

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

DIPLOMOVÁ PRÁCE

Plzeň, 2008

Petr Puncman

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

Použití triangulací pro reprezentaci videa

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 20.5.2008, Petr Puncman,

Rád bych na tomto místě poděkoval svojí matce Věře za podporu během tvorby této práce i po celou dobu studia. Slova díky patří také vedoucí práce Doc. Dr. Ing. Ivaně Kolingerové za důležité konzultace, které vedly k dokončení díla v reálném čase. V neposlední řadě děkuji Tomáši Vomáčkovi za trpělivost s přizpůsobováním jeho knihovny pro moje účely, Ing. Josefu Kohoutovi, Ph.D. za poskytnutí jeho programu včetně doprovodných komentářů a Ing. Petru Lobazovi za motivující nápady ohledně realizace.

Usability of triangulations for video representation

A need for different ways of video representation is significant for both entertainment and research purposes. Each of them has its own benefits and drawbacks in terms of its properties. The main properties comprise usability, compression, performance and extensibility.

The proposed method takes full advantage of triangulations being used as an interpolation basis for resulting video frames. Vertices needed to construct the triangulations in keyframes are obtained by analyzing selected input frames. Transformations of the triangulations in consequent frames are predicted by methods of motion estimation.

The resulting data stream is parametric and thus is widely adaptable for purposes of comparison with classic techniques for video compression. Compression itself together with performance has been verified in a set of experiments and the results are presented in this thesis, too. In this thesis, we have taken a closer look on grayscale video. We have concentrated on video sequences with moving camera and/or moving objects and also on rendered video. As a whole, the proposed method presents a suitable solution for various kinds of video contents. Extensibility of the method is assured by modularity, which up to the present allowed two essential third-party contributions.

Obsah

1 Úvod.....	7
1.1 Základní myšlenka.....	7
1.2 Organizace textu.....	8
2 Digitální video.....	9
2.1 Rozlišení a snímková frekvence.....	9
2.2 Obrazový formát.....	10
2.3 Barevné modely.....	10
2.4 Podvzorkování barev a prokládání.....	11
2.5 Bezeztrátová komprese.....	12
2.5.1 HuffYUV.....	12
2.5.2 Lagarith.....	12
2.6 Ztrátová komprese.....	13
2.6.1 MPEG-1, MPEG-2.....	13
2.6.2 DV.....	15
2.6.3 M-JPEG 2000.....	16
2.6.4 Cinepak.....	16
3 Odhad pohybu.....	18
3.1 Blokové metody.....	18
3.2 Kanade-Lucasova metoda.....	19
3.3 Scale Invariant Feature Transform.....	20
4 Triangulace.....	23
4.1 Delaunayova triangulace.....	24
4.1.1 Pohyblivá DT.....	25
4.1.2 DT s omezením.....	27
4.2 Vykreslení dat z triangulace.....	27
4.2.1 Bilineární interpolace.....	30
4.2.2 Barycentrické souřadnice.....	30
4.2.3 Soustava rovnic.....	31
4.2.4 Warming se třemi páry úseček.....	32
5 Hledání relevantních bodů ve snímku, existující řešení.....	33
5.1 Náhodný výběr a uniformní triangulace.....	34
5.2 Hranová detekce.....	35
5.3 Adaptivní triangulace.....	36
6 Existující řešení na bázi triangulací pro video.....	37
6.1 Adaptivní triangulace a odhad pohybu.....	37
6.2 Tetradizace.....	39
6.3 Triangulace 2D a 3D objektů ve videu.....	40
7 Návrh vlastní metody.....	42
7.1 Kodér.....	42
7.1.1 Intra kódování.....	43
7.1.2 Inter kódování.....	45
7.2 Dekodér.....	48
7.3 Slovní srovnání návrhu s existujícími metodami.....	48
7.4 Bezeztrátová komprese triangulace.....	49
8 Implementace.....	52
8.1 Přístup ke vstupním datům.....	52
8.2 Datové struktury.....	53
8.3 Detaily implementace.....	53

9 Testy a srovnání s dostupnými metodami.....	54
9.1 Systém testování.....	54
9.2 Intra kódování.....	55
9.2.1 Časová složitost.....	56
9.2.2 Datový tok a kompresní poměr.....	58
9.2.3 Výběr optimální hranové detekce.....	58
9.2.4 Srovnání s decimační ADT a s kodekem M-JPEG.....	66
9.3 Intra + inter kódování.....	68
9.3.1 Časová složitost.....	69
9.3.2 Datový tok a kompresní poměr.....	70
9.3.3 Podrobné testy.....	71
9.3.4 Srovnání s kodekem XviD.....	78
10 Závěr.....	79
11 Použité zdroje.....	81
12 Zkratky a pojmy.....	83
Příloha A.....	84
Příloha B.....	89
Příloha C.....	93

1 Úvod

Způsob reprezentace digitálního videa je klíčovým pojmem souvisejícím s kvalitou, kompresním poměrem, ale také s oblastí použití videa. Podstatnou měrou zodpovídá za objem datového toku i za subjektivní dojem, který z videa máme. Oblast použití je poměrně široká. Můžeme se s ním setkat na nosičích SVCD a DVD, na internetu ve streamované podobě, v digitálním televizním vysílání a v neposlední řadě jsou to samostatné anebo vestavěné kamery, které v dnešní době přispívají k nebývalému rozvoji této domény. Zde všude je potřeba video po zakódování uložit (resp. odeslat) v takové podobě, abychom při opětovném čtení (resp. přijímání) udrželi datový tok dostatečně nízký. Po dekodování videa se naše požadavky většinou týkají věrnosti aproximace originálu, tj. zda je vyvážený poměr kvality videa a zmíněného datového toku.

Hlavní odlišnost mezi jednotlivými reprezentacemi spočívá ve volbě metody komprese pro statický snímek. V případě, že je snímek komprimován bezeztrátově, pak je snaha o totéž i v rámci celého videa. Narozdíl od nějaké sofistikovanější metody lze takové video označit za sérii snímků bez využití souvislostí mezi nimi. Většina dnešních metod pro reprezentaci videa však kromě pouhé redukce prostorové redundance redukuje i redundanci časovou. V tom případě se odlišnost metod projeví ve způsobu výpočtu pohybových vektorů a následné kompenzaci pohybu.

Klasická reprezentace obrazové informace prostřednictvím prosté matice pixelů nebo frekvenčního spektra s sebou přináší omezenou možnost aplikace afinních transformací spolu s nejčastěji používanou interpolací při zvětšování obrazu z původní velikosti. V těchto případech může produkovat rušivé artefakty typu deformace obrazu při rotaci nebo obecně nepřesně reprezentované hrany. V souvislosti s videem jsou patrné stejné problémy rozšířené o stíženou možnost transformací nad obrazy z celé videosekvence a interpolací v časové ose, tj. při vkládání dalších snímků mezi stávající.

1.1 Základní myšlenka

Předkládaná diplomová práce, řešená v rámci projektu KJB101470701 - Alternativní reprezentace obrazové informace s využitím triangulací, má za úkol ověřit použitelnost triangulací pro video. Cílem je navrhnout metodu, která s důrazem na přesnost reprezentace, časovou složitost a kompresní poměr bude alternativou ke stávajícím metodám. Tohoto cíle je dosahováno za použití implementace metody, která spadá do kategorie poměrně neprozkoumaných. Jde o metodu ztrátovou, s uživatelsky omezeným datovým tokem. Podstatná odlišnost od většiny dnes používaných metod pak spočívá v tom, že není zapotřebí obraz převádět do frekvenční oblasti a zpět. Přesto je v základu těmto metodám velmi podobná tím, že snímky ve videosekvenci rozdělují na snímky nesoucí maximum vizuální informace, které jsou kódovány absolutně a snímky dopředně predikované, které jsou kódovány relativně vždy vzhledem k předchozímu.

Kompresní schema pro statický snímek (intra kódování) vychází z triangulací nad vybranou

Kapitola 1 | Úvod

podmnožinou bodů obrazu. Takové body určíme buď na hranách anebo je získáme jako důsledek aplikace adaptivní Delaunayovy triangulace. V prvním případě odhalíme hrany v obraze prostřednictvím konvoluce s nejpoužívanějšími hranovými detektory (Laplace, Sobel, Robinson, aj.). Po každé z těchto operací musíme obraz automaticky prahovat, aby hranové body byly jednoznačné. V druhém případě přímo tvoříme optimální triangulaci decimací z kompletní triangulace, která má jako vzorky ve vrcholech všechny pixely obrazu. V této oblasti práce využívá poznatků Ing. Josefa Kohouta, Ph.D. [Koho07]. Zpětná rekonstrukce obrazu ze vzorků probíhá buď interpolací s použitím barycentrických souřadnic, anebo interpolací s použitím obecné soustavy rovnic.

K provázání jednotlivých snímků (inter kódování) využíváme metody pro odhad optického toku [Horn81]. V implementaci jsou využity jednak blokové metody minimalizující metriky MSD, MAD a PDC algoritmem block matching a pak také diferenciální metoda Kanade-Lucas-Thomasi [Luca81]. Poslední zmíněný algoritmus je převzatý z knihovny OpenCV společnosti Intel [OpenCV] a zpřístupněn pomocí wrapperu SharperCV vyvíjeného na jihoafrické Rhodes University [SharCV]. Prostředkem pro uplatnění pohybových vektorů v kompenzaci pohybu konkrétního predikovaného snímku je knihovna pro pohyblivou triangulaci vyvíjená v rámci diplomové práce Tomáše Vomáčky [Voma08]. Vykreslení takovýchto snímků je možno realizovat stejnými prostředky jako u předchozího kódování anebo použít warping založený na posledním vykresleném intra kódovaného snímku.

1.2 Organizace textu

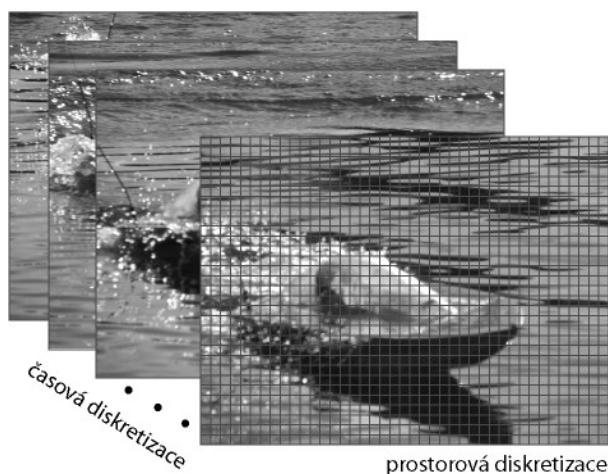
Na následujících stránkách první části práce si představíme digitální video a často používané metody pro jeho reprezentaci. Ozřejmíme si, jakým způsobem lze určit optický tok mezi dvojicí konsekventních snímků a jak lze takové informace využít pro kompenzaci pohybu. Jako dalšího tématického celku se dotkneme Delaunayovy triangulace, její konstrukce a toho, jak se chová při pohybu. Navážeme popisem jejího uplatnění v reprezentacích statického obrazu, potažmo popisem možností tvorby intra kódovaného snímku. Protože kvalita výsledné aproximace závisí nemalou měrou i na interpolaci vzorků z triangulace, uvedeme si i algoritmy nejčastěji používané k tomuto účelu. Následovat budou existující, na triangulacích založené reprezentace videa. Zde poukážeme na velice rozdílný přístup k této oblasti z řad jednotlivých řešitelů jako cesty, kterými je možno se při řešení tohoto problému vydat. To vše by mělo významnou měrou přispět k pochopení celé problematiky a pomoci vytvořit hlubší představu o použitelnosti druhé části práce. Právě druhá část práce popisuje návrh a implementaci metody, která nám bude sloužit jako podklad pro srovnání výsledků ostatními standardně užívanými metodami. Jedná se hlavně o vlastnosti typu kvalita reprezentace, kompresní poměr a časová složitost, které budou porovnávány. Algoritmus sám o sobě (viz kap. 7) dosahuje výsledku za pomoci několika různých variant. Proto předmětem srovnávání bude rovněž najít data, na kterých se ta která varianta chová nejlépe, popř. určit, která z nich je pro většinu testovacích dat optimální. Obrázky, které jsou příliš velké, jsou umístěny v příloze A. V příloze B je uživatelský manuál k testovací aplikaci přiložené na CD. Stručný obsah CD spolu s tipy pro seznámení s aplikací, lze najít v příloze C.

2 Digitální video

Oproti statickému obrazu máme v případě videa jeden rozměr navíc. Narozdíl od reprezentace obrazové funkce $\mathfrak{I}(x, y)$, vracející intenzity pixelů o souřadnicích $[x, y]$, musíme zavést obrazovou funkci $\mathfrak{I}(x, y, t)$, kde t je kladné celé číslo označující pořadové číslo snímku ve videosekvenci. Odpovídající souřadnice pak vyjadřuje uspořádaná trojice $[x, y, t]$, jejímiž prvky jsou nezáporná celá čísla. Pro další práci budeme rozlišovat ještě obraz $\hat{\mathfrak{I}}(x, y, t)$, který představuje zakódovaný obraz v čase t .

2.1 Rozlišení a snímková frekvence

Uvědomíme-li si, že scény z reálného světa jsou spojitě v prostoru i čase, velmi záleží na úrovni diskretizace, kterou zvolíme (viz obr. 2.1). Prostorová diskretizace vzniká při projekci na čidlo digitální kamery, popřípadě při převodu analogového záznamu na digitální. Zveme ji rozlišením. Na druhou stranu časovou diskretizaci představuje snímková frekvence. Jako vedlejší efekt těchto diskretizací sice dochází ke ztrátě většiny obrazové i pohybové informace, ovšem diskretizační kroky jsou v obou případech voleny tak, aby lidský zrak a vnímání zaznamenaly oproti realitě minimální zhoršení kvality.



Obr. 2.1.: Diskretizace scény

V souvislosti s analogovou SDTV, kterou charakterizuje pouze rozlišení vertikální¹, se můžeme setkat s 576 řádky u normy PAL a 480 řádky u normy NTSC. Digitální video oproti tomu poskytuje diskretizaci v obou směrech jako digitální HDTV, která nabízí rozlišení 1280x720 a 1920x1080. Oblast videa na PC pracuje buď s obecným rozlišením anebo s některým ze standardních rozlišení grafických karet (viz tab. 2.1).

VGA	640x480	SXGA	1280x1024
SVGA	800x600	UXGA	1600x1200
XGA	1024x768	QXGA	2048x1365

Tabulka 2.1: standardní rozlišení

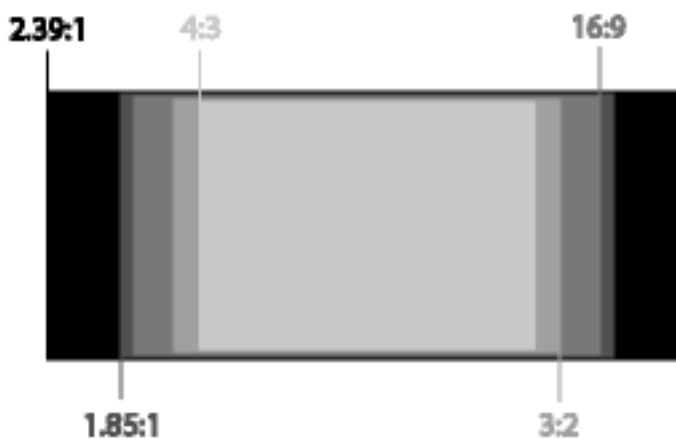
1 V horizontálním směru je signál spojitý

Kapitola 2 | Digitální video

Snímková frekvence se pro navození vjemu pohybu vzhledem k povaze lidského vnímání volí mezi 20fps a 30fps. Pro extrémně pomalé pohyby je možné použít frekvence kolem 10fps. Naopak u extrémně rychlého videa či videa s ostrým obrazem se můžeme setkat i s 50fps. Pro většinu aplikací se postačí inspirovat televizními normami, kdy PAL používá 25fps a NTSC 29.97fps.

2.2 Obrazový formát

Obrazový formát snímků videa má přímou návaznost na poměr stran pixelů, z nichž jsou sestaveny. Analogová televize má typicky obdélníkový pixel s poměrem 12:11 v normě PAL a 10:11 v normě NTSC. Pro potřeby digitálního videa, které vzniká digitalizováním videa podléhajícím jedné z těchto norem, musí být velikost pixelu dodatečně upravována, neboť pixel je v digitálním videu typicky čtvercový. Dnešní nejčastěji používané obrazové formáty jsou 4:3, 3:2, 16:9, 1.85:1, 2.39:1 (viz obr. 2.2).



Obr. 2.2.: Obrazové poměry

2.3 Barevné modely

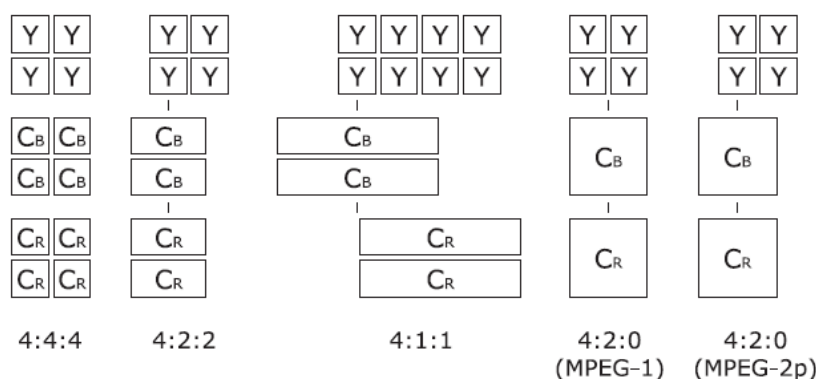
Barevné modely a s nimi související barevná hloubka představují způsob, jak v daném pixelu aproximovat barvu z původní scény. Výchozím modelem pro digitální video je model RGB, kde každá složka obsahuje lineární jas. S tímto modelem počítá valná většina kamerových čidel a všechny ostatní barevné modely lze s ním vzájemně převádět, jak uvádí [Jack05]. Kvůli zpětné kompatibilitě s černobílými televizory se v normě PAL a NTSC uchytil model YUV, kde Y reprezentuje jasovou složku a U,V jsou barevné složky. Tento model lze z RGB po úpravě jasu na R'G'B' aproximovat jako (2.1). Dalším často používaným modelem je $Y C_R C_B$ který představuje rozšíření a posun barevného prostoru YUV. V PC se lze setkat s převodní variantou pro rozsah jedné složky R'G'B' v intervalu 0-255, jak ukazuje následující soustava (2.2).

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \quad (2.1)$$

$$\begin{bmatrix} Y \\ C_B \\ C_R \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (2.2)$$

2.4 Podvzorkování barev a prokládání

Podvzorkování barev je podle [Rich03] často používanou technikou, která má za cíl potlačit informaci o barvě na úkor jasů v určitém poměru. Důvodem je fakt, že lidské vnímání je daleko více citlivé na jasovou informaci oproti informaci o barvě. První dvě čísla trojice označující podvzorkování (viz obr. 2.3) indikují poměr podvzorkování jasové a barevné složky v horizontálním směru. Třetí číslo trojice popisuje vertikální podvzorkování barevné složky. V souvislosti s ním mohou nastat dvě různé situace. Když je toto číslo shodné s druhým číslem trojice, pak je vertikální rozlišení stejné jako originální. V případě, že je toto číslo rovno nule, pak je vertikální rozlišení oproti originálu poloviční. Například u vzorkování 4:4:4 na každé čtyři vzorky Y připadají čtyři vzorky C_R a čtyři vzorky C_B. Vzorkování 4:2:2 má C_R a C_B se stejným vertikálním rozlišením jako Y, ale horizontální rozlišení je poloviční. Výsledné video má pak obecně o třetinu datový tok, přičemž vizuální odlišnosti jsou minimální. Naproti tomu vzorkování 4:2:0 využívané standardem MPEG snižuje datový tok až na polovinu tím, že C_R a C_B mají poloviční horizontální i vertikální rozlišení oproti Y. Vypuštění barevné informace probíhá formou průměrování určitého počtu sousedících pixelů, které mohou být dle příslušných norem pro každou ze složek v horizontálním směru různě posunuty.



