

Výsledkem je plný deinterlacovaný snímek, který je uložen na výstup filtru. Vzhledem k tomu, že se jeden půlsnímek zahazuje (resp. se z něho berou data pro druhý půlsnímek), je výsledný počet snímků za sekundu poloviční.



## 5.3 Výsledky

Při analýze výsledků deinterlacování jsem zkoušel porovnávat výstupy dvou různých metod pro výpočet optického toku, *cvCalcOpticalFlowBM* a *cvCalcOpticalFlowHS*. První počítá optický tok porovnáváním bloků a výstupem je matice toku pro bloky v obraze, druhá je založena na algoritmu Horn & Schunck a počítá tok pro každý pixel v obraze. Na následujícím obrázku je zachycena scéna, kde je pohyb kamery zleva doprava a lehce nahoru. Takto vypadá výstup metody porovnávání bloků s vizualizovanými pohybovými vektory:



Obrázek 9 - metoda porovnávání bloků

Jak je vidět, tak naprostá většina vektorů je určena správně. Problém nastává ve dvou oblastech. Vpravo na květináči se vyskytuje spojitá plocha jednoduše barvy a algoritmus pak v takových oblastech selhává. Na druhou stranu, i když jsou vektory v této oblasti špatně, nebude to poznat, protože se budou přesouvat pixely stejné barvy. K dalšímu problému dochází na levém okraji, za který kvůli pohybu kamery zleva doprava mizí scéna. Pak dojde k situaci kdy na jednom

půlsnímku obrazová data z okraje ještě jsou a na druhém už ne, a algoritmus pak opět vybere nejpodobnější blok z dostupných dat podél okraje, což je ale vždycky špatně (správný vektor by musel vést za okraj snímku).

Následující obrázek ukazuje výstup metody HS na stejném snímku videa:



**Obrázek 10 - metoda HS**

Zde je nutno podotknout, že nejsou zobrazeny všechny vektory, protože narozdíl od metody porovnávání bloků počítá tato metoda vektory pro všechny pixely, ne jen pro každý blok. Zobrazen je tedy stejný počet vektorů jako v případě porovnávání bloků.

Na tomto obrázku je vidět, že metoda výpočtu optického toku pomocí algoritmu HS je pro toto použití nevhodná. Vypočtené vektory jsou v podstatě všechny špatně, navíc se mezi jednotlivými snímky výrazně mění. Díky tomu, že ve většině obrázku jsou zde vypočteny vektory blízké nule, tak tento obrázek demonstruje i jak vypadá původní snímek s prokládáním.

## 5.4 Možná vylepšení

Zde jsou navrhovaná vylepšení stávající implementace filtru v rámci samotné DP.

### 5.4.1 Zvětšování půlsnímků

V současné implementaci se porovnávají dva půlsnímky, tedy snímky o poloviční velikosti oproti plnému snímku. Pro dosažení větší přesnosti bych chtěl použít některou z pokročilých metod zvětšování obrazu, např. tzv. *inpainting*. Půlsnímky pak budou mít plnou výšku a více obrazové informace.

### 5.4.2 Využití předchozích snímků

Pro dosažení větší přesnosti a detailu je možné využít i informace obsažené v předchozím snímku. Konkrétně VirtualDub poskytuje filtru i minulý snímek, tedy jsou k dispozici dva předchozí půlsnímky. Ty pak můžeme využít například pro získání obrazové informace, která je na aktuální snímku už za okrajem obrazu, za který se nějaký objekt pohybuje.

### 5.4.3 Jiné prostředí

Nevýhodou VirtualDubu je, že zpracovává snímky v poměru 1:1, tedy jeden snímek na vstup, jeden na výstup. To znamená, že deinterlacované video bude mít typicky 25 snímků za sekundu a pohyb nebude úplně dokonale plynulý tak, jak byl zaznamenán (50 půlsnímků za sekundu). To je možné vyřešit implementací například ve stylu samostatné konzolové aplikace, která bude produkovat z každého snímku dva výsledné plné snímky (jeden se základem v lichých řádcích, druhý v sudých). Tak bude dosaženo původní plynulosti pohybu ve scéně.

## 5.5 Závěr

Problematika odstraňování prokládání nadále zůstává velmi podstatným tématem při zpracování videa, zejména co se televizního materiálu týče. Televize technologie CRT jsou na ústupu a v domácnostech je nahrazují LCD a plasma televize, zobrazující pouze neprokládaný obraz. Současně ale televizní společnosti a studia stále používají techniku prokládání a dokonce ji zavádí i do nových formátů s vysokým rozlišením. Deinterlacing tak bude i nadále velice podstatným faktorem při zpracování obrazu na takovýchto zařízeních, stejně jako na počítačích.

V tomto dokumentu jsem popsal problematiku interlacingu a deinterlacingu, včetně základních, v současné době používaných metod. Také jsem popsal současný stav implementace základní verze deinterlace filtru v rámci projektu do předmětu Počítačové vidění a dosažené výsledky.

Navrhovaná rozšíření v rámci samotné diplomové práce v letním semestru byla popsána v kapitole 5.4. Jedná se o zvětšování půlsnímků na plnou velikost, využití minulých snímků pro získání informace navíc a implementaci v jiném prostředí než je VirtualDub. Tyto modifikace zajistí filtru větší přesnost, tím pádem i kvalitnější výstupní obraz, a také lepší plynulost pohybu ve scénách.

# Literatura

- [1] Deame, J. *Motion Compensated De-Interlacing: The Key to the Digital Video Transition*. 1999
- [2] Gyaourova, A., Kamath, C., Cheung, S.-C. *Block Matching for Object Tracking*, 2003
- [3] Tschumperlé, D., Besserer, B. *High Quality Deinterlacing Using Inpainting and Shutter-Model Directed Temporal Interpolation* , 2004
- [4] Pisarevsky, V *Introduction to OpenCV*, 2007
- [5] Thrun, S. *Optical Flow*, 2006
- [6] Žára, J., Beneš, B., Sochor, J., Felkel, P. *Moderní počítačová grafika*, 2004, ISBN 80-251- 0454-0