Obr. 2.3.: Často používaná podvzorkování barev

Prokládání videa má za úkol rozdělit množinu snímků videosekvence na dva půlsnímky, přičemž všechny řádky každého z nich jsou ve složeném snímku buď sudé nebo liché. Důvodem, proč tato komplikace vešla v praxi, je skutečnost, že se o polovinu snižuje přenosové pásmo vysílače. Každý z půlsnímků je totiž snímán (popř. vyslán) a zobrazován s poloviční snímkovou frekvencí oproti celku. Nevýhody prokládání se projeví při sledování videa jako chvění a blikání obrazu. Při zpracování videa se typicky zhorší podmínky pro rozpoznávání v obraze. Případně, je-li video dále komprimováno, je prokládáním výrazně ztěženo odhad pohybu a musíme nejprve použít odstranění prokládání.

2.5 Bezeztrátová komprese

Digitální video podle způsobu použití je vhodné buď nekomprimovat vůbec, komprimovat bezeztrátově anebo (nejčastěji) ztrátově. Uvážíme-li pak, že hodina nekomprimovaného videa ve formátu PAL má nároky 104,2GB na úložný prostor a 237Mbit/s na datový tok, musíme pro takto velké datové soubory kompresi uvažovat téměř vždy. Bezeztrátová komprese (Huffyuv, Logarith, FFV1, FLI/FLC, aj.) s poměrem průměrně 2:1 až 5:1 představuje právě vzhledem k stále neúnosné velikosti a datovému toku pouze řídkce používanou alternativu k nekomprimovanému videu. V případě, že natáčíme video, které bude podléhat dalším úpravám, a máme zároveň požadavek na zmenšení obsazeného prostoru na médiu, pak je velmi vhodné komprimovat bezeztrátově. Takovouto úvahu lze snadno odůvodnit. Použijeme-li totiž dnes v kamerové technice často dostupný formát M-JPEG nebo MPEG-4, obraz se lidskému oku sice zdá téměř dokonalý. Ovšem po úpravě a opětovné komprimaci takového videa vyjdou artefakty zmíněných metod nepříjemně najevo. Většina často používaných bezeztrátových metod je kódována jako série statických snímků, bez uplatnění predikce. To samozřejmě vede k nižším kompresním poměrům než s predikcí. Zároveň ale důsledkem toho, že každý snímek je intra kódovaný a obsahuje všechnu informaci potřebnou k jeho dekódování, je velice snadný přímý střih a editace.

2.5.1 HuffYUV

[Huff] je bezeztrátový kodek, který vyvíjí pod licencí GPL Ben Rudiak-Gould. Navzdory svému názvu nekomprimuje obraz ve formátu YUV, ale ve formátech YUY2², RGB a RGBA. Jeho hlavními přednostmi jsou rychlost a hardwareová nenáročnost, neboť jeho výkonné části jsou psány přímo v assembleru. Byl navrhován tak, aby na běžném PC prováděl real-time kompresi videa v rozlišení 640x480 při snímkové frekvenci 30fps. Používá volitelně jednu ze tří predikčních funkcí seřazených podle výsledného kompresního poměru. První funkce nazvaná „left“ odhaduje pixel $p_{i,j}$ podle $p_{i-1,j}$. Druhá funkce „gradient“ odhaduje $p_{i,j}$ z $p_{i-1,j}$, $p_{i,j-1}$ a $p_{i-1,j-1}$. Poslední funkce „median“ odhaduje median hodnot z funkcí „left“ a „gradient“. Tato funkce je ale k dispozici pouze po převodu na YUY2, přičemž se k RGB nelze bezeztrátově vrátit. Výsledně se kóduje chyba odhadu pixelu od původního pixelu za pomoci Huffmanova kódování. Kanály jsou kódovány separátně, vyjma formátu RGB, kde se ukázalo být vhodnější kódovat zvlášť rozdíl $R-G$, B , a $B-G$. Celý kodek je založen pouze na intra kódování, tedy každý snímek je kódován zvlášť a predikce snímků není uvažována.

2.5.2 Lagarith

[Laga] je kodek s licencí GPL, jehož autorem je Ben Greenwood. Obecně je při kódování průměrně o 50% pomalejší než HuffYUV (kap. 2.5.1), avšak oproti němu poskytuje o 10-30% lepší kompresní poměr. Kompresní náskok je znatelný hlavně na statických a dobře komprimovatelných scénách. Z důvodu možné ztráty kvality při převodu mezi barevnými

2 Často používaný název pro podvzorkování 4:2:2 normy YC_bC_r .

Kapitola 2 | Digitální video

modely, pracuje nativně s YV12, YUY2, RGB a RGBA. Barva v pixelech je predikována funkcí „median“, která poskytuje nejlepší výsledky v rámci kodeku HuffYUV. Na rozdíl od Huffmanova kódování je však použito RLE a následně aritmetické kódování. Lagarith rovněž podporuje vyřazení identických konsekventních snímků. Každý snímek sekvence je intra kódovaný, s predikcí snímků se počítá v budoucích verzích. Kromě uvedených vlastností je tento kodek zajímavý tím, že byl dodatečně přeprogramován pro 64-bitové procesory.

2.6 Ztrátová komprese

V oblasti ztrátové komprese pro video se nejčastěji uvádějí tři hlavní směry, které se těší komerčnímu využití vzhledem ke kompresnímu poměru 5:1 až 25:1. Jsou jimi kodeky založené na diskretní kosinové transformaci (MPEG1-2, DV, MJPEG, H261-4, aj.), jejichž základem je úprava obrazu ve frekvenční oblasti. Dále jsou to kodeky pracující s diskretní waveletovou transformací (MJPEG 2000, Intel Indeo 5), které v krocích upravují obraz také ve frekvenční oblasti. Poslední používanou skupinou jsou kodeky využívající vektorovou kvantizaci (Cinepak, Sorenson Video) pro seskupování obrazových bodů podobných vlastností. V následujících podkapitolách si z každé ze tří zmíněných kategorií uvedeme jednoho či dva nejvýznamnější zástupce, abychom si na nich vysvětlili odlišné principy kódování. Z tab. 2.2 můžeme získat přehled o tom, jaký je jejich kompresní poměr a čas kódování při standardním nastavení.

metoda	kompresní poměr	čas kódování [s]
VCD; MPEG-1; CBR 1150kb/s	22,22:1	15
DVD-Video; MPEG-2; CBR 7975kb/s	15,58:1	54
Panasonic DV	4,32:1	20
Morgan M-JPEG 2000	16,40:1	17
Radius Cinepak	7,52:1	424

Tabulka 2.2: Test ztrátových kodeků na videu 720x480 (360x240), 29.97fps, 4:2:0, 0:30min

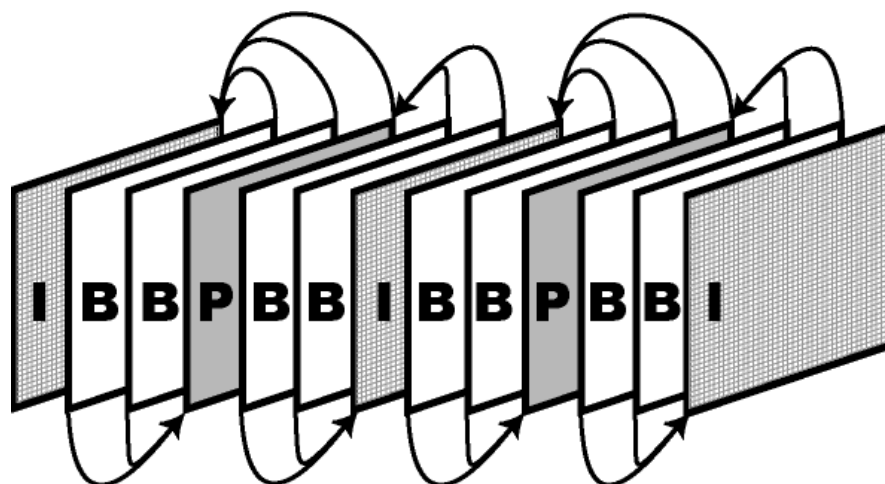
2.6.1 MPEG-1, MPEG-2

Kodek MPEG-1 byl navržen tak, aby dokázal vhodně komprimovat video uložené na VideoCD. Protože rychlost CD mechanik byla v době vzniku kodeku dána datovým tokem na AudioCD, tedy přibližně 150kB/s³, byl pro kodek stanoven datový tok 1,5Mbit pro video a zvuk. Vzhledem ke zvýšeným požadavkům na kvalitu a spolu se snahou dostat celovečerní film na CD byl zvolen inovativní přístup ohledně inter kódování (viz obr. 2.4). Komprese probíhá nad vstupními snímky formátu SIF, které mají rozlišení 352x288 při 25fps (PAL) nebo 352x240 při 30fps (NTSC). Prokládané video není nativně podporováno. Obraz je nejprve rozdělen do bloků 8x8 pixelů a tyto bloky jsou po skupinách 2x2 sloučeny do makrobloků 16x16 pixelů. Podvzorkování barev je podporováno pouze 4:2:0. Makrobloky

3 Přesně 44100[vzorků] x 16[bitů na vzorek] x 2[kanály] = 1,411,200 bitů

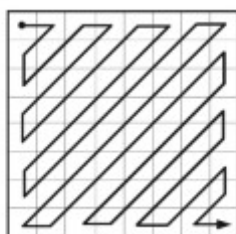
Kapitola 2 | Digitální video

jsou základními jednotkami pro kompresi založenou na kompenzaci pohybu. Snímky jsou rozděleny na tři typy podle způsobu kódování intra (I), dopředného inter kódování (P) a obousměrného inter kódování (B). Jsou sdruženy ve skupinách snímků (GOP), jejichž délka odpovídá počtu snímků od jednoho I-snímku k následujícímu. Snímky I a P představují rovněž tzv. záchytné snímky, od nichž mohou být ostatní snímky odvozovány.



Obr. 2.4: Dvojice sousedních GOP délky 6 (interval mezi záchytnými body je 3); Mezi dvěma intra kódovanými I-snímky se nacházejí dvě dvojice obousměrně inter kódovaných B-snímků a jeden dopředně inter kódovaný P-snímek.

I-snímky jsou kódovány jako samostatné obrazy bez odkazu na předchozí nebo následující snímky. Jejich zakódování probíhá podobně jako v případě statického JPEG. Každý z bloků je nejprve transformován z prostorové do frekvenční oblasti za použití DCT, která jeho obraz rozdělí na frekvenční pásma. Po tomto kroku jsou data kvantizována. Kvantizaci si je možno představit jako ignorování vysokých frekvencí v obraze. Je to spolu s chroma podvzorkováním jediná část intra kódování, kde dochází ke ztrátě obrazové informace. Výsledná data jsou kódována za pomoci RLE po seřazení cik cak (viz obr. 2.5). Tímto způsobem jsou vzorky seřazeny tak, že končí sekvencí nul. Podstata tohoto postupu vychází ze skutečnosti, že po převodu do frekvenční oblasti a následné kvantizaci byla všude kromě levého horního rohu bloku frekvenční pásma eliminována.



Obr. 2.5: Seřazení cik cak

P-snímek je kódován relativně vzhledem k záchytnému snímku, který mu předchází. Každý z makrobloků v konkrétním P-snímku může být kódován buď jako I-makroblok nebo P-makroblok. I-makroblok je kódován stejně jako makroblok v I-snímku. Naproti tomu

Kapitola 2 | Digitální video

P-makroblok je kódován jako vzorová oblast 16x16 pixelů z předchozího záchytného snímku spolu s informací o chybě. K odkázání se na původní makroblok slouží pohybový vektor $(\delta x, \delta y)$. Vektor $(0,0)$ znamená, že vzorový makroblok je na stejné pozici jako ten, který právě kódujeme. V ostatních případech se makroblok pomyslně přesunuje na novou pozici danou vektorem. Takový vektor může obsahovat i půlpixelové hodnoty, které jsou vytvořeny mezi původními pixely jako jejich interpolace. Má to za následek zkvalitnění odhadu pohybu v problematických případech, kdy korespondence není úplně jasná. Zmíněná informace o chybě je vypočtena diferencí posunutého a vzorového makrobloku jako tzv. reziduum. Důvodem je očekávaný vyšší stupeň komprese, neboť oproti posunutému a vzorovému makrobloku je v jejich rozdílu obsaženo podstatně méně informace. Komprese probíhá za postupného aplikování DCT, kvantizace a RLE. Makroblok může být rovněž přeskočen, což nastane při nulovém vektoru a (podle nastavené tolerance) nulové chybě.

B-snímek je kódován relativně k předchozímu a následujícímu záchytnému snímku. Kódování je podobné jako u P-snímku s tou výjimkou, že pohybové vektory mohou ukazovat do následujících snímků. Kontroverzní vlastností tohoto typu snímku je, že obecně menší velikost snímku oproti P-snímku je vykoupena složitějším dekodérem. Dále je předmětem sporů, zda zařazení B-snímku zlepšuje obraz. Hledání optimálního pohybového vektoru ve snímcích P a B je nejvíce inovativní částí kodeku MPEG-1. Konkrétně je použit BM z kapitoly 3.1.

Kodek MPEG-2 je rozšířením MPEG-1. Je reakcí na potřebu kódování videa s prokládáním a videa s vyšším rozlišením. Setkat se s ním můžeme na discích SVCD s rozlišením 480x480 (NTSC) a 480x576 (PAL), DVD-Video s rozlišením 720x480 (NTSC) a 720x576 (PAL) nebo při kódování satelitních přenosů. Typický datový tok se pohybuje od 3 do 30Mbit/s. Vzhledem k složitosti kodéru a dekodéru byly zavedeny tzv. profily, které stanovují úroveň standardu kvůli snadnější implementaci třetími stranami. Oproti MPEG-1 dále nabízí další úrovně barevného podvzorkování, a sice 4:2:2 a 4:4:4.

2.6.2 DV

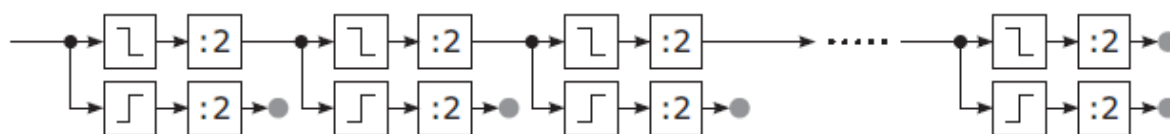
Standard DV obsahuje specifikace hardware (kazety, způsob uložení záznamu) a kodeku pro amatérské video. Právě tímto kodekem a jeho nekomerční implementací se zabývá řada subjektů v čele s firmou Panasonic, komerční variantu pak nabízí např. firma MainConcept. I když tyto kodeky mohou podléhat různým standardům, je zaručena jejich vzájemná kompatibilita. Co se týče komprese, je použito pouze intra kódování, které je speciálně upraveno pro ukládání na DV kazetách. Obraz je nejprve dělen na makrobloky. Následuje chroma podvzorkování 4:1:1 (NTSC) nebo 4:2:0 (PAL). Podle výsledku analýzy pohybu v každém makrobloku jsou převážně statické makrobloky kódovány obdobně jako data ve formátu JPEG pro statický snímek. Přitom každý statický makroblok má kódování každou ze svých složek Y , C_B a C_R odděleně. Celý makroblok tak rozložíme na frekvenční pásma reprezentovaná kosinovými složkami. V případě pohybu uvnitř makrobloku je využito adaptivní komprese založené na pulsničích, přičemž je kódován jejich rozdíl. Zvláštní

Kapitola 2 | Digitální video

případ nastane za předpokladu, že celá scéna je téměř statická. Uspořený podíl na datovém toku je poté dán k dobru vyšší kvalitě a tedy lze vyvodit závěr, že statické scény jsou reprezentovány věrněji než dynamické. Naproti tomu na pohybujících se objektech lze často pozorovat blokové artefakty. Kodek samotný poskytuje lepší výsledky než M-JPEG a je srovnatelný s MPEG-2, který naproti tomu kóduje i mezinímkově. Formát DV25 pracuje s datovým tokem 25Mbit, přičemž se zcela vyjímecně lze setkat s DV50, který má 25/50Mbit. Datový tok však po přidání zvukové stopy, dodatečných dat a opravy chyb může u formátu DV25 vzrůst až na 29-35Mbit.

2.6.3 M-JPEG 2000

M-JPEG 2000 využívá pouze intra kódování založené na metodě JPEG 2000 pro statický obraz. Tato komprese může fungovat dvojím způsobem. Jestliže se při komprimaci používají celá čísla, pak může komprimovat bezztrátově. V případě použití desetinných čísel a následných zaokrouhlování ten samý algoritmus komprimuje ztrátově. Algoritmus použitý při kódování je založen na DWT, jejímž cílem je reprezentace obrazu za pomoci ortonormálních bázevých funkcí. Princip je založen na prioritním vytvoření hrubé reprezentace, ke které jsou doplňovány detaily.



Obr. 2.6: Série filtrů realizující DWT

Nejprve data z řádků a poté data ze sloupců při kódování prochází přes sérii filtrů typu dolní a horní propust (viz obr. 2.6), při dekódování by pak šlo o tzv. rekonstrukční filtry. V obou případech se jedná o konvoluci s odpovídajícím jádrem. Na výstupu horních propustí jsou podrobné koeficienty a na výstupu dolních propustí jsou koeficienty aproximační. Protože je po průchodu každým z filtrů polovina frekvencí odebrána, může být vždy obraz podvzorkován dvěma. Výsledný signál je tvořen z výstupů horních propustí a z poslední dolní propusti, kdy jednotlivá frekvenční pásma tvoří bloky. Jednou podstatnou výhodou (M-)JPEG 2000 je, že v kódovaném souboru jsou k dispozici kvantizační koeficienty seřazené podle důležitosti. Tedy pokud je požadavek na konkrétní velikost snímku v bytech, stačí z každého snímku ponechat pouze toto množství bytů, zbytek smazat a kvalita bude úměrná nové velikosti.

2.6.4 Cinepak

V oblasti kodeků, které jsou založeny na vektorové kvantizaci, je Cinepak tím nejrozšířenějším. Základní myšlenkou vektorové kvantizace je poznatek, že lepší kompresní poměr lze dosáhnout nad shlukem elementů než nad jednotlivými elementy nějaké domény. Podle [Cine01] je každý snímek segmentován na bloky 4x4px a ty jsou popsány jedním nebo

Kapitola 2 | Digitální video

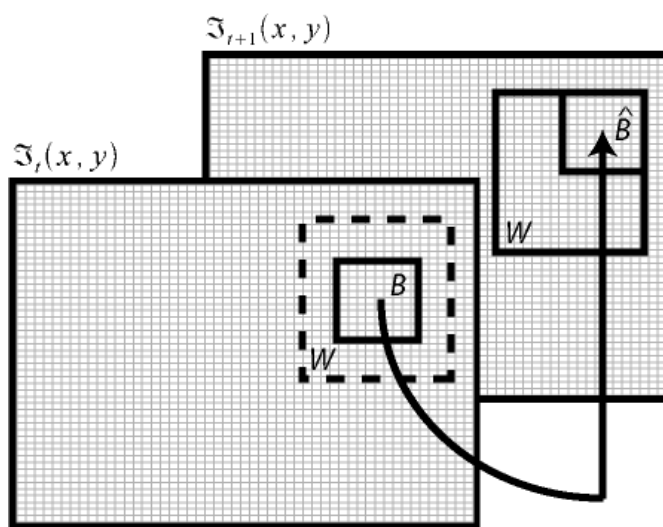
čtyřmi vektory. Při popisu jedním vektorem je použita osmibitová hodnota popisující blok 4x4px za pomoci podvzorkování barev. Popis čtyřmi vektory představuje blok popsáný kombinací čtyř osmibitových hodnot. Pro každý z typů popisů existuje samostatná kódovací tabulka, kde jsou uloženy nejběžnější varianty těchto vektorů. Namísto bloků reprezentovaných vektorem jsou dále kódovány pouze indexy z tabulky. Kvalita videa je potom úměrná velikosti takové tabulky. Pro lepší aproximaci jednotlivých oblastí snímku je plocha snímku dělena na pásy s proměnnou délkou. Každý z pásů obsahuje vlastní unikátní tabulku vektorů. Kódovaný soubor obsahuje jako sekce jednotlivé pásy a v nich kódovací tabulky spolu s indexy vektorů.

3 Odhad pohybu

Pod pojmem odhad pohybu si lze představit výpočet optického toku mezi snímky $\mathfrak{I}(x, y, t)$ a $\mathfrak{I}(x, y, t+1)$, přičemž optický tok je aproximací vektorového pole 2D projekcí rychlostí bodů scény. Změny uvnitř vektorového pole mohou být zapříčiněny buď pohybem kamery anebo pohybem objektů ve scéně. Protože toto vektorové pole nelze z obrazu přímo vyčíst, je nutné se zaměřit na pozorovatelný pohyb vzorů tvořených proměnnou intenzitou jasu v obraze.

3.1 Blokové metody

Blokové metody pro odhad pohybu jsou i přes svoji relativně vysokou algoritmičnou⁴ a časovou⁵ složitost nejpoužívanějším způsobem, kterým je zjišťována souvislost mezi konsekventními snímky. Jejich základním úkolem je zjišťování korespondence vzorového bloku $B(x_b, y_b, k, l)$ ve snímku videa $\mathfrak{I}(x, y, t)$ s potenciálně shodným blokem $\hat{B}(x_b + \delta x_b, y_b + \delta y_b, k, l)$ z okénka $W(x_b, y_b, o, p)$ ve snímku $\mathfrak{I}(x, y, t+1)$. Zde x_b, y_b jsou souřadnice středu hledaného bloku a vyhledávacího okénka. Parametry k, l pak představují rozměry hledaného bloku a o, p jsou rozměry vyhledávacího okénka. Ilustrace celého procesu s naznačením významu výše popisovaných parametru je na obr. 3.1.



Obr. 3.1: Block matching

Vzorové bloky mohou pokrývat obraz beze zbytku v podobě pravidelné mřížky jako v kap. 2.6.1 nebo mohou být centrovány okolo některých obecných bodů triangulace jako v případě metod v kapitole 6.1. V praxi jsou nejčastěji voleny obdélníkové bloky mající $k > l$, protože ve videosekvencích výrazně převládá horizontální pohyb. Volba rozměrů takového bloku není triviální úloha. Příliš malé bloky jsou extrémně citlivé na šum v obraze, zatímco příliš velké bloky produkují méně kvalitní korespondence.

⁴ $O(M^2 N^2)$, kde M^2 je rozměr bloku a N^2 rozměr obrazu

⁵ Podíl na časové složitosti kodéru bývá obvykle 40-50%.

Kapitola 3 | Odhad pohybu

Vyhledávací kritéria

Pro zjištění, zda bloky B a \hat{B} korespondují, bylo vyvinuto množství metrik, které mezi nimi různým způsobem vyhodnocují odlišnosti. Vždy se jedná o per-pixel založené výpočty, čili každé jejich použití na dvojici bloků je nutno kvůli podstatnému zvýšení časové složitosti zvážít.

- $f_{MSD}(\delta x, \delta y) = \frac{1}{kl} \sum_{x=1}^k \sum_{y=1}^l (\hat{B}[x + \delta x, y + \delta y] - B[x, y])^2$ Nejlepší korespondence je určena minimální hodnotou součtu druhých mocnin odchylek odpovídajících pixelů.
- $f_{MAD}(\delta x, \delta y) = \frac{1}{kl} \sum_{x=1}^k \sum_{y=1}^l |\hat{B}[x + \delta x, y + \delta y] - B[x, y]|$ Sčítány jsou absolutní hodnoty rozdílu intenzit odpovídajících pixelů v obou blocích. Nejlepší korespondenci získáme minimalizací této funkce pro jednotlivé páry.
- $f_{PDC}(\delta x, \delta y) = \sum_{x=1}^k \sum_{y=1}^l [f_T(|\hat{B}[x + \delta x, y + \delta y] - B[x, y]| \leq T)]$, kde T představuje práh a funkce $f_T(x)$ vrací 1 při pravdivém vyhodnocení výrazu x a 0 v opačném případě. Vyhovující pixely jsou sčítány, a o nejlepší korespondenci tedy rozhoduje maximální hodnota součtu.

Vektor $(\delta x, \delta y)$, jehož složky jsou parametry optimální funkce, nyní může zastupovat vzorový blok, aniž by tento musel být v následujícím snímku intra kódován.

3.2 Kanade-Lucasova metoda

Algoritmus KLT snižuje podle [Luca81] algoritmickou složitost metod z kap. 3.1 tím, že se nesnaží přímo najít optimální vektor posunutí, ale vytvoří jeho odhad \vec{v} a ten dále zpřesňuje. Dostává se tak až na $O(M^2 \log N)$, kde M^2 je rozměr bloku a N^2 rozměr obrazu. Vlastní výpočet odhadu vektoru je založen na diferenciálním výpočtu a vychází z Horn-Schunkovy metody [Horn81]. Předpokladem je, že jas sledovaného bodu v obraze $\mathfrak{I}(x, y, t)$ zůstává pro všechna vyšetřovaná t neměnný, tj. mění se pouze souřadnice takového bodu. Podle věty o derivaci složené funkce lze odvodit

$$\frac{d\mathfrak{I}}{dt} = \frac{\partial \mathfrak{I}}{\partial x} \frac{dx}{dt} + \frac{\partial \mathfrak{I}}{\partial y} \frac{dy}{dt} + \frac{\partial \mathfrak{I}}{\partial t} = 0 \quad (3.1)$$

Dle [Wiki01] se v rovnici provedou dvě substituce. První představuje dosazení složek rychlosti sledovaného bodu V_x, V_y , druhá parciální derivaci obrazu I_x, I_y, I_t .

$$I_x V_x + I_y V_y = -I_t \quad (3.2)$$

Poslední úpravou je přepsání (3.2) tak, aby v něm figuroval gradient

Kapitola 3 | Odhad pohybu

$$\nabla I \vec{v} = -I_t \quad (3.3)$$

Tato rovnice má dvě neznámé, což znemožňuje přímé řešení. [Luca81] proto předpokládá lokálně konstantní optický tok a vztahuje rovnost danou (3.3) na všechny pixely v okolí daném okénkem $M \times M$, centrovaném ve vyšetřovaném pixelu. Následující soustava je již zohledňuje.

$$\begin{bmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \vdots & \vdots \\ I_{xM} & I_{yM} \end{bmatrix} \begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} -I_{t1} \\ -I_{t2} \\ \vdots \\ -I_{tM} \end{bmatrix} \quad (3.4)$$

Namísto nedourčené soustavy je (3.4) přeúřčená soustava tvaru $A \vec{v} = -b$. Algoritmus KLT vypočítá vektor \vec{v} za pomoci metody nejmenších čtverců jako $\vec{v} = (A^T A)^{-1} A^T (-b)$. Výsledně dostaneme soustavu dvou rovnic o dvou neznámých, kterými jsou složky rychlosti.

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^M I_{x_i}^2 & \sum_{i=1}^M I_{x_i y_i} \\ \sum_{i=1}^M I_{x_i y_i} & \sum_{i=1}^M I_{y_i}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_{i=1}^M I_{x_i t_i} \\ -\sum_{i=1}^M I_{y_i t_i} \end{bmatrix} \quad (3.5)$$

, přičemž každý bod v okénku je přenásoben váhovou funkcí, která upřednostní pixely blíže středu. Ideální je v tomto případě použít gaussian. Po vypočtení odhadu \vec{v} následuje jeho iterativní zpřesňování. To probíhá v měřítkovém prostoru (viz. kap. 3.3) nebo v tzv. pyramidách, v algoritmu popsaném v [Boug99]. V měřítkovém prostoru je nejprve počítána aproximace vektoru posunutí pro dva konsekventní snímky v nejmenších měřítcích. Jeden z nich je následně podroben warpingu podle doposud odhadnutých vektorů. Zlepšení je na základě takto vypočteného obrazu dosaženo iterativně ve větších měřítcích.

Podstatnou výhodou vektorů produkovaných metodou KLT je malá citlivost na šum, nevýhodou pak malá schopnost sledovat body uvnitř téměř homogenních ploch.

3.3 Scale Invariant Feature Transform

Největší pozornost ohledně odhadu pohybu, sledování objektů ve scéně a určování pozice kamery je dnes věnována takovým metodám, které jsou nezávislé na rotaci, posunu, změně měřítka a změně kontrastu a jasu. Jedním ze zástupců takových metod je SIFT [Lowe03], který je v současné době špičkou v aplikacích pro přidávání syntetických objektů do videa, skládání panoramat, rozpoznávání objektů v obraze, aj. Základem pro hledání vhodných bodů pro sledování je sestavení měřítkového prostoru, který velice dobře slouží jako původce význačných bodů v obraze při různých měřítcích.

Výchozím bodem pro sestavení měřítkového prostoru je funkce DoG (3.6), kde x, y

Kapitola 3 | Odhad pohybu

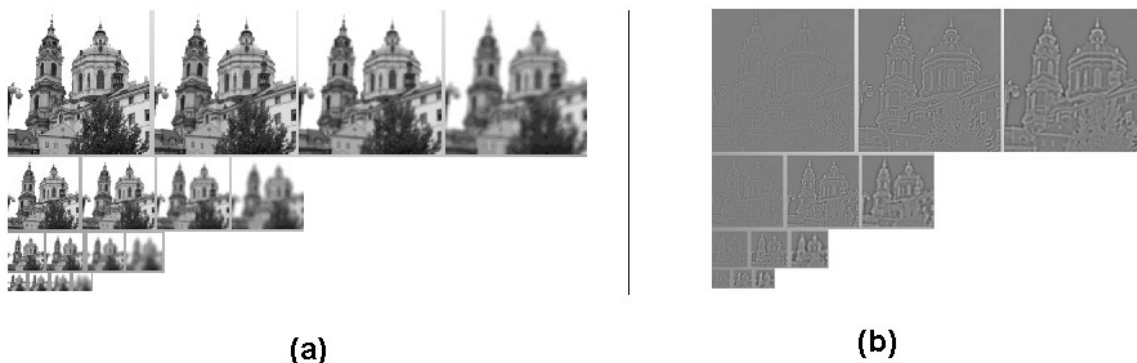
představují souřadnice snímku a σ , $k\sigma$ jsou rozptyly dvou dílčích rozostření. Tento přístup představuje možnou aproximaci Marr-Hildrith algoritmu [Marr80].

$$f(x, y, \sigma) = \frac{1}{2\pi(k\sigma)^2} \exp^{-(x^2+y^2)/(2k\sigma)} - \frac{1}{2\pi\sigma^2} \exp^{-(x^2+y^2)/(2\sigma)} \quad (3.6)$$

Každý obraz v měřítkovém prostoru je pak určen (3.7) konvolucí se vstupním obrazem \mathfrak{I} .

$$D(x, y, \sigma) = f(x, y, \sigma) * \mathfrak{I}(x, y) \quad (3.7)$$

V praxi sestavení vypadá tak, že vezmeme snímek a zmenšíme ho třikrát tak, že získáme další tři nestejně velké snímky. Každý snímek z této čtveřice poté čtyřikrát gaussovsky pokaždé silněji rozostříme a dostaneme celkově snímků šestnáct (viz obr. 3.2a). Stejně veliké snímky od sebe odečteme a získáme tak dvanáct snímků, které tvoří měřítkový prostor (viz. obr. 3.2b).



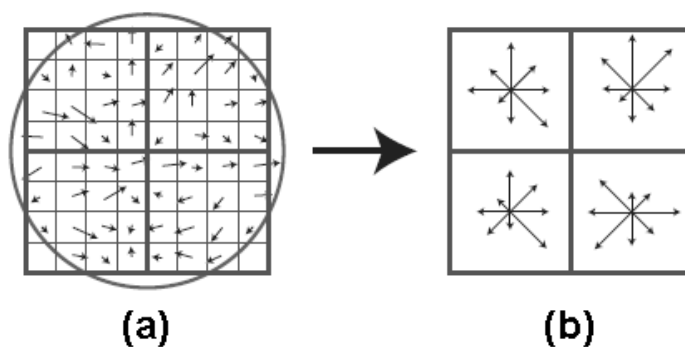
Obr. 3.2: Tvorba měřítkového prostoru. (a) rozostřené zmenšeniny; (b) vzájemný rozdíl (a); převzato z [Mach]

Měřítkový prostor slouží k hledání významných bodů. K tomu jsou vyšetřovány lokální extrémy ve všech snímcích. Pro každý pixel ve stejně velkých snímcích je vyhodnoceno 26 pixelů čítající jeho osmiokolí v aktuálním snímku a odpovídající pixely včetně 8-okolí v předchozím a následujícím snímku. Má-li tento pixel nejmenší nebo největší intenzitu ze všech testovaných, pak je označen za extrém a bude dále zkoumán. Volitelným krokem je proložení lokálního extrému včetně osmiokolí kvadratickou funkcí. Je-li poté vypočten její extrém, vede k výpočtu subpixelových souřadnic významného bodu, což má podstatný vliv na přesnost. Dalším postupem, který významně přispívá ke zpřesnění, je odstranění významných bodů z okolí hran. V místě každého z bodů je vypočten Hessian (3.8), jehož vlastní čísla rozhodují o hlavní křivosti v daném místě obrazu. Body v okolí hran identifikujeme podle toho, že mají velkou hlavní křivost ve směru kolmo na hranu a malou podél ní.

Kapitola 3 | Odhad pohybu

$$H = \begin{bmatrix} \frac{\partial^2 D}{\partial x^2} & \frac{\partial^2 D}{\partial x \partial y} \\ \frac{\partial^2 D}{\partial x \partial y} & \frac{\partial^2 D}{\partial y^2} \end{bmatrix} \quad (3.8)$$

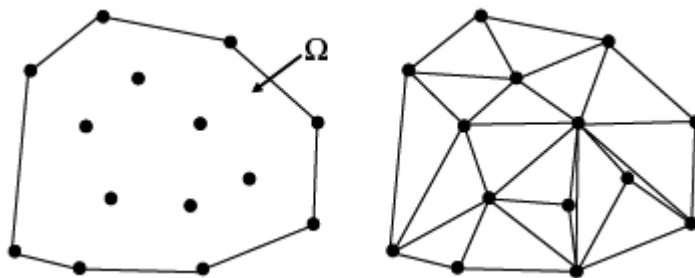
Posledním krokem je vytvoření vektorů označujících zbylé významné body a jejich popsání tzv. deskriptorem. Tyto vektory jsou dány vektorovým součtem prvních derivací ve směru souřadnicových os v místě významného bodu, z něž vycházejí. Deskriptor (viz obr. 3.3) se vypočte vždy relativně k takovému vektoru. Tím je zaručena nezávislost na otočení okolí bodu. Reprezentuje ho 128-složkový vektor představující změny v okolí významného bodu. Při každém mezisním hledání korespondence mezi významnými body se vektory deskriptorů s určitou tolerancí porovnávají podle délky a úhlu.



Obr. 3.3: (a) gradienty v okolí významného bodu, (b) jejich histogram/deskriptor [Lowe03]

4 Triangulace

Triangulace, neboli trojúhelníkové sítě, představují v oblasti počítačové grafiky způsob, jak obecnou oblast v rovině či prostoru rozdělit na simplex. Účelem je vytvoření po částech spojitě plochy, která by vyplnila prostor mezi body, které ji definují. Vzhledem k nutnosti organizace kolekce trojúhelníků v triangulaci je nutné uplatnit určitá omezení. Přesněji, musíme vyžadovat omezení taková, že se v důsledku triangulace Δ stane rozdělením množiny Ω na kolekci přilehlých, neprotínajících se trojúhelníků.



Obr. 4.1.: množina bodů a její triangulace [Hjel06]

Ve většině případů vycházíme při konstrukci triangulace ze zadané kolekce N bodů $P = \{p_i\}, i = 1, \dots, N$ a množiny Ω , která obsahuje všechny body z P . Předpokládáme, že hranicí Ω jsou jeden nebo více uzavřených jednoduchých polygonů. Jednoduchý polygon je přitom takový polygon, který sám sebe neprotíná. Ve většině případů preferujeme, aby Ω byla konvexní obálka dané množiny. Konvexní obálka množiny bodů P je totiž nejmenší konvexní množina obsahující P , přičemž množina S je konvexní, jestliže všechny úsečky spojující libovolné dva body z S leží celé v S . Obrázek 4.1 ukazuje konvexní obálku Ω množiny bodů v rovině a korespondující triangulaci těchto bodů. Hranici Ω budeme zvat $\partial\Omega$. V následujícím textu budeme užívat slovo bod, když se budeme odkazovat na geometrickou pozici vrcholu v rovině. Vrchol⁶ triangulace představuje topologický element, který nutně nemusí jako atribut obsahovat geometrickou pozici. Topologie triangulace záleží na vztazích mezi jejími vrcholy, hranami a trojúhelníky. Nyní asociujeme body p_i, p_j, p_k s vrcholy v_i, v_j, v_k . Každý trojúhelník $t_{i,j,k}$ v triangulaci Δ je tvořen právě třemi vrcholy v_i, v_j, v_k se souřadnicemi $[x_i, y_i]$, $[x_j, y_j]$ a $[x_k, y_k]$. Předpokládáme, že uspořádaná trojice (i, j, k) je taková, že vrcholy jsou v ní seřazeny proti směru hodinových ručiček. V tomto okamžiku rovněž předpokládáme, že např. $e_{i,j}$ a $e_{j,i}$ představují tutěž hranu mezi v_i a v_j .

Pro reprezentaci celé triangulace si dále představíme množinu trojic I představující všechny trojúhelníky v triangulaci, tj. trojice $(i, j, k) \in I$ pro celá čísla i, j, k , označuje trojúhelník $t_{i,j,k}$ v triangulaci Δ . Obecně můžeme nazývat jakoukoliv kolekci trojúhelníků v rovině triangulací, ačkoliv z praktických i teoretických důvodů nás zajímá pouze ta rodina triangulací, která vyhovuje následujícím požadavkům:

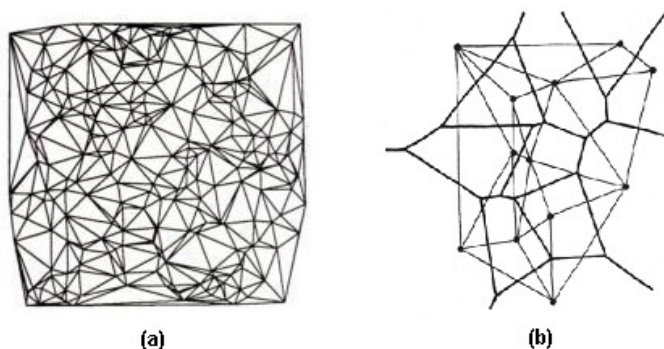
6 V anglosaské literatuře a jiných zdrojích je někdy též označován jako vertex nebo uzel.

Kapitola 4 | Triangulace

- **Absence degenerovaných trojúhelníků:** Žádný trojúhelník $t_{i,j,k}$ v triangulaci Δ není degenerovaný, tj. jestliže $(i,j,k) \in I$, pak p_i, p_j, p_k neleží na téže přímce.
- **Neprotínající se trojúhelníky:** Plochy jakýchkoliv dvou trojúhelníků v Δ se neprotínají, tj. jestliže $(i,j,k) \in I$ a zároveň $(\alpha,\beta,\gamma) \in I$, pak $t_{i,j,k} \cap t_{\alpha,\beta,\gamma} = \emptyset$ s výjimkou případné společné hrany a společných vrcholů.
- **Uspořádanost:** Hranice dvou trojúhelníků mohou mít maximálně jednu společnou hranu a maximálně dva společné vrcholy.
- **Úplnost:** Sjednocení všech trojúhelníků v Δ pokrývá beze zbytku plochu určenou konvexní obálkou, tedy $\Omega = \cup_{(i,j,k) \in I} t_{i,j,k}$.
- **Spojitosť pokrytí:** Triangulace nemá díry.
- **Spojitosť konvexní obálky:** Jestliže je v_i vrchol na hranici $\partial\Omega$, pak musí existovat právě dvě hrany na konvexní obálce, které mají v_i jako společný vrchol. Z toho vyplývá, že počet vrcholů na konvexní obálce je roven počtu hran tamtéž.

4.1 Delaunayova triangulace

Nejvíce používanou triangulací je v současnosti Delaunayova triangulace (viz obr. 4.2a), která nabízí vyrovnaný poměr mezi algoritmickou složitostí⁷ a optimalitou trojúhelníků z hlediska úhlů. Tato triangulace má širší návaznost na Voronoi diagram⁸, který je jejím duálem. Voronoi diagram v podstatě představuje způsob, jakým znázornit oblasti vlivu jednotlivých bodů z množiny P . V E^2 se hrana mezi body p_i a p_j v Delaunayově triangulaci vyskytuje tehdy a jen tehdy, pokud příslušné buňky Voronoi diagramu mají společnou hranu (obr. 4.2b).



Obr. 4.2.: (a) příklad DT [Ding95] (b) konstrukce DT z Voronoi diagramu [Bowy93]

Pro další detaily ohledně Voronoi diagramu, viz [Dey06]. Delaunayova triangulace má jako jedna z mála triangulací velké množství geometrických vlastností, které lze ve většině případů odvodit přímo z vlastností Voronoi diagramu:

⁷ Typicky $O(N)$ až $O(N \log N)$ podle typu konstrukce

⁸ Někdy nazývané též Dirichletovy teselace

Kapitola 4 | Triangulace

- **Opsaná kružnice:** Triangulace je Delaunayova právě tehdy, když opsaná kružnice trojúhelníku této triangulace uvnitř neobsahuje žádný další bod z množiny P .
- **Prázdný kruh:** Dva body p_i a p_j jsou spojeny hranou Delaunayovy triangulace tehdy a jen tehdy, když jimi lze proložit kružnici, jejíž kruh je prázdný. Tato teze je snadno odvoditelná z uvedené vlastnosti opsané kružnice. Obecně prochází-li p_i a p_j kružnice, potom její střed c je bodem na hraně Voronoiova diagramu mezi p_i a p_j . Je tomu tak proto, že bod c je shodně vzdálený od těchto dvou bodů a neexistuje žádný bližší bod.
- **Konvexní obálka:** Okrajové hrany Delaunayovy triangulace tvoří konvexní obálku množiny bodů, nad níž je triangulace zkonstruována.
- **Nejbližší pár:** Buňky nejbližšího páru z množiny P spolu ve VD přímo sousedí.
- **Max-min úhlu a min-max poloměru:** Delaunayova triangulace maximalizuje minimální vnitřní úhel a minimalizuje maximální poloměr opsané kružnice výsledných trojúhelníků.
- **Minimální kostra:** Minimální kostra úplného grafu tvořeného body z množiny P je podmnožinou Delaunayovy triangulace.
- **Unikátnost:** Pro danou množinu bodů existuje pouze jediná Delaunayova triangulace, pokud žádné čtyři body neleží na společné kružnici.

4.1.1 Pohyblivá DT

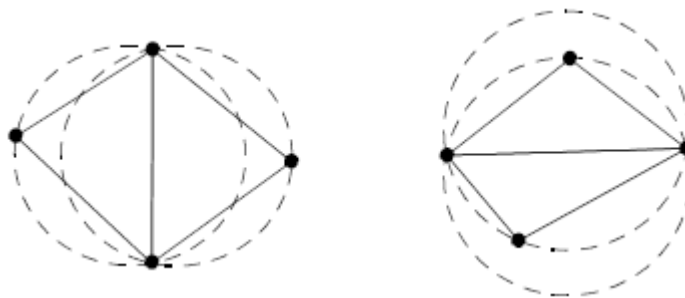
Pro vyšetřování toho, jak se Delaunayova triangulace mění v průběhu času vlivem posunu podmnožiny bodů $q_m \in Q \subset P$ při zadaných pohybových vektorech $v_m = (\delta x_m, \delta y_m)$ pro $m = \{1, \dots, |Q|\}$, navrhl [Voma08] algoritmus, který udržuje konzistenci takové triangulace. Důvodem je vznik topologických změn, které vyplývají z dočasně nevyhovění kritériím uvedených v kap. 4.1. Takovýchto situací může během jednoho časového úseku nastat libovolně mnoho, neboť trajektorie jednoho bodu může být zkřížena až $|Q|-1$ body. Podstatou algoritmu je kontrola podmínky prázdné kružnice (viz obr. 4.3) u pohybujících se bodů $q_m \in Q$ se souřadnicemi $q_i = [x_m, y_m]$ a případná následná legalizace hrany lokálním prohozením.

Zkoumaný trojúhelník $t_{i,j,k}$, do jehož opsané kružnice bod vstoupil, je definován trojicí bodů⁹ (p_i, p_j, p_k) majících souřadnice

$$p_i = [x_i, y_i], \quad p_j = [x_j, y_j], \quad p_k = [x_k, y_k] \quad (4.1)$$

9 uspořádaných v trojici proti směru hodinových ručiček

Kapitola 4 | Triangulace



Obr. 4.3.: Příklad Delaunayova kritéria opsané kružnice.
Vlevo je splněno, vpravo nikoliv. [Burg00]

Na základě takto definovaných čtyř bodů lze sestavit tzv. in-circle test (test kružnice opsané), který představuje matice (4.2), pro jejíž determinant platí $\det(I)=0$. Těto soustavě rovnic vyhovují všechny body ležící na opsané kružnici.

$$I = \begin{bmatrix} x_i & y_i & x_i^2 + y_i^2 & 1 \\ x_j & y_j & x_j^2 + y_j^2 & 1 \\ x_k & y_k & x_k^2 + y_k^2 & 1 \\ x_m & y_m & x_m^2 + y_m^2 & 1 \end{bmatrix} \quad (4.2)$$

Mohlo by se zdát, že cílem testu je zjistit, zda $q_m = [x_m, y_m]$ leží uvnitř kružnice, tedy zda determinant je větší než nula. Z důvodu praktické realizace této metody, která je orientovaná událostně, však zjišťujeme přímo výskyt čtyř bodů na jedné kružnici. K tomuto účelu je dále vyžadována parametrizace poloh bodů prostřednictvím časové proměnné. Matice I se poté změní následovně

$$I = \begin{bmatrix} x_i & y_{pi} & x_i^2 + y_i^2 & 1 \\ x_j(t) & y_j(t) & x_j(t)^2 + y_j(t)^2 & 1 \\ x_k(t) & y_k(t) & x_k(t)^2 + y_k(t)^2 & 1 \\ x_m(t) & y_m(t) & x_m(t)^2 + y_m(t)^2 & 1 \end{bmatrix} \quad (4.3)$$

Za předpokladu, že se zkoumané body pohybují pohybem rovnoměrným přímočarým, musíme hledat kořeny polynomu 4. stupně. Nyní si parametricky vyjádříme přímku reprezentující trajektorii každého bodu, která např. pro bod q_m vypadá takto:

$$\begin{aligned} x_m(t) &= x_m(0) + t \cdot \delta x_m \\ y_m(t) &= y_m(0) + t \cdot \delta y_m \end{aligned} \quad (4.4)$$

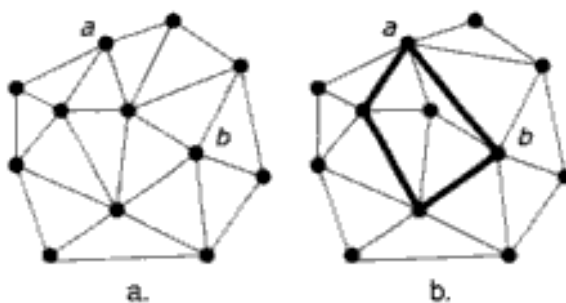
, kde $[x_m(0), y_m(0)]$ představuje výchozí bod trajektorie, δx_m a δy_m jsou rychlosti bodu v příslušných osách a t je parametr představující čas. Po dosazení za všechny body podle (4.4) do (4.3) pak samotné kořeny představující hodnoty časové proměnné t , kdy triangulace

Kapitola 4 | Triangulace

přestává být konzistentní, vypočteme metodou založenou na Sturmových posloupnostech polynomů, podrobněji viz [Voma08].

4.1.2 DT s omezením

V případech, kdy je potřeba přesně reprezentovat hrany a zachovat určitý detail uvnitř celku, který je jinak reprezentován hrubě, nemusí být nasazení prosté DT postačující. Z důvodů, kdy potřebujeme triangulaci například k reprezentaci písma, můžeme jeho libovolné hrany triangulaci „vnutit“ a triangulace je pak bude obsahovat i v případě, že se geometrie obrazu v okolí změní. Další aplikací je triangulace nekonvexního tvaru při zachování vnější nekonvexní obálky, viz [Fuks06]. Tyto možnosti poskytuje CDT (Constrained Delaunay triangulation), která je definována jako taková triangulace Δ nad množinou Ω , vrcholy jejíchž nevynucených hran jsou navzájem viditelné. Bod je při tom viditelný z jiného bodu, pokud úsečka spojující tyto dva body neprotíná vnucenou hranu. S možností vnucení hran se mírně modifikuje definice DT, aby tyto hrany zůstaly zachovány i při případných dalších úpravách v DT. Na obr. 4.4 můžeme vidět vnucení hran.

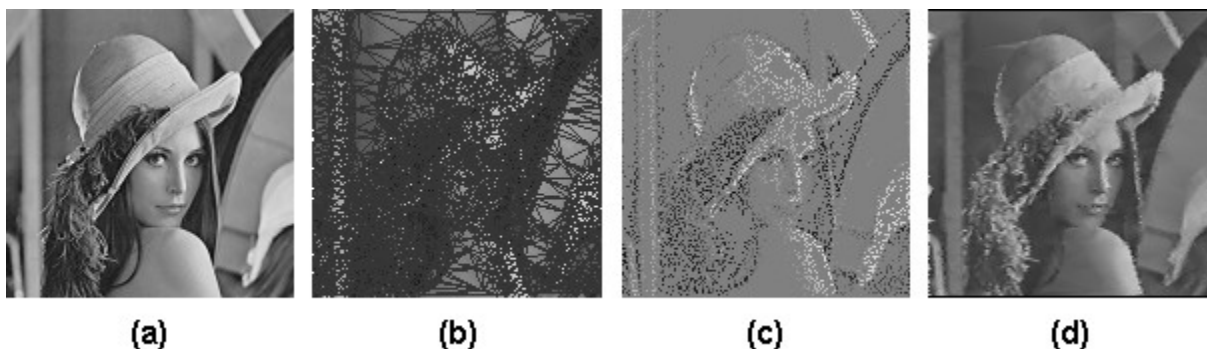


Obr. 4.4: V triangulaci (a) bylo aplikováno omezení (b) [Kell04]

4.2 Vykreslení dat z triangulace

V případě, že s každým vrcholem Delaunayovy triangulace máme asociovaný vzorek intenzity z daného pixelu v původním obrazu, pak potřebujeme metodu, která dopočte intenzitu pro všechny ostatní pixely ležící v ploše a na hranách mezi takovými vrcholy. Účelem je získání rekonstrukce původního obrazu z dostupných vzorků při dekompresi statických obrazů nebo videa jako na obr. 4.5. Interpolace může významnou měrou ovlivnit kvalitu výstupu, proto se lze setkat s vícero přístupy, které tento problém řeší. Jmenovitě se jedná například o bilineární interpolaci anebo interpolaci za pomoci barycentrických souřadnic. Předmětem výzkumu jsou ale i experimentální metody interpolací prostřednictvím Coonsových a Bézierových plátů, více v [Jana07].

Kapitola 4 | Triangulace



Obr. 4.5: (a) vstupní snímek, (b) DT nad (a), (c) vrcholy DT, (d) interpolovaný snímek

Obecně definovaná interpolace předpokládá spojitou plochu definovanou body trojúhelníka $t_{i,j,k}$ s body p_i, p_j, p_k a souřadnicemi $[x_i, y_i]$, $[x_j, y_j]$ a $[x_k, y_k]$ (viz obr. 4.6a). My však potřebujeme tento aparát použit v diskrétní podobě, tedy pro výpočet intenzity pixelů v rámci pravidelné mřížky. Prostředek, s pomocí kterého docílíme takovou interpolaci, představuje Bresenhamův algoritmus [Bres65].

Nejprve se zaměříme na získání rastrových reprezentací hran $e_1=(p_i, p_j)$, $e_2=(p_j, p_k)$ a $e_3=(p_k, p_i)$ (viz obr. 4.6b). Vybereme-li z této trojice hranu¹⁰, jejíž kolmý průmět do osy y má stejný rozměr jako kolmý průmět celého trojúhelníka do osy y , pak tuto hranu označíme e_{ymax} . Zbylé dvě hrany dále přejmenujeme na e_{ya} a e_{yb} . Body náležící těmto hranám budeme sdružovat v množinách E_{ymax} , E_{ya} a E_{yb} seříděné podle souřadnice y . Pro přímky, na nichž hrany leží, platí následující vztahy:

$$\begin{aligned} e_1 &\equiv y = k_1 x + q_1 \\ e_2 &\equiv y = k_2 x + q_2 \\ e_3 &\equiv y = k_3 x + q_3 \end{aligned} \tag{4.5}$$

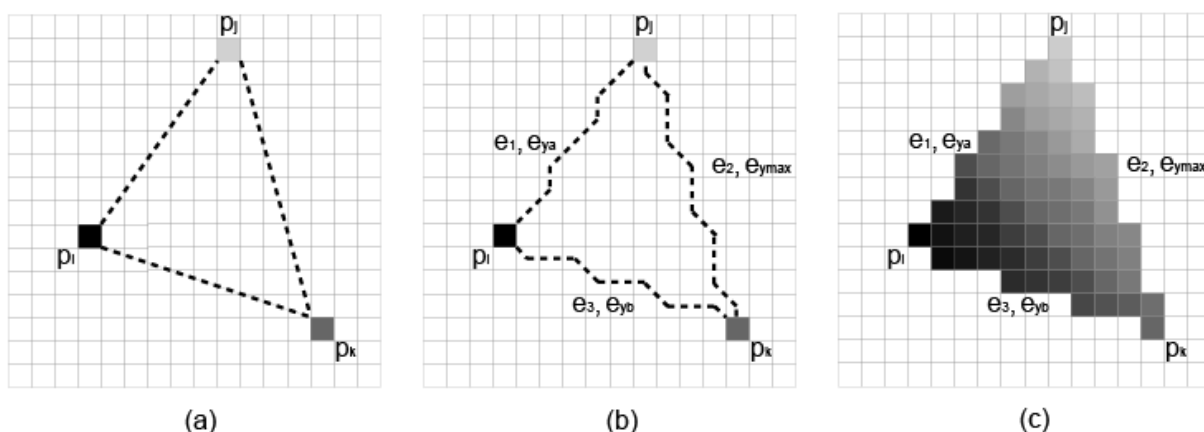
, kde $k_1 = \frac{y_j - y_i}{x_j - x_i}$, $k_2 = \frac{y_k - y_j}{x_k - x_j}$ a $k_3 = \frac{y_i - y_k}{x_i - x_k}$ představují směrnice přímek procházejících hranami a q_1, q_2, q_3 udávají místo jejich průsečíku s osou y .

Další postup, shodný pro všechny tři hrany, demonstrujeme na příkladu obecné hrany e s koncovými body $[x_a, y_a]$ a $[x_b, y_b]$. Nejprve zavedeme $\Delta y = y_b - y_a$ a $\Delta x = x_b - x_a$ jako rozdíly vzdáleností koncových bodů této hrany v odpovídajících osách. Pro jednoznačnost výpočtu je nutné úsečku transformovat tak, aby vždy procházela I. a V. oktantom (viz obr. 4.7). Důsledek této transformace je dosažení $x_a < x_b \wedge y_a < y_b$.

Při získávání rastrové reprezentace úsečky tedy rozhodujeme mezi případem, kdy každý další bod, vycházíme-li z libovolného $[x_i, y_i]$ takového, že platí $x_a < x_i < x_b \wedge y_a < y_i < y_b$, bude vložen na pozici $[x_i + 1, y_i]$ nebo $[x_i + 1, y_i + 1]$.

¹⁰ V případě, že podmínce vyhovují dvě hrany, vybereme libovolnou z nich.

Kapitola 4 | Triangulace



Obr. 4.6: (a) vrcholy trojúhelníka s asociovanými intenzitami, (b) Bresenhamovým algoritmem vytvořené hrany, (c) řádkové plnění a interpolace trojúhelníka

O tom, jaký z těchto bodů v dané fázi zvolit, rozhodují jejich vzdálenosti od bodu se stejnou souřadnicí x na původní přímce, viz (4.6). Proměnná y zde představuje jednu souřadnici vyšetřovaného bodu na původní přímce. Do vzorců je dále dosazeno právě za y z libovolné přímky ve (4.5). Jestliže platí $d_1 > d_2$, pak zvolíme bod $[x_i+1, y_i+1]$, neboť ten se nachází blíže původnímu bodu. V opačném případě $d_1 \leq d_2$ volíme $[x_i+1, y_i]$.

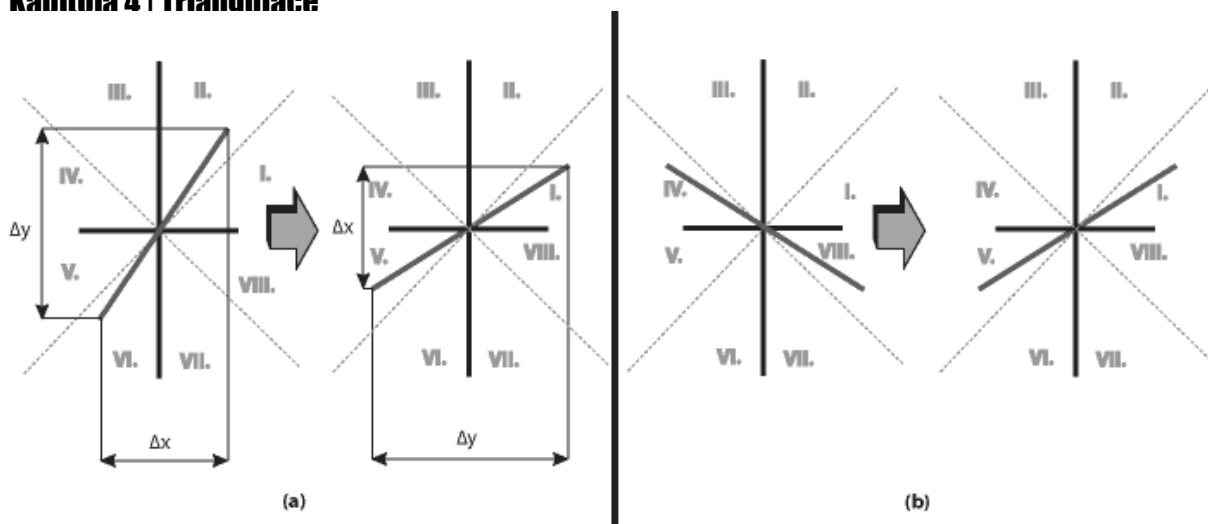
$$\begin{aligned} d_1 &= y - y_i = k(x_i+1) + q - y_i \\ d_2 &= y_i + 1 - y = y_i + 1 - k(x_i+1) - q \end{aligned} \quad (4.6)$$

Je možno dokázat [Žára98], že tento vztah lze přepsat do iterační podoby:

$$\begin{aligned} r_i \leq 0: & \quad r_{i+1} = r_i + 2\Delta y: \text{ posun vpravo} \\ r_i > 0: & \quad r_{i+1} = r_i + 2\Delta y - 2\Delta x: \text{ posun nahoru vpravo} \end{aligned}$$

, kde r_i označuje predikci pozice i -tého bodu. Za r_1 použijeme počáteční bod zkoumané hrany $[x_a, y_a]$. Nyní předpokládejme, že máme množiny $E_{y_{max}}$, E_{y_a} a E_{y_b} naplněny body představujícími rasterovou reprezentaci příslušných tří hran. Tyto body, jsou-li v každé z množin uspořádány sestupně podle souřadnice y , tvoří v E_{y_a} a E_{y_b} odpovídající páry vždy s jedním bodem množiny $E_{y_{max}}$. Každý jeden takový pár obsahuje koncové body rasterizované úsečky rovnoběžné s osou x , které udávají interval, na kterém se bude interpolovat.

Samotná interpolace trojúhelníku $t_{i,j,k}$ nyní představuje už jen aplikování jedné z níže uvedených metod na body všech úseček definovaných koncovými body ze zmíněných množin. Výpočet je možno realizovat algoritmem řádkového plnění. Výsledek je naznačen na obr. 4.6c.



Obr. 4.7.: (a) záměna os, (b) osová souměrnost podle y

4.2.1 Bilineární interpolace

Bilineární interpolace intenzity z bodu o souřadnicích $[x, y]$ v trojúhelníku $t_{i,j,k}$ s hranami $e_{i,j}, e_{j,k}, e_{k,i}$ a vrcholy v_i, v_j, v_k o souřadnicích $[x_i, y_i], [x_j, y_j], [x_k, y_k]$ s intenzitami z_i, z_j, z_k obnáší aplikaci lineární interpolace nejprve na dvou hranách pro body se souřadnicemi $[x_0, y], [x_1, y]$, kde $x_0 < x < x_1$. Po získání těchto dvou intenzit je teprve provedena interpolace na hraně mezi nimi pro vyšetřovaný bod.

Výpočet samotné lineární interpolace probíhá např. pro hranu $e_{i,j}$ tak, že je nejdříve sestavena parametrická rovnice přímky, která jí prochází

$$\begin{aligned} x &= x_i + t(x_j - x_i) \\ y &= y_i + t(y_j - y_i) \end{aligned} \tag{4.7}$$

Pro každý bod o souřadnicích $[x, y]$ ležící na úsečce je určena jeho intenzita z interpolací, kde t je parametr předchozí parametrické rovnice (4.7), jako

$$z = z_i + t(z_j - z_i) \tag{4.8}$$

4.2.2 Barycentrické souřadnice

Uvnitř obecného trojúhelníka $t_{i,j,k}$ s vrcholy v_i, v_j, v_k a souřadnicemi $[x_i, y_i], [x_j, y_j]$ a $[x_k, y_k]$ představují barycentrické souřadnice libovolného bodu váženou sumu v_i, v_j, v_k koeficienty $\lambda_1, \lambda_2, \lambda_3$ takovými, že platí

$$\lambda_1 + \lambda_2 + \lambda_3 = 1 \tag{4.9}$$

Jsou-li ve vrcholech zavedeny intenzity z_i, z_j, z_k , pak polohu vyšetřovaného bodu popisuje soustava v maticové notaci (4.10). Vzhledem k tomu, že neznáme všechny tři barycentrické

Kapitola 4 | Triangulace

souřadnice bodu a zároveň jeho intenzitu, je tato soustava nedourčena. Dosazením za $\lambda_3 = 1 - \lambda_1 - \lambda_2$ z (4.9) vznikne (4.11).

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_i & x_j & x_k \\ y_i & y_j & y_k \\ z_i & z_j & z_k \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \quad (4.10)$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_i - x_k & x_j - x_k & x_k \\ y_i - y_k & y_j - y_k & y_k \\ z_i - z_k & z_j - z_k & z_k \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ 1 \end{bmatrix} \quad (4.11)$$

Zde z prvních dvou rovnic, kde neznáme pouze λ_1, λ_2 , tyto barycentrické souřadnice vypočteme. Jejich dosazením do rovnice třetí získáme výslednou intenzitu z ve zkoumaném bodu (4.12). V případě, že některý ze jmenovatelů zlomků vychází roven nule, ošetříme výpočet dočasným prohozením souřadnicových os.

$$\lambda_2 = \frac{(y_1 - y_3)(x_3 - x) - (x_1 - x_3)(y_3 - y)}{(x_1 - x_3)(y_2 - y_3) - (y_1 - y_3)(x_2 - x_3)}$$
$$\lambda_1 = -\frac{(x_j - x_k)\lambda_2 - (x_k - x)}{(x_i - x_k)} \quad (4.12)$$
$$z = (z_i - z_k)\lambda_1 + (z_j - z_k)\lambda_2 + z_k$$

Je důležité poznamenat, že výše uváděný výpočet je potřeba provést celý znova pro každý bod trojúhelníka a tím je celý výpočet podstatně pomalejší než např. v kap. 4.2.1 a 4.2.3.

4.2.3 Soustava rovnic

Základem pro výpočet intenzity z pro libovolný bod p v ploše trojúhelníku $t_{i,j,k}$ je znalost soustavy rovnic, která vyjadřuje lineární závislost intenzit ve vrcholech na jejich poloze:

$$\begin{aligned} ax_i + by_i + c &= z_i \\ ax_j + by_j + c &= z_j \\ ax_k + by_k + c &= z_k \end{aligned} \quad (4.13)$$

Koeficienty a , b a c , které řešením soustavy (4.13) získáme, použijeme posléze v rovnici určující vztah polohy bodu p k intenzitě z . V případě, že odečteme druhý a třetí řádek od prvního, získáme soustavu dvou rovnic.

$$\begin{aligned} a(x_i - x_j) + b(y_i - y_j) &= z_i - z_j \\ a(x_i - x_k) + b(y_i - y_k) &= z_i - z_k \end{aligned} \quad (4.14)$$

Přepíšeme-li soustavu (4.14) do maticové notace, resp. do podoby matice I , můžeme snadno

Kapitola 4 | Triangulace

získat determinant takové matice.

$$\det(I) = (x_i - x_j)(y_i - y_k) - (x_i - x_k)(y_i - y_j) \quad (4.15)$$

Determinant (4.15) použijeme k řešení soustavy za pomoci Cramerova pravidla.

$$\begin{aligned} a &= \frac{(z_i - z_j)(y_i - y_k) - (z_i - z_k)(y_i - y_j)}{\det(I)} \\ b &= \frac{(x_i - x_j)(z_i - z_k) - (x_i - x_k)(z_i - z_j)}{\det(I)} \\ c &= z_i - ax_i - by_i \end{aligned} \quad (4.16)$$

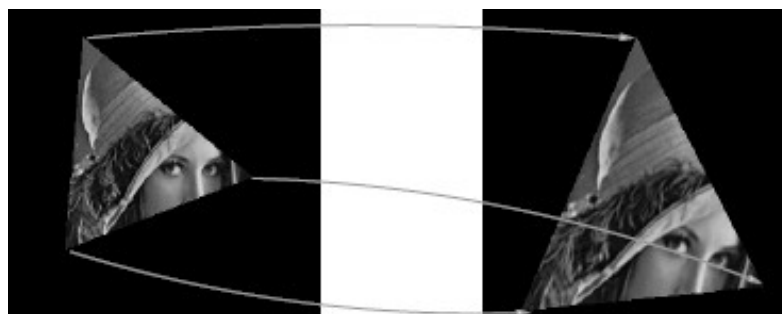
Při znalosti koeficientů a, b, c intenzitu z bodu $p = [x, y]$ určíme z následujícího lineárního vztahu. Výhodou tohoto přístupu je fakt, že koeficienty a, b, c počítáme pro každý trojúhelník pouze jednou.

$$z = ax + by + c \quad (4.17)$$

4.2.4 Warping se třemi páry úseček

Warping nepředstavuje pravou interpolaci ve smyslu metod popsanych v předcházejících kapitolách. K získání kompletního obrazu totiž není využita intenzita ve vrcholech triangulace, ale změna jejich polohy oproti nějaké předcházející. Pro každý trojúhelník triangulace můžeme použít algoritmus popisovaný v [Žára98], který je natolik rozsáhlý a důkladně popsán, že by jeho uvádění zde postrádalo na významu.

Na obr. 4.8 lze vidět názornou demonstraci toho, jakým způsobem je obraz warpován. Každý bod plochy trojúhelníka je jako v případě výše popisovaných interpolací navštíven při průchodu algoritmu řádkového plnění po předchozí aplikaci Bresenhamova algoritmu (viz kap. 4.2).



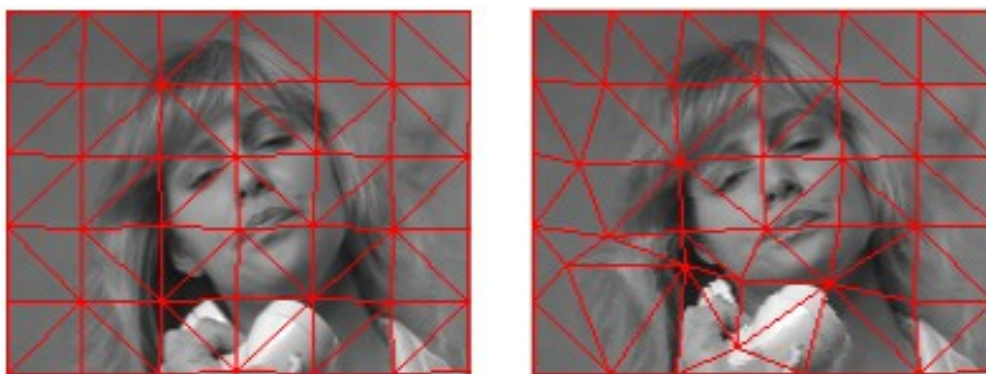
Obr. 4.8: Warping trojúhelníkové oblasti; převzato z [Marq00]

5 Hledání relevantních bodů ve snímku, existující řešení

Statický jednosložkový obraz, jakožto matici $M \times N$ pixelů, lze reprezentovat rovněž za pomoci Delaunayovy triangulace. Důvody, proč použít tuto reprezentaci, jsou v podstatě dva.

Jedním je odstranění redundantní informace anebo odstranění takové informace, která je pro subjektivní vnímání obrazu nepodstatná. Tohoto cíle dosahujeme odlišnými prostředky než např. v případě často používaného ztrátového formátu JPEG. Tedy očekáváme, že některé nedostatky komprese JPEG, jako je ztráta detailu, bloky a globální ztráta kvality při opakovaném zakódování, nebudou přítomny anebo jen v omezené míře. Druhým důvodem je nemožnost provádění geometrických transformací nad zakódovaným obrazem v případě JPEG, kdy podstatou reprezentace jsou koeficienty diskrétní kosinové transformace. Rovněž JPEG2000 není možné tímto způsobem upravovat, neboť obraz zakóduje do koeficientů subpásem diskrétní vlnkové transformace. Reprezentace obrazu triangulací naproti tomu představuje možnost, jak změnou morfologie samotné triangulace dosáhnout po dekódování odpovídající změny v matici obrazu (viz obr. 5.1).

Otázkou zůstává, jakým způsobem budeme konstruovat triangulaci, abychom získali co možná nejpřesnější reprezentaci původního obrazu při co nejmenším počtu trojúhelníků. Ověřené způsoby, jakými takovou triangulaci získat, spadají do několika základních kategorií. Jednou možností je vybrat určitý počet náhodně či uniformně rozmístěných bodů. Jako další je hledání relevantních bodů na hranách a v jejich blízkosti. Jestliže jsou tyto body voleny s ohledem na skutečnost, že hrany musí být hustě pokryty trojúhelníky, pak je tato úvaha správná. Další možností jsou konvergenční metody vycházející buď z kompletní triangulace, resp. z pouhého rámce ohraničujícího obraz, které body v závislosti na lokálním vyhodnocení chyby odebírají, resp. přidávají.

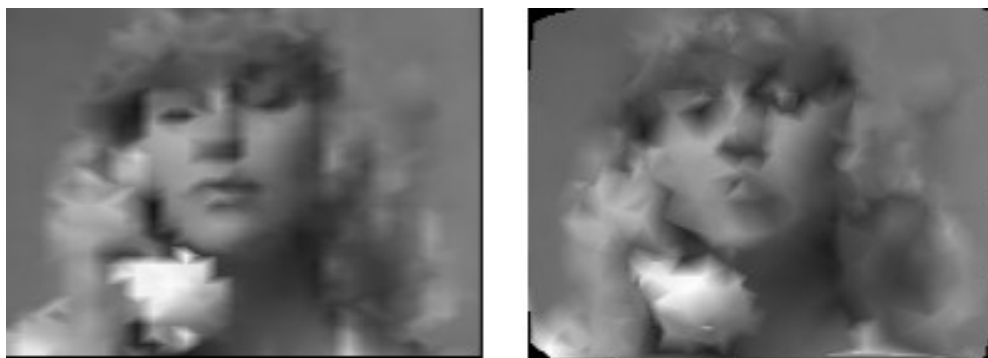


Obr. 5.1: Změna morfologie sítě ve vybraných bodech a následný warping nad maticí snímku

Vzhledem k tomu, že většina metod, vyjma některých speciálních případů při konstrukci uniformní triangulace, produkuje DT s obecnou (neobdélíkovou) konvexní obálkou, která nepokrývá celou plochu obrazu, je pro dosažení vyvážené kvality nutné přidat některé body na okrajích obrazu. Můžeme takto přidat body v rozích a v půli okrajů obrazu anebo také všechny body z okrajů obrazu. Záleží zde na preferencích pro kvalitu a objem reprezentace.

5.1 Náhodný výběr a uniformní triangulace

Pokud je při získávání relevantních bodů časová složitost nejdůležitějším kritériem a na kvalitu není kladen velký důraz, je možné velice jednoduše aproximovat snímek náhodným výběrem anebo uniformní triangulací (viz obr. 5.2). Vzhledem k nerespektování hran lze ale očekávat, že při subjektivním porovnávání taková reprezentace nebude dobře ohodnocena.



Obr. 5.2: 4% bodů z uniformní triangulace a z náhodného výběru

Uniformní triangulaci lze vytvořit na základě čtvercové anebo obdélníkové mřížky, kdy každá z buněk takových mřížek je dělena na dva trojúhelníky. V prvním případě lze tímto způsobem vložit jen $100/i^2[\%]; i=1,2,\dots,\min\{M-1,N-1\}$ bodů, kde i je délka hrany čtverce a $M \times N$ je rozměr obrazu. Mimo to, dosáhnout stoprocentního pokrytí obrazu triangulací je tímto způsobem nemožné pokud $M \neq N$. Rovněž dosáhnout pokrytí až k jednomu z okrajů snímku není vždy jednoduché, zvláště když M nebo N je prvočíslo. Ve druhém případě je situace obdobná s tím rozdílem, že stoprocentního pokrytí lze teoreticky dosáhnout i při $M \neq N$. Při menším procentu vkládaných bodů poskytuje mřížka poměrně slušnou aproximaci, ale pokud jde o vyšší procento, výsledný dojem z obrazu kazí symetrické artefakty, na které je lidské vnímání velmi citlivé. Uniformní triangulace se proto nejvíce hodí pro demonstrativní účely, jako na obr. 5.1.

Náhodný výběr nejčastěji s rovnoměrným rozdělením pravděpodobnosti představuje pravděpodobně tu nejintuitivnější metodu, kterou lze okamžitě získat množinu bodů z plochy celého obrazu. Pro menší počet bodů je subjektivní dojem z věrnosti aproximace malý, zato při větším počtu vkládaných bodů se na výstupu nesetkáme se symetrickými artefakty tak jako v případě uniformní triangulace.

V [Krey01] je DT zkonstruovaná z náhodných bodů použita jako počáteční konfigurace pro vstup do algoritmu provádějícího simulované žihání, které je mj. popsáno v kap. 6.2. Podstatnou inovací je, že vyšetřované body nejsou ubírány a přidávány, ale je jimi na základě tohoto algoritmu pohybováno do míst, kde tolik nepřispívají k celkové chybě. Tento zajímavý přístup nicméně vede k tomu, že produkované trojúhelníky jsou na hranici toho, co bychom očekávali od DT, tedy lze v nich najít extrémně malý úhel.

5.2 Hranová detekce

Velká skupina metod se zaměřuje na prioritní reprezentaci hran. Vychází se zde totiž z poznatku, že lidské vnímání je na konzistenci hran citlivé z důvodu, že podle nich určuje informace o tvaru a skutečné velikosti objektů.

Pro další účely máme diskrétní obrazovou funkci $f(x, y), x \in \{0, \dots, M-1\}, y \in \{0, \dots, N-1\}$ definovanou nad rastrovým obrazem velikosti $M \times N$ vracející intenzitu z rozsahu $\langle 0; 255 \rangle$.

K získávání hranových bodů v $f(x, y)$ se nejlépe hodí hranové detektory. Jedná se nezřídka o víceřadkové algoritmy, které se shodují v tom, že v určité fázi provádějí konvoluci (5.1) s pro ně specifickým konvolučním jádrem¹¹ a určí tak podstatné změny v intenzitách v různých směrech daných povahou konkrétního jádra.

$$\hat{f}(x, y) = f(x, y) * h(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k f(i, j) h(x-i, y-j) \quad (5.1)$$

, kde \hat{f} je zpracovaný obraz a h představuje konvoluční jádro, tedy čtvercovou matici s rozměrem k . Souřadnice i, j udávají polohu uvnitř jádra a souřadnice x, y uvnitř obrazu. Podrobíme-li celý vstupní obraz (viz obr. 5.3a) průchodu hranového detektoru (viz obr. 5.3b), je následně třeba výsledek konvoluce prahovat (viz obr. 5.3c)

Parciální derivace ve směrech $(1,0)$ a $(0,1)$ jsou $\partial f / \partial x = f(x, y) - f(x-1, y)$ a $\partial f / \partial y = f(x, y) - f(x, y-1)$. Aproximaci vyhledávající maxima těchto derivací lze spatřit v hranových detektorech Sobel, Prewittová, Roberts, aj.

Vedle prvních derivací se lze setkat i s metodami založenými na druhých parciálních derivacích, vyjádřených jako $\partial^2 f / \partial x^2 = (f(x+1, y) - f(x, y)) - (f(x, y) - f(x-1, y))$ a $\partial^2 f / \partial y^2 = (f(x, y+1) - f(x, y)) - (f(x, y) - f(x, y-1))$ (Marr-Hildreth, Laplacian). Zvláště druhé derivace jsou citlivé na šum, proto by v nejlepším případě mělo konvoluci s takovým hranovým detektorem předcházet gaussovské rozostření.



Obr. 5.3.: (a) vstupní obraz, (b) po konvoluci s Laplaciánem, (c) prahování výstupu

¹¹ Používá se též označení maska.

Kapitola 5 | Hledání relevantních bodů ve snímku, existující řešení

V [Pras06] je popisován algoritmus, který v počátku vygeneruje hranové body za pomoci Canny hranového detektoru. Po procesu spojení těchto bodů v množinu úseček jsou tyto hrany jsou přijímány do CDT (viz. kap. 4.1.2) jako povinné. Výsledek je, že trojúhelníky jsou naskládány mezi hranami a žádný z nich je neprotíná. Plocha každého trojúhelníka je navzorkována metodou Monte Carlo a tak je odhadnuta jeho výsledná barva. Takto vzniklé trojúhelníky nazývají autoři trixely. Trixely jsou poté podle různých kritérií založených na vlastnostech lidského vnímání sdružovány do skupin za pomoci procházení grafu. Z trojúhelníků se tak stávají polygony o větším počtu hran, přičemž jejich barva je průměrována z jednotlivých dílčích trixelů.

[Garc99] uvádí metodu, která je založena na výběru hranových bodů způsobem adaptivního vzorkování. Z těchto bodů je zkonstruována DT, která je dále dělena, dokud není dosaženo přijatelné chyby. Jedná se tedy o postup podobný metodám z kapitoly 5.3, ale výchozím bodem je triangulace vzniklá z hran.

5.3 Adaptivní triangulace

V případě, kdy se vychází z kompletní triangulace obsahující každý pixel obrazu, resp. z triangulace obsahující pouze čtyři pixely v rozích obrazu, je nalezení optimální triangulace otázkou dosažení minimální chyby při určitém množství odebíraných (decimace), resp. vkládaných bodů (refinace). Výsledná aproximace je z uvedených třech typů metod pro získávání relevantních bodů nejpřesnější, ale zároveň se jedná o metody časově nejnáročnější.

Zajímavá metoda je popisována v [Rila98], kde autoři konstruují triangulaci refinací ze čtyř rohových bodů obrazu a bodu, ve kterém je největší odchylka od průměrné intenzity v celém obraze. Na základě kvality dosavadní aproximace vybírají bod v takovém trojúhelníku, který je dle metriky SSE aproximován nejhůře. Poté přistupují k výběru nejhůře reprezentovaného bodu v trojúhelníku. Ten je určen maximální absolutní hodnotou rozdílu s odpovídajícím pixelem v původním obraze. Po vložení bodu do triangulace jsou nové trojúhelníky vytvořeny a interpolovány a celý proces se opakuje. Autoři dále objasňují postup pro eliminaci artefaktů na hranách vzniklých nerespektováním hran při výběru bodů. Navrhli dodatečné vylepšení v podobě rozostření s maskou 3x3 a následný výběr nejhorších oblastí a bodů v takto zpracovaném obraze.

V současné době byla v [Koho07] publikována decimační metoda, která vychází z kompletní DT. Nejprve je vyhodnocena důležitost každého z vrcholů a podle ní jsou vrcholy seřazeny do prioritní fronty. Vrcholy jsou poté po vyjmutí z ní odstraňovány z triangulace do té doby, než je dosaženo zadaného maximálního počtu. Co se týče kritérií, podle kterých jsou vrcholy řazeny, autor otestoval šest různých přístupů, z nichž s přehledem nejlépe vyšel algoritmus BRUTE. Význam vertexu je s ní počítán jako součet MSE trojúhelníků, které by byly zkonstruovány, kdyby byl daný vertex odstraněn. Tato metoda používající algoritmus BRUTE je rovněž v kap. 9.2.4 podrobena testům.

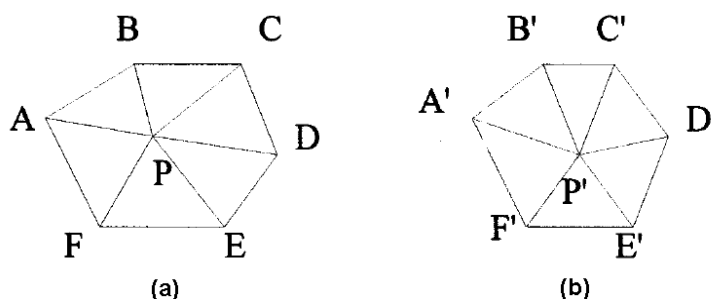
6 Existující řešení na bázi triangulací pro video

Ačkoliv zkušenosti s triangulacemi pro statický obraz hovoří o nahrazení bitmapových dat trojúhelníkovou sítí povětšinou optimisticky a to bez ohledu na použitou metodu, myšlenka o promítnutí této domény do sekvence navazujících snímků je často opomíjena. Přínosy trojúhelníkové reprezentace jsou u videa přitom větší než u statického snímku. Jedná se tedy o snížení redundance v obrazové informaci a možnost provádění geometrických transformací pouze nad trojúhelníkovou sítí namísto nad všemi pixely každého snímku videosekvence. Nad to, zohledníme-li podobnost triangulací jednotlivých snímků videa, můžeme mezi nimi hledat korespondence na základě skutečnosti, že některé trojúhelníky jsou obsaženy v předchozím snímku. Metody, které v této oblasti vznikly, se dají rozdělit podle způsobu aplikace triangulace na vstupní snímky a využití jejich podobnosti.

6.1 Adaptivní triangulace a odhad pohybu

Následující dvě metody na rozdíl od pouhého nacházení vhodné reprezentace jednotlivých snímků uvažují právě jejich zmíněnou podobnost. Přímé korespondence mezi dvojicemi triangulací ale nestanovují. Pouze předpokládají, že pokud máme výchozí snímek videosekvence reprezentován triangulací, potom můžeme této triangulaci snímek po snímku posouvat body s tím cílem, že aproximace každého následujícího snímku bude podobná té ve vzorové videosekvenci. Přínos z toho plynoucí je v podstatně menším objemu dat, která potřebujeme pro reprezentaci odvozeného snímku .

[Yaop98] popisuje metodu ztrátového kódování vycházející buď z pravidelné trojúhelníkové sítě anebo ze získání adaptivní Delaunayovy triangulace refinací. Konstrukce refinované DT začíná vložením dvou trojúhelníků pokrývajících celý obraz beze zbytku. Každý z těchto trojúhelníků je dále rekurentně dělen v případě, že aproximace dané plochy trojúhelníkem je příliš hrubá, což kontroluje zadaný práh. Dělení probíhá za pomoci vložení bodu do těžiště trojúhelníku a následného vzniku tří nových trojúhelníků, a to tak dlouho, dokud není překročen počet iterací anebo aproximace všech trojúhelníků triangulace nepřekračuje odchylku danou prahem.



Obr. 6.1: (a) bod triangulace spolu s přilehlými trojúhelníky, (b) metodou UMA deformovaný polygon [Yaop98]

Kapitola 6 | Existující řešení na bázi triangulací pro video

Další úprava je mírně sofistikovanější. V rámci každého z přilehlých trojúhelníků každého bodu P triangulace (viz obr. 6.1a) je vypočtena průměrná hodnota intenzity. Pokud se tato hodnota příliš neliší od průměrné intenzity polygonu tvořeného přilehlými trojúhelníky nebo plocha celého polygonu je menší než zadaný práh, pak je bod P považován za redundantní a je vypuštěn. Tento krok je opakován do té doby, dokud není dle zmíněných kritérií shledána absence redundantních bodů.

Odhad pohybu provádí klasický BMA, který nejprve středy jednotlivých bloků s volitelnou velikostí hrany i velikostí vyhledávacího okénka ztotožní s body zkonstruované triangulace. V návaznosti na zjištění pohybu v obraze dojde k pohybu triangulace. Takovou kompenzaci pohybu však autoři označují za příliš hrubou. Dále uvádějí vlastní algoritmus UMA, který má za úkol zmíněnou kompenzaci zlepšit. Jedná se o cílený pohyb bodem P (viz obr. 6.1b) v diskrétním intervalu $\langle -2, 2 \rangle$ následovaný zjištěním MSE mezi předchozím vstupním obrazem ohraničeným výchozím polygonem a predikovaným obrazem pod takto upraveným polygonem. Kompenzace pohybu poté probíhá s takovým vektorem, pro nějž polygon vykazuje nejnižší MSE. Po dekódování je snímek vykreslován interpolací ze vzorků ve vrcholech.

V [Srik05] se můžeme setkat s podobným přístupem jako u výše uvedené metody, i když zde jsou namísto videa bezztrátově reprezentovány snímky získané pomocí MR. Souvislost s videem je možné spatřovat v tom, že zde jsou kódovány postupné řezy tělem pacienta, mezi kterými lze snadno hledat souvislosti a dokonce je i jako video s určitou snímkovou frekvencí přehrát. Pojetí konstrukce triangulace odpovídá povaze reprezentovaných dat. Nejprve je prahováním snímek rozdělen na popředí (tělo pacienta) a pozadí. K tomuto účelu je vytvořena binární maska, nepatrně zvětšená morfologickou erozní operací. Případné díry v přední části masky jsou přitom označeny za popředí a zbylé pozadí se dále neuvažuje.



Obr. 6.2: (a) původní snímek, (b) bitová maska, (c) potlačené pozadí [Srik05]

Před tvorbou DT jsou nejprve všechny pixely popředí označeny jako nenavštívené. Pro dva konsekventní snímky je vypočtena průměrná DFD (6.1)

$$DFD_{avg} = \sum_{(x,y)} \frac{DFD^2(x,y)}{K} \quad (6.1)$$

Kapitola 6 | Existující řešení na bázi triangulací pro video

, kde K je počet nenavštívených pixelů a $DFD(x, y) = \mathfrak{T}(x, y, t) - \mathfrak{T}(x - \partial x, y - \partial y, t + 1)$ je rozdílový snímek vzniklý odečtením predikovaného a předchozího snímku. Ve snímku jsou posléze detekovány hrany vícekrokovým hranovým detektorem. Na nich jsou po jednom vytvářeny body triangulace v nenavštívených pixelech v zadané vzdálenosti od navštívených. Okolo každého z nově vzniklých bodů je vytvořen kruh, při jehož zvětšování je testováno, zda $\sum (DFD^2(x, y)) > DFD_{avg}$. Ve chvíli, kdy se podaří tuto nerovnost splnit, jsou všechny pixely uvnitř kruhu označeny jako navštívené. Poslední část algoritmu je počínaje výpočtem DFD_{avg} použita na výpočet každého dalšího vrcholu triangulace až do chvíle, kdy je dosaženo jejich zvoleného maximálního počtu nebo vzhledem ke kritériu vzdálenosti nelze další bod vložit. Odhad pohybu vykonává jako ve výše popisované metodě BM a dekódované snímky jsou ze vzorků rovněž vykreslovány bilineární interpolací. Pro dosažení bezeztrátového kódování je zohledněno residuum $r(x, y) = \mathfrak{T}(x, y, t) - \hat{\mathfrak{T}}(x, y, t)$ představující odchylku zakódovaného a původního snímku. To je kódováno aritmetickým kódováním a uloženo společně s vektory posunu.

6.2 Tetraedrizační

Chápeme-li čas jako třetí rozměr videa a seřadíme-li snímky při pohledu kolmo na ně přímo za sebe, potom můžeme vytvořit objemová data mající pixely reprezentovaných snímků jako vzorky ve vrcholech. Právě na objemových datech byl algoritmus poprvé s úspěchem testován v souvislosti s návrhem ztrátové komprese pro video. Princip byl poprvé využit ve [Varg07] při vytváření Delaunayovy tetraedrizační. Ta je z objemových dat konstruována třemi různými přístupy.

První přístup představuje inkrementální vkládání s lokálním prohazováním hran. To spolehlivě zajistí, že všechny hrany budou po jeho průchodu lokálně delaunayovské. Základní myšlenkou je vkládání nových bodů tam, kde je v aktuální reprezentaci největší RMSE. Podstatně se tím sníží chyba v určitém okolí takových vkládaných bodů. Zastavovací podmínka je dosažení určité RMSE nebo vložení maximálního počtu bodů.

Druhým přístupem je genetický algoritmus, který používá jako heuristiku chybu (6.2) dle RMSE ve 3D, kde $f(i, j, k)$ vrací intenzitu v původních datech a $g(i, j, k)$ v aproximovaných. Konstanty m, n, o jsou rozměry kvádru ohraničujícího data.

$$RMSE = \sqrt{\frac{1}{mno} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \sum_{k=0}^{o-1} \|f(i, j, k) - g(i, j, k)\|^2} \quad (6.2)$$

Genotypem řešení je 3D bitová maska shodně velká se vstupními daty, která představuje přítomnost bodu na dané pozici. Podle počtu zadaných vstupních bodů je vygenerován určitý počet náhodných jedinců (řešení). Nejvhodnější jedinci jsou vybíráni náhodně s vahou, která odpovídá výsledku heuristiky. Tím, že v dalších generacích mohou být obsaženi i jedinci slabší se předejde se uvíznutí algoritmu v lokálním minimu, neboť i ti mohou nést část optimálního řešení. Samotná reprodukce probíhá za pomoci genetických operátorů mutace

Kapitola 6 | Existující řešení na bázi triangulací pro video

a křížení. Podmínkou pro ukončení tetradrizace je dosažení určité RMSE.

Posledním zde použitým přístupem je simulované žihání. Vychází z náhodně generované konfigurace bodů nebo z výstupů libovolného ze dvou výše uvedených přístupů. Pravděpodobnost přijetí stavu s vyšší energií je definována jako

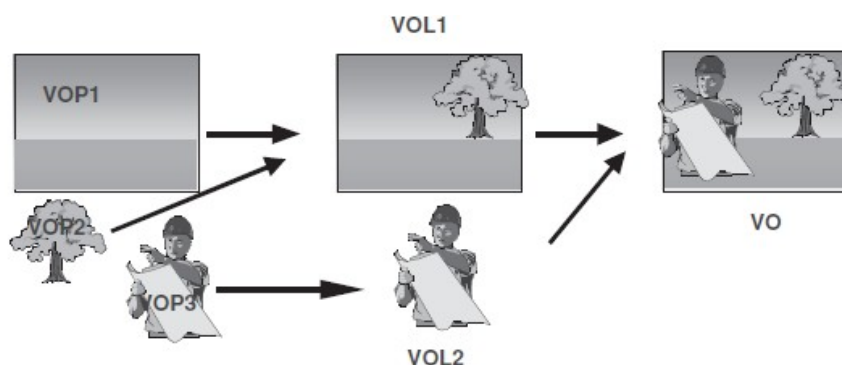
$$p(\Delta E) = e^{-\frac{\Delta E}{T}} \quad (6.3)$$

, kde ΔE je změna energie reprezentovaná RMSE (6.2) a T je teplota, která je exponenciálně snižována. Vznik nové konfigurace je vždy realizován odebráním libovolného počtu bodů ze stávající konfigurace a přidáním jiných z původní datové množiny. Zastavovací podmínkou je dosažení určité RMSE.

Relativní výhodou tetradrizace je, že všechny snímky videosekvence jsou kódovány najednou, jako jedna datová množina. Nevýhodou je poměrná nepřesnost kvůli způsobu převodu videosekvence do 3D a následně těžko definovatelná podobnost mezi snímky. Tato metoda byla testována pro videa typu renderované video, statická kamera a pohyblivá kamera, nejlepší výsledky poskytovala pro statickou kameru.

6.3 Triangulace 2D a 3D objektů ve videu

Základním vylepšením, které s sebou přinesl kodek MPEG-4¹² [ISO01], je hierarchické rozdělení videa na plochy (VOP), vrstvy (VOL) a objekty (VO). Význam těchto komponent pro konstrukci celkového snímku je patrný z obr. 6.3. VOP jsou obecně tvarované a pokud je jich se stejnými vlastnostmi více, mohou být sdružovány do skupin (GOV) a ty pak podle polohy vzhledem ke kameře do VOL. Každý VO je kódován nezávisle na ostatních. Platí, že VOP a VOL obecně nepokrývají celou plochu snímku, takže je-li s nimi pohybováno, je kódovaný popis poměrně úspornější než v případě předpokladu, že se pohybuje snímek jako celek.



Obr. 6.3: Kompozice scény z rovin a vrstev; převzato ze [Scha06]

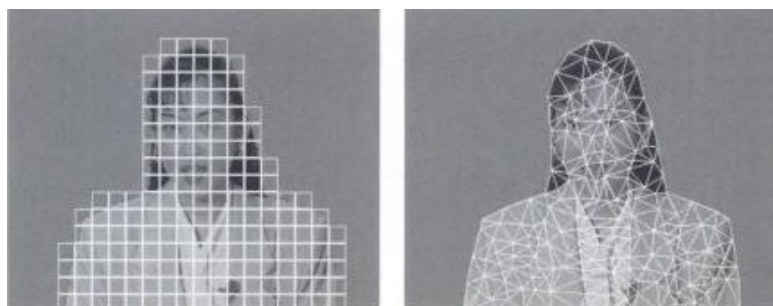
Kromě výše uvedené hierarchie kódované nejčastěji negeometrickým přístupem se lze

¹² Konkrétně jeho část MPEG-4 Visual

Kapitola 6 | Existující řešení na bázi triangulací pro video

setkat s objekty kódovanými za pomoci 2D nebo 3D trojúhelníkových sítí. Konkrétně je použita Delaunayova triangulace. Stejný přístup je zvolen i v [Chen99].

Původní záměr, proč tyto objekty používat, vyšel z potřeby lépe kódovat syntetická videa a potěšit tak 2D/3D animátory. Pokud je totiž např. popisován objekt, který se otáčí podle vertikální osy, pak vzdálenosti pixelů textury, která mu byla přiřazena, už nejsou déle uniformní. Tento problém řeší triangulace takového objektu před pohybem. Po vyšetření jejího pohybu jsou přepočteny pozice pixelů v asociované textuře a k žádným problémům s neuniformností nedochází.



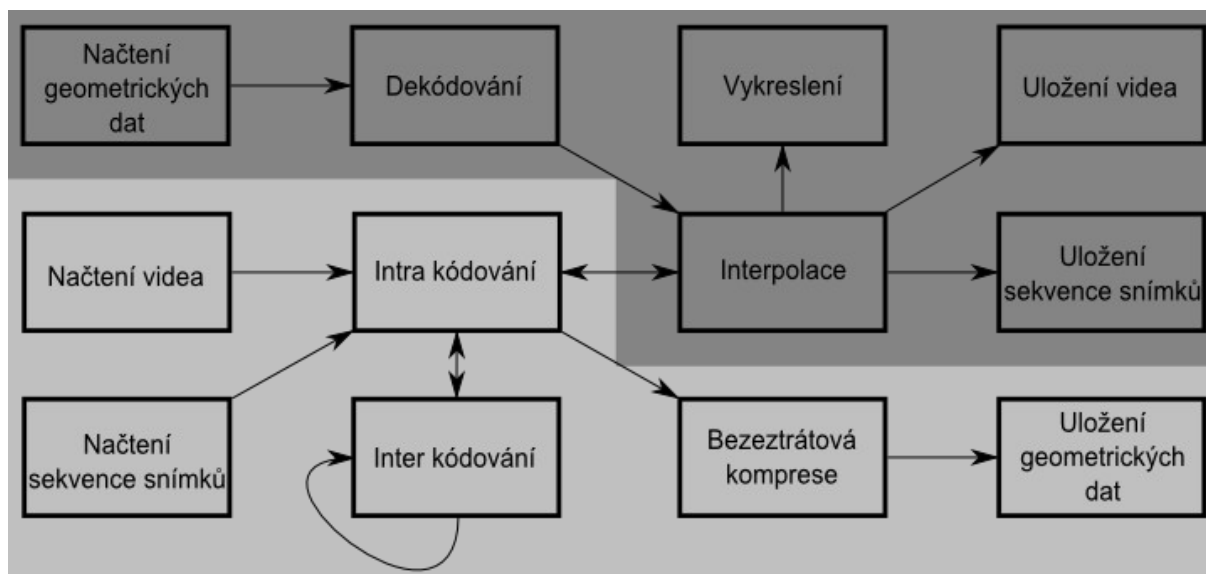
Obr. 6.4: Objekt kódovaný bloky a objekt kódovaný triangulací; převzato z [Chen99]

Proces triangulování objektu probíhá ve dvou fázích. Nejprve je nalezena hranice VOP extrakcí ze speciální alfa-viditelnostní roviny. Body, které jsou odsud získány, slouží jako obálka budoucí triangulace. Druhá fáze je výběr vnitřních bodů a triangulace všech bodů včetně těch z obálky. Konstrukci triangulace je možné pojmut jako tvorbu uniformní sítě anebo jako datově závislou triangulaci (DDT). V prvním případě jsou sice body rozmístěny uniformně uvnitř objektu, ale v sousedství obálky vznikají početnější shluky, které zde povedou k obecně menším trojúhelníkům. Výhoda spočívá v tom, že není nutné kódovat pozice vnitřních bodů, neboť ty mohou být snadno vypočteny v dekodéru při přehrávání. Ve druhém případě je snaha o ztotožnění hran v triangulaci s hranami danými proměnnou intenzitou v obraze. Hustota pokrytí je v obou případech úměrná lokální pohybové aktivitě, kterou kodér také vyhodnocuje. Zbytek algoritmu výběru bodů je založený na DFD a je totožný se [Srik05] v kap. 6.1. Objekty jsou v obou případech Delaunayovsky triangulovány s kritérii kladenými na CDT z kap. 4.1.2, kde povinné hrany jsou hranami z obálky. Předchází se tak triangulaci mimo obálku, protože ta nemusí být konvexní.

Způsob, jakým je pohyb odhadován, se dotýká bodů na obálce i bodů uvnitř triangulace objektu. Použit je dopředný odhad, tedy z pozice objektu v čase $t-1$ je odhadnuta pozice v čase t . V [ISO01] je použit algoritmus block matching, naproti tomu v [Chen99] byl údajně s úspěchem otestován například ještě algoritmus KLT (viz kap. 3.2). Kompenzaci pohybu pak obstarává warping se třemi páry úseček (viz kap. 4.2.4).

7 Návrh vlastní metody

Hlavní myšlenkou na počátku úvah ohledně realizace bylo rozdělení metody na část představující kodér a část představující dekodér. Dále bylo potřeba určit vstupy a výstupy těchto dvou celků. Jako vstupní data kodéru slouží soubor videa (resp. sekvence statických snímků). Výstupem kodéru jsou geometrická data. Dekodér naproti tomu provádí opačný proces, tedy z geometrických dat produkuje soubor videa (resp. sekvenci statických snímků) nebo toto video přímo zobrazuje. Blokové schéma je možno vidět na obr. 7.1.



Obr. 7.1: Schema navržené metody; kodér – světlá sekce, dekodér – tmavá sekce

Jak vyplývá ze schématu, kodér a dekodér nejsou striktně odděleny. Hrana jdoucí z kodéru do dekodéru naznačuje, že pro potřeby náhledu je možno zakódované snímky interpolovat a posléze zobrazit či uložit. V opačném směru může být využito interpolace zakódovaného intra snímku k vytvoření inter kódovaných snímků.

7.1 Kodér

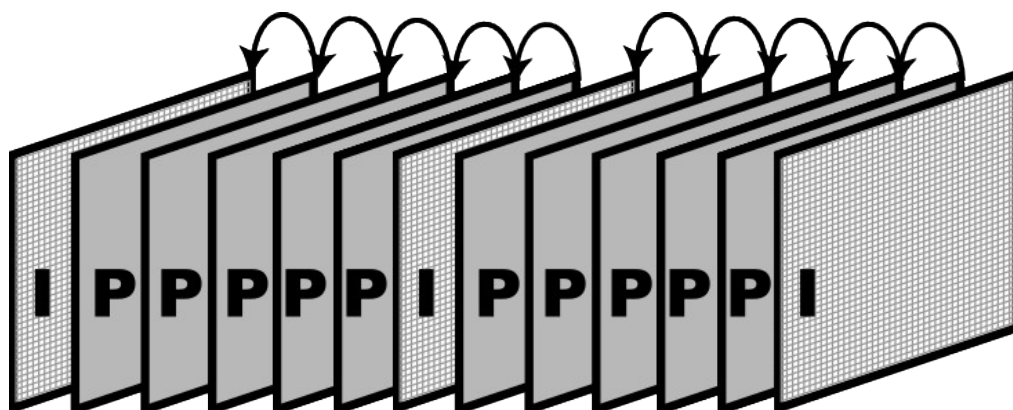
Smyslem kodéru je vytvoření aproximace vstupní videosekvence tak, aby residuum vzniklé odečtením dvou odpovídajících snímků (viz obr. 7.2) bylo minimální. Dále je potřeba minimalizovat časovou složitost a maximalizovat kompresní poměr.



Obr. 7.2: (a) původní snímek; (b) aproximace původního snímku; (c) jejich residuum

Návrh typu kódování

Vzhledem ke všem požadavkům na kodér bylo uvažováno nad tím, zda snímky kódovat pouze intra, a tedy s nízkou časovou složitostí, vysokou přesností a relativně malým kompresním poměrem. Další možností bylo zahrnout také inter kódování a tím zvýšit kompresní poměr. Výsledkem je návrh po vzoru MPEG (viz kap. 2.6.1), kde jsou snímky videa po vstupu do kodéru rozděleny na snímky klíčové a snímky dopředně predikované. Snímky obousměrně predikované nejsou pro zjednodušení a sporný přínos pro kompenzaci pohybu a obrazovou kvalitu uvažovány. Navržená GOP (viz obr. 7.3) obsahuje snímky, které se odkazují vždy na předchozí snímek. Vyjimku tvoří snímek klíčový, který představuje okamžik vyzískání nových absolutních údajů o složení scény. Interval mezi klíčovými snímky (velikost GOP) je vždy konstantní a je volen manuálně tak, aby frekvence, s jakou klíčové snímky přicházejí, co nejvíce odpovídala výskytu totálních změn ve scéně. Tím rozumíme vznik objektů, které v obraze v klíčovém snímku nebyly, nebo naopak objevení se nových. Lze tedy předpokládat, že tímto způsobem můžeme docílit rovněž samostatného intra kódování bez predikce, jestliže je velikost GOP rovna 1. Další úvahy pak probíhaly odděleně nad klíčovými a nad predikovanými snímky.

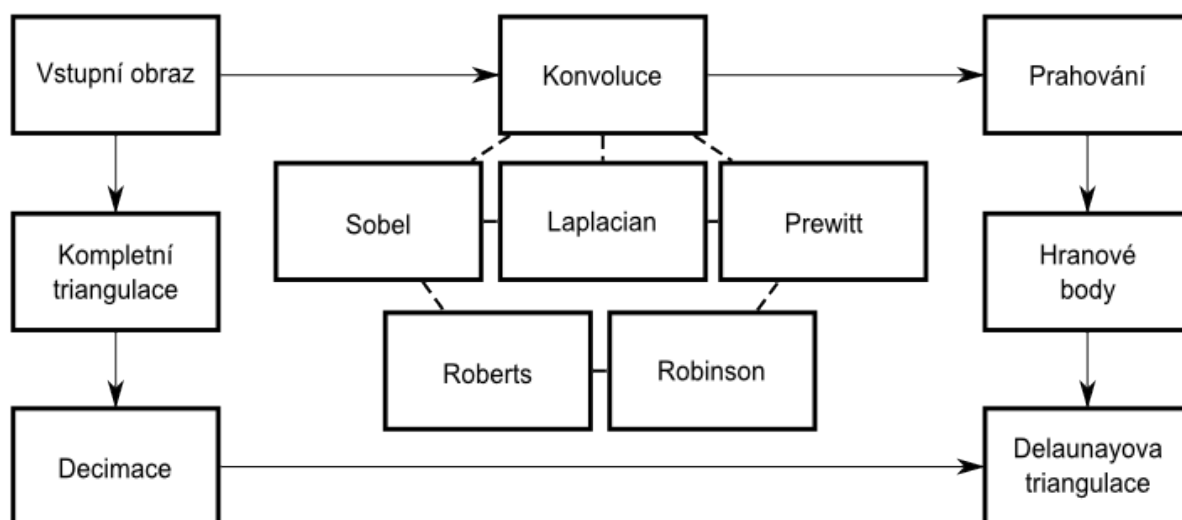


Obr. 7.3: Dvojice sousedních GOP s velikostí 6 navržené metody; Intra kódovaný I-snímek je následován pěticí dopředně inter kódovaných P-snímku

7.1.1 Intra kódování

Při intra kódování snímků bylo nutné zvážit, zda preferovat rychlou, ale nepřesnou hranovou detekci (viz kap. 5.2) anebo využít pomalou, ale zato poměrně přesnou adaptivní triangulaci (viz kap. 5.3). Vzhledem k této nejednoznačnosti došlo k rozvětvení návrhu už v samém začátku (viz obr. 7.4). Výpočtu intra kódovaného snímku pomocí adaptivní triangulace dosahujeme decimací kompletní triangulace, která má jako vzorky ve vrcholech všechny pixely obrazu, algoritmem zapůjčeným od autora metody [Koho07]. Tato metoda zatím nebyla vyzkoušena na snímcích z videa, které jsou ve většině případů neostré a tím se zásadně liší od snímků statických. Maximální datový tok kontrolujeme pouhým požadavkem na počet ponechaných bodů v takové triangulaci.

Kapitola 7 | Návrh vlastní metody



Obr. 7.4: Schema intra kódování

Hranová detekce

Pro účely hranové detekce byla implementována pětice hranových detektorů. U těch, které se zaměřují na hledání maxim prvních derivací, byla zvolena pouze dvojice masek pro směry obou souřadnicových os. Vyjimkou je Laplaceův operátor, který aproximuje průchod druhých derivací nulou a jeho maska je invariantní k otočení. V jeho případě proto stačí pouze jedna.

- Sobelův $h_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}; h_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

- Laplaceův $h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

- Robertsův $h_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}; h_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$

- Robinsonův $h_x = \begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}; h_y = \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$

- Prewittové $h_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}; h_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$

Všechny tyto metody produkují pro různé případy poměrně odlišné výsledky, přičemž optimální hranovou detekci je nutné zjistit v testech (viz kap. 9.2). Výsledek, kterého chceme

Kapitola 7 | Návrh vlastní metody

takovým postupem dosáhnout, je pokrytí hrany a jejího bezprostředního okolí. Chceme-li totiž věrně aproximovat hrany za pomoci trojúhelníků, musíme jejich vrcholy rozmístit v okolí hran tak, abychom zamezili distribuci intenzity hrany do okolí a přitom zároveň hrany pokryli body natolik hustě, aby si svou intenzitu zachovaly.

Výstupem hranového operátoru je obecný šedotónový snímek, proto musí nastoupit prahování s automatickým prahem, které jednoznačně určí, jaký interval intenzit budeme považovat za hrany. Problém automatického prahu spočívá v tom, že dopředu nevíme, která intenzita z intervalu $\langle 0; 255 \rangle$ bude jako prahová hodnota produkovat počet hranových bodů přibližně tentýž, který požadujeme. Obraz proto nejprve prahujeme s maximální intenzitou jako prahem, kde se dá předpokládat, že hranových bodů bude nejméně. Následně zkontrolujeme, zda se zde nachází více nebo méně bodů než požadujeme. Pakliže se zde nachází bodů více, nemá cenu snižovat práh, neboť prahováním s nižším prahem budou hranové body pouze přibývat a prahování v tomto případě ukončíme. Ve většině případů bude ale při prahování s prahem 255 bodů méně než (rozumný) požadavek. Poté za opakovaného snižování prahu s dekrementem 1 vždy po novém prahování vyhodnocujeme stejnou podmínku jako pro maximální intenzitu.

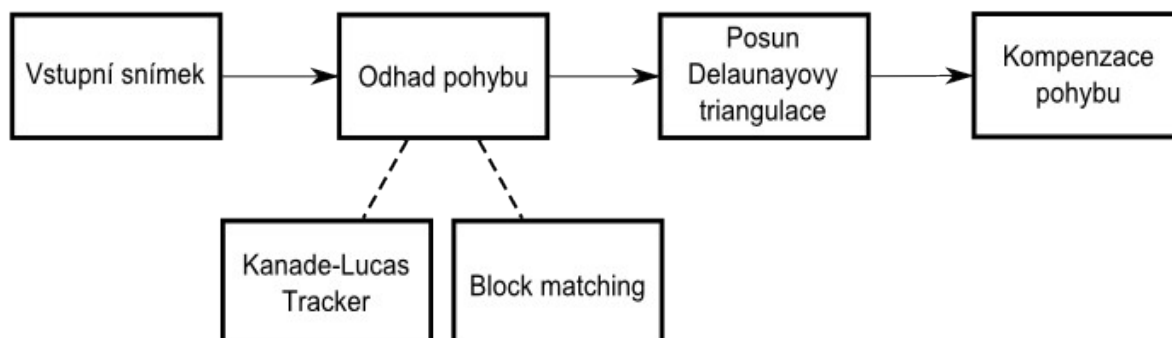
Po prahování snímku obsahujícího hrany se do výpočtu intra kódovaného snímku vměšuje kontrola datového toku. Z určitého počtu hranových bodů, které jsme získali po prahování, vezmeme do DT pouze tolik bodů, kolik jsme si před prahováním určili, neboť počty hranových bodů získaných prahováním s určitým prahem nejsou pro různé snímky konstantní. Zamezíme tím možnému kolísání počtu bodů, které by se projevilo velmi nestálým datovým tokem. Kdybychom ale tuto podmnožinu bodů vybírali sekvenčně, mohlo by se snadno stát, že by došlo k nežádoucímu aliasingu, proto uplatníme náhodný výběr. Zbytek bodů nám volitelně může doplnit množina bodů náhodně generovaných v místech, kde se ještě hranové body nenacházejí. Prostřednictvím požadavků na pevný počet hranových a takto vybraných náhodných bodů shora omezíme datový tok, který je typicky největší právě v intra kódovaných snímcích. V predikovaných snímcích se podle předpokladu budeme držet nízko pod touto hodnotou. O přesném vyjádření datového toku pojednává kap. 9.2.2.

7.1.2 Inter kódování

Od použití inter kódování na triangulaci reprezentované snímky si v první řadě slibujeme podstatné snížení datového toku. Prostředek pro další snižování je potřeba hledat v mezisnímkové korespondenci. Příkladem je případ, kdy je s minimální odchylkou v každém ze dvou konsekventních snímků $\mathfrak{I}(x, y, t)$ a $\mathfrak{I}(x, y, t+1)$ nalezen ten samý bod (včetně okolí). Tento bod je uložen jako vzorek v triangulaci. Nabízí se, aby u něj ve snímku pro $t+1$ nebyla zbytečně znovu kódována intenzita, ale pouze vektor posunutí. V nejhorším případě by pak byl v predikovaných snímcích o třetinu menší datový tok, neboť celá jedna souřadnice není potřeba. Další přínos může nastat ve chvíli, kdy je scéna převážně statická. Jsou-li nalezeny vektory posunutí, pak většina z nich jsou nulové vektory a ty jsou ve větším množství dobře kódovatelné, ať už pomocí delta kódování, deflate, popř. kombinace obou.

Kapitola 7 | Návrh vlastní metody

Obr. 7.5 zobrazuje postup při zpracování predikovaného snímku od odhadu pohybu po jeho kompenzaci. Zde je nutno poznamenat, že k vyšetřování odhadu pohybu potřebujeme rovněž předchozí vstupní snímek.

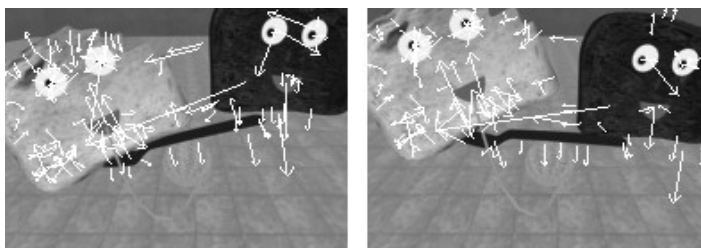


Obr. 7.5: Schema inter kódování

Protože predikované snímky mohou být odvozovány i z předchozích predikovaných snímků, je nutné v těchto snímcích triangulaci posouvat a při tom ji udržovat delaunayovskou. Zaměříme-li se na způsob, jakým zjistíme, že k pohybu došlo, pak máme k dispozici diferenciální a blokové metody pro odhad pohybu.

Výběr vhodných algoritmů pro odhad pohybu

V souvislosti s odhadem pohybu bylo nejprve vyšetřováno použití algoritmu SIFT (viz kap. 3.3). Tento algoritmus sám o sobě v intra kódovaných snímcích vybírá relevantní body, které jsou pak identifikovány i v následných snímcích. Lze tedy určit odhad jejich mezisnímkového pohybu. Z důvodu zjištění vhodnosti nasazení pro účely hledání korespondencí byly vyhotoveny snímky demonstrující průchod první fáze algoritmu nad testovacím videem (viz obr. 7.6). Z nich je patrné, že relevantní vektory mající za počátek relevantní body vycházejí výhradně z oblastí blízkých hranám, ale o hranách samotných žádná informace do dalšího zpracování nejde. Hlubší analýza algoritmu ukázala, že hranové body jsou z podstaty metody nezpracovatelné, protože metoda produkuje špatné výsledky pro invariantní oblasti. Je nutno poznamenat, že vzhledem k tomuto nekompromisnímu přístupu nelze algoritmu SIFT žádné body vnutit, proto by při zpracovávání obecných bodů z triangulace neobstál.



Obr. 7.6.: SIFT vektory ve dvojici konsekventních snímků

Kapitola 7 | Návrh vlastní metody

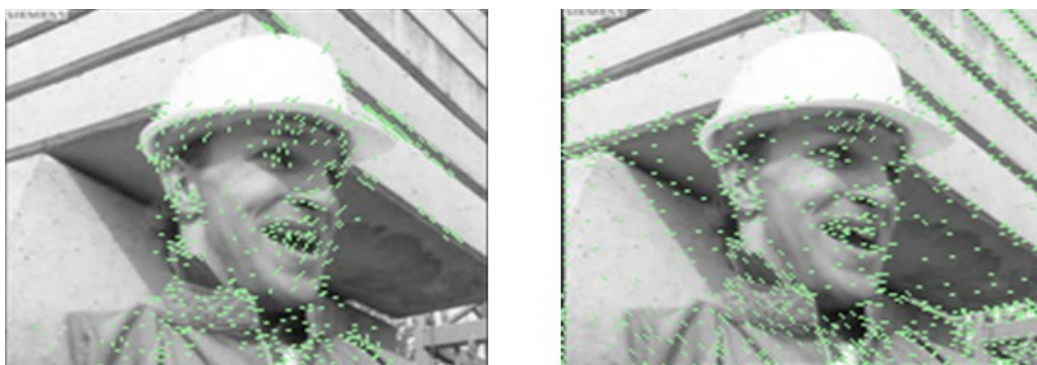
Zvažováno bylo rovněž použití algoritmu Block matching, který je popsán v kapitole 3.1. Ačkoli je tato bloková metoda užívána pro zjištění pohybu makrobloků v kodecích rodiny MPEG (viz kap. 2.6.1), lze ji stejně dobře použít i na oblasti centrované v obecných vrcholech triangulace, jak dokazují metody z kap. 6.1. Její výhodou je snadnost implementace, nevýhodou pak časová složitost a poměrná nepřesnost, neboť pohybové vektory jsou počítány pouze lokálně.

Na obr. 7.7 je vyhotovena demonstrace řešení založeného na této metodě. Ve středech čtverců o ploše 4x4px jsou umístěny vrcholy triangulace. Jejich velikost i zachycený výřez obrazu v nich odpovídá blokům, jejichž korespondence jsou metodou sledovány. V sekvenci dochází k pohybu v pravém horním rohu (viz obr. A.2; snímky 11 a 12). Během tří snímků došlo k přemístění většiny bloků na hraně objektu na odpovídající místo ve scéně.



Obr. 7.7: Test odhadu pohybu algoritmem block matching

Naproti tomu nasazení algoritmu KLT (viz kap. 3.2), u kterého se naskytla možnost provést výpočty nad daty z triangulace, poskytuje poněkud horší aproximaci optického toku v obecných bodech obrazu. V místech, kde jsou plochy téměř homogenní, algoritmus nachází pohyb i v případě, že zde k žádnému pohybu nedochází (viz obr. 7.8). Výsledkem je, že se body v celé scéně až na pár výjimek pohybují po ploše určitým směrem, který nezohledňuje skutečný pohyb.

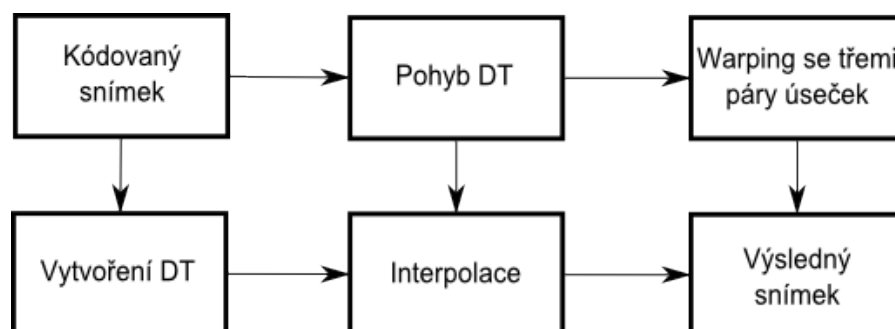


Obr. 7.8: Block matching vektory (vlevo) a KLT vektory (vpravo) pro 5% vložených bodů

Pro testování se ze tří uvedených algoritmů nejvíce hodí algoritmus block matching, neboť vektory, které produkuje pro obecné body, mají nejbližší k aproximaci optického toku ve dvojici obrazů.

7.2 Dekodér

Hlavní funkcí dekodéru je vytvořit z kódované reprezentace snímků bitmapové obrazy a ty zobrazit. Vzniklo několik možných způsobů, jakými dekodování provést (viz obr. 7.9).



Obr. 7.9: Schema dekodéru

Pokud se jedná o intra kódovaný snímek, je vše bez větších problémů. Na základě geometrických dat a informací o intenzitách v bodech je algoritmem Pohyblivé triangulace (viz kap. 4.1.1) zkonstruována DT. Ta je následně po jednotlivých trojúhelníkových algoritmem řádkového plnění v rastru interpolována za pomoci barycentrických souřadnic (viz kap. 4.2.2) anebo soustavy rovnic (viz kap. 4.2.3). Bilineární interpolace (viz kap. 4.2.1) nebyla vzhledem ke složitější implementaci dále zvažována.

Při inter kódovaném snímku na vstupu ale musí dojít k úpravě triangulace z předchozího inter nebo intra kódovaného snímku. Touto úpravou je myšlen pohyb, při kterém může dojít ke změně topologie v DT. Vzhledem k tomu, že touto transformací už DT jednou prošla během kódování, není se potřeba obávat, že by došlo k neočekávaným výsledkům. Vykreslení celé plochy reprezentované triangulací ze vzorků může probíhat buď interpolací jako v případě intra kódování anebo může být speciálně v tomto případě použit warping se třemi páry úseček (viz kap. 4.2.4). Zde se situace mírně komplikuje, neboť warping není založen na interpolaci vzorků z vrcholů DT. Navržené schéma proto předpokládá, že je ve videosekvenci k dispozici již zpracovaný intra kódovaný snímek, který vyšetřovanému inter kódovanému snímku předchází. Pakliže máme uloženy pozice bodů, na kterých se tyto nacházely ve zmíněném intra snímku, máme k dispozici zároveň i počáteční transformaci trojúhelníků, které chceme v inter snímku warpovat. Odpovídající tři páry úseček rozdělené mezi intra a inter snímek tedy vstupují jako parametry do warpingu.

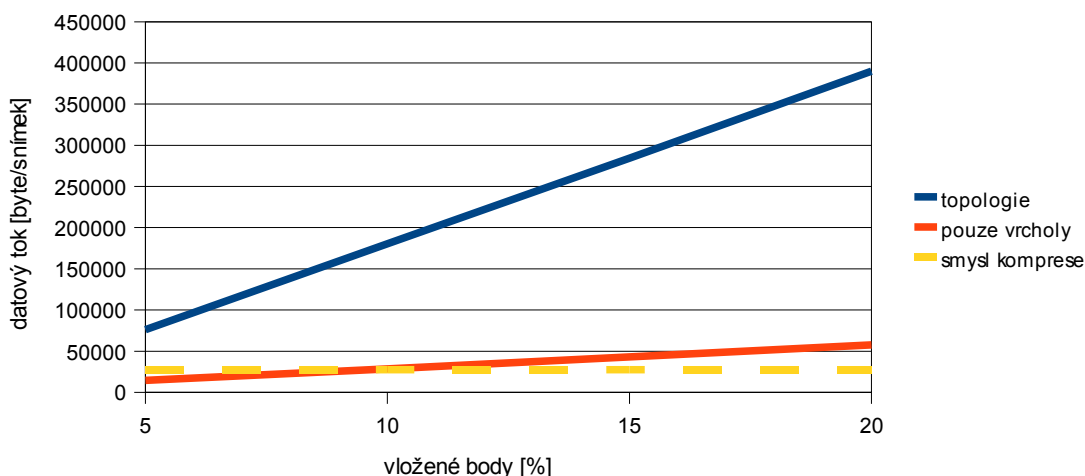
7.3 Slovní srovnání návrhu s existujícími metodami

Ve smyslu intra kódovaných snímků se navrhovaná metoda nejvíce blíží metodám pro statický obraz [Pras06] a [Garc99] popsaným v kap. 5.2. Na rozdíl od nich ale po detekci hran neaplikuje konvergenční metodu pro snížení chyby, ani nezachovává hrany pomocí CDT. Inter kódování je realizováno kombinací odhadu pohybu blokovými metodami podobně jako v [Yaop98] a [Srik05] z kap. 6.1 spolu s kompenzací pohybu warpingem pro celý snímek podobně, jako [ISO01] z kap. 6.3 zpracovává jen některé objekty ve videu.

7.4 Bezeztrátová komprese triangulace

Smyslem této kapitoly není navrhnout novou optimální bezeztrátovou kompresi triangulace, ale v první řadě provést rozbor paměťových nároků výstupních dat. Na tyto data jsou poté aplikovány intuitivně vhodné kompresní algoritmy, které paměťové nároky sníží tak, aby výstupní data byla uložena přinejmenším stejně hospodárně jako data původní. Pro dosažení lepších kompresních poměrů při kódování triangulace viz [Koho07].

Triangulace reprezentující obraz v intra kódovaném snímku je poměrně objemná struktura čítající množinu pozic vrcholů, indexy vrcholů jednotlivých trojúhelníků a volitelně i sousednost. Navíc je s každým z vrcholů asociována intenzita, která slouží k rekonstrukci původního obrazu. Pokud by byla kódována veškerá zmíněná informace, nebylo by možné dosáhnout kompresního poměru 1:1 ani při mizivém počtu vložených bodů z původního obrazu. K potvrzení této domněnky byla porovnávána velikost původního snímku reprezentovaného bitmapou s velikostí triangulace.



V prvním pokusu je triangulace uložena v textovém souboru rozděleném do tří sekcí. V první je na každé řádce tabulátory oddělená trojice $\{x_i, y_i, e_i\}$, jejímiž prvky jsou nezáporná celá čísla označující pozici a intenzitu i -tého vrcholu. V druhé sekci jsou na každé řádce indexy do seznamu bodů a ve třetí jsou indexy do seznamu (sousedních) trojúhelníků. Výsledná reprezentace pro původní snímek o velikosti 27648 bajtů pro 5-20% vložených bodů je vynesena v grafu 7.1. Větší procento vložených bodů postrádá smysl ve chvíli, kdy je překročena velikost původních bitmapových snímků, která je v grafu naznačena čárkovanou čarou. Vzhledem k tomu, že v této práci je prostředkem pro reprezentaci Delaunayova triangulace, je možné celou topologii zanedbat a vypsát pouze pozice bodů s intenzitami. Důvodem je fakt, že DT je pro danou množinu bodů unikátní (viz kap. 4.1). Nutno ale poznamenat, že tyto body se v našem případě nacházejí v pravidelné mřížce, kde je větší riziko vzniku singularit. Unikátnost lze opětovně jednoduše zaručit přičítáním náhodných subpixelových odchylek ke každé souřadnici bezeztrátově kódovaných bodů. Při kódování

Kapitola 7 | Návrh vlastní metody

postaveném na zanedbání topologie byla v souboru ponechána pouze první sekce. Výsledek je vyneseno rovněž v grafu 7.1. Na první pohled je vidět, že reálně použít první případ nelze ani pro 5% vložených bodů. Druhý případ pak má smysl pouze pro méně jak 10% bodů. V obou případech je tato hodnota však příliš nízká, než aby bylo dosaženo rozumné hodnoty PSNR.

Nyní je nutno zvážit, zda zápis do textového souboru představuje nejlepší možnou volbu. Velká míra redundance je zjevně způsobena oddělováním jednotlivých souřadnic tabulátory a odřádkováním. Je nasnadě zavést ve výpisu pevnou délku jednoho záznamu a tyto pak explicitně neoddělovat. Na základě této myšlenky byl určen způsob zápisu bodu v pěti bajtech a výsledný formát byl pojmenován RAW. Každý bod je tedy určen uspořádanou pěticí

$$\begin{aligned} p_1 &\equiv \langle x_{a1}, x_{b1}, x_{a1}, x_{b1}, e_1 \rangle \\ p_2 &\equiv \langle x_{a2}, x_{b2}, x_{a2}, x_{b2}, e_2 \rangle \\ &\dots \\ p_i &\equiv \langle x_{ai}, x_{bi}, x_{ai}, x_{bi}, e_i \rangle \\ &\dots \\ p_n &\equiv \langle x_{an}, x_{bn}, x_{an}, x_{bn}, e_n \rangle \end{aligned}$$

, kde první dva byty x_{ai}, x_{bi} představují x -souřadnici i -tého bodu, druhé dva y_{ai}, y_{bi} y -souřadnici a pátý e_i je ponechán pro intenzitu. Velikost souboru v bajtech v závislosti na počtu uložených bodů je tedy jejich pětinásobkem.

Další varianta jak docílit ještě většího kompresního poměru spočívá v zakomprimování archivátorem gzip, který je standardní součástí .NET. Jedná se o archivátor založený na algoritmu Deflate [RFC], který je spojením bezztrátového kódování LZ77 a následného Huffmanova kódování. Způsob kódování byl označen jako RAW+GZIP.

Při úvahách nad efektivnějším kódováním pětičky byl navržen lepší způsob zápisu, který se inspiroje delta kódováním. Vychází z předpokladů, že seznam bodů nemusí být uspořádaný a že histogram intenzit ze vzorků v bodech triangulace obsahuje několik dominantních intenzit. Vzhledem k těmto předpokladům je možno pětičky vzestupně setřídit podle e_i a pro $i > 1$ zapisovat namísto hodnoty intenzity pouze rozdíl oproti intenzitě v předchozí pětičce. Data pak budou archivátorem gzip komprimována lépe, neboť pátý prvek bude ve většině případů nula. Způsob kódování byl označen jako DELTA+GZIP a má následující formát

$$\begin{aligned} p_1 &\equiv \langle x_{a1}, x_{b1}, x_{a1}, x_{b1}, e_1 \rangle \\ p_2 &\equiv \langle x_{a2}, x_{b2}, x_{a2}, x_{b2}, e_2 - e_1 \rangle \\ &\dots \\ p_i &\equiv \langle x_{ai}, x_{bi}, x_{ai}, x_{bi}, e_i - e_{i-1} \rangle \\ &\dots \\ p_n &\equiv \langle x_{an}, x_{bn}, x_{an}, x_{bn}, e_n - e_{n-1} \rangle \end{aligned}$$

Kapitola 7 | Návrh vlastní metody

Pro inter kódované snímky reprezentované uspořádanými dvojicemi $v_i \equiv \langle \delta x_i, \delta y_i \rangle$, kde δx_i a δy_i jsou složky vektoru posunutí i -tého bodu, ale delta kódování ani řazení použít nelze. Za prvé, seznam bodů musí být uspořádaný, neboť vektory jsou v souboru pozičně spjaty s odpovídajícími body v intra kódovaném snímku. Za druhé, delta kódování bez seřazení nemá význam. Nicméně další, ještě úspornější, zápis n -tice lze založit na poznatku, že maximální velikost každé ze složek vektorů posunutí je omezena velikostí vyhledávacího okénka metody pro odhad pohybu. To je voleno v řádu jednotek až desítek pixelů. Pro každou ze složek tedy stačí jeden byte a velikost dvojic i výsledného souboru je tím snížena o polovinu oproti intuitivním dvoubajtovým složkám. Toto kódování budeme zvat jako v případě jeho obdoby u intra kódování RAW+GZIP.

Navržená kódování RAW, RAW+GZIP a DELTA+GZIP pro intra kódované snímky jsou podrobena testování v kap. 9.2.2, zatímco test navrženého kódování RAW+GZIP pro inter kódované snímky lze najít v kap. 9.3.2.

8 Implementace

Aplikace byla vytvořena v jazyce C# v prostředí SharpDevelop. Jazyk C# byl vybrán proto, že nabízí v současné době nejlepší kompromis mezi syntaxí strukturovaných jazyků (Pascal, C) a mezi zapouzdřením a podporou .NET v moderních jazycích založených na OOP (Java, Visual Basic). Rovněž důležitým argumentem, proč byl vybrán právě jazyk C# ve spojení se SharpDevelop, je na rozdíl od MSVS 2005 opravdu snadná tvorba uživatelských rozhraní. Dále skutečnost, že toto vývojové prostředí je vyvíjeno pod licencí LGPL, umožňuje v případě potřeby nahlédnout do zdrojového kódu.

8.1 Přístup ke vstupním datům

Protože většinu prováděných operací, které se podílejí na výsledcích, představují operace se statickým obrazem v podobě matice pixelů, bylo potřeba vzhledem k úrovni programovacího jazyka obejít nedostupnost pointerové aritmetiky. K tomuto účelu posloužila deklarace *unsafe*, která dovoluje pracovat s pointerem v C# na úrovni jeho předchůdců ANSI C a C++. Jediné, o co takovým způsobem přijdeme, je bezpečnost. Nicméně vzhledem ke skutečnosti, že matice obrazu jako struktury typu pole mají pro celé videosekvence vždy konstantní rozměr, a my pointery používáme pouze pro lokalizaci pixelu v nich, je prakticky nemožné, abychom daný rozměr nedopatřením překročili. Pozitivum používání pointerů v C# lze ilustrovat na příkladu kódu 8.1 z implementované metody, který běží zvláště pro velké rozměry obrazu několikrát rychleji než hypotetický kód 8.2, ačkoliv vykonávaná úloha¹³ v obou je shodná.

```
public Bitmap clear_bitmap(Bitmap b, byte i) {
    for (int y = 0; y < height; y++)
        for (int x = 0; x < width; x++)
            b.SetPixel(x, y, Color.FromArgb(i, i, i));
    return b;
}
```

Kód 8.1: Naivní přístup k průchodu maticí obrazu

```
public unsafe Bitmap clear_bitmap(Bitmap b, byte i, Rectangle rect,
PixelFormat p_f) {
    int y_part, offset;
    BitmapData data = b.LockBits(rect, ImageLockMode.ReadOnly, p_f);
    byte* ptr = (byte*)(data.Scan0);
    for (int y = 0; y < height; y++) {
        y_part = y * data.Stride;
        for (int x = 0; x < width; x++) {
            offset = y_part + x * bpp;
            ptr[offset] = ptr[offset + 1] = ptr[offset + 2] = i;
        }
    }
    b.UnlockBits(data);
    return b;
}
```

Kód 8.2: Pointerový přístup k průchodu maticí obrazu

¹³ vyplnění všech barevných složek všech pixelů matice obrazu zadanou intenzitou

8.2 Datové struktury

Pro konstrukci, ukládání a pohyb triangulací aplikace používá Pohyblivou triangulaci T. Vomáčky [Voma08], která je k dispozici prostřednictvím dll knihovny. Jejými základními datovými typy jsou:

- **Trojúhelník (Triangle)**: Obsahuje odkazy na tři body a tři hrany, které jej tvoří.
- **Hrana (Edge)**: Obsahuje odkazy na dva body.
- **Bod (Point2D)**: Atributy jsou pozice x , y typu `double` a odkaz na vektor posunutí.
- **Vektor posunutí (Vector2D)**: Atributy jsou posunutí x , y typu `double`.

Protože získání dat z triangulace je řešeno vždy předáním celého seznamu instancí výše uvedených typů, a je tedy časově velmi náročné, bylo nutné vytvořit další struktury uvnitř vlastní aplikace takové, že obsahují duplicitně informaci o poloze bodů. Mimo jiné jsou zde uloženy také údaje o intenzitách vzorků, protože zaměření Pohyblivé triangulace nesouvisí primárně s reprezentací obrazu a videa. Kompletní seznam použitých typů je následující:

- **Seznam bodů**: Obsahuje instance bodů, které jsou již v triangulaci.
- **Tabulka intenzit**: Asociuje body z triangulace s intenzitami.
- **Tabulka pixelů**: Asociuje indexy pixelů s body v triangulaci.
- **Tabulka poloh 1**: Asociuje instance bodů s jejich pozicí v předchozím snímku.
- **Tabulka poloh 2**: Asociuje instance bodů s jejich pozicí v předchozím intra snímku.

8.3 Detaily implementace

Převzatými metodami do implementace je kromě Pohyblivé triangulace ještě KLT tracker z knihovny OpenCV [OpenCV], zpřístupněný pro jazyk C# wrapperem SharperCV [SharCV]. Dále je použita adaptivní triangulace [Koho07], která je používána jako externí program, přičemž je pro každý snímek spouštěna jako na aplikaci nezávislý proces.

Mezi původní metody, které byly v aplikaci s úspěchem implementovány, patří konvoluce s hranovými operátory (kap. 5.2) a automatické prahování, block matching (kap. 3.1), Bresenhamův algoritmus (kap. 4.2), interpolace za pomoci barycentrických souřadnic (kap. 4.2.2) a soustavy rovnic (kap. 4.2.3) a posléze i warping se třemi páry úseček (kap. 4.2.4).

9 Testy a srovnání s dostupnými metodami

Pro účely srovnání byla zvolena tři videa lišící se v povaze zobrazované scény a způsobu pohybu v ní. Vzhledem k vysoké časové složitosti je jejich délka omezena na 100 snímků při 25fps. Jejich rozlišení je zhruba poloviční oproti standardu VCD. Barevný formát byl volen RGB24 spíše než Y8, protože dekompressor nutný k přehrávání Y8 není běžně dostupný.

Video s pohyblivou kamerou je zkrácené „Foreman“ převzaté z [Xiph] (zdroj volně šířitelných videosekvencí), zmenšené při zachování poměru na rozlišení 176x144 při šedotónových snímcích. V některých snímcích je přítomno rozmazání vlivem pohybu. Mezi začátkem a koncem se postupně kompletně změní scéna (viz obr. A.1).

Renderované video „Toasts“ pochází z vlastní výroby. Jeho rozlišení je 192x144 a co do intenzity je 8-bit šedotónové. Podstatnou vlastností je úplná absence rozmazání pohybem a neexistence šumu. Vzniká tak vysoký kontrast, při kterém lze snadno detekovat pohyb jednotlivých objektů. Dále je téměř polovina scény netexturovaná, tedy místy lze narazit vzhledem k intenzitě na téměř homogenní plochy. Vzorok snímku z celé videosekvence jsou k nahlédnutí na obr. A.2.

Video se statickou kamerou je zkrácené „Miss America“ převzaté z [Xiph], zmenšené při zachování poměru na rozlišení 176x144 a převedeno na šedotónové. Jedná se o typické video typu „talking head“. Jeho zvláštností je téměř homogenní pozadí a minimální pohyb osoby v popředí (viz obr. A.3).

Všechny testy probíhaly na sestavě s procesorem AMD Athlon 3200+ s nastavenou frekvencí 2000MHz. Byla užitá paměť 1,5GB DDR2 SDRAM s frekvencí 800MHz a latencí 5-5-5-15.

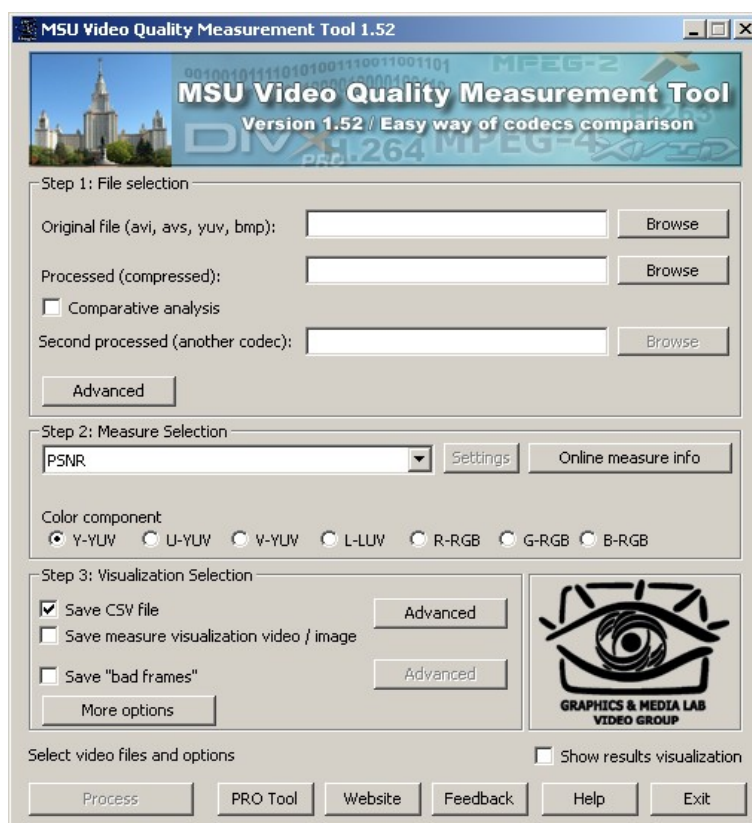
9.1 Systém testování

Porovnání původního videa s videem vytvořeným kóděrem probíhá na základě vyhodnocení metriky PSNR, která byla implementována podle vzorce 9.1, kde konstanta 255 vyjadřuje maximální hodnotu intenzity. MSE je střední kvadratická chyba. \mathfrak{S}' je zakódované video a \mathfrak{S} je video původní. $M \times N$ představuje plochu snímku, i, j jsou souřadnice uvnitř snímku a t je čas snímku určující zařazení ve videosekvenci. V následujících kapitolách se při testování můžeme setkat s vyjádřením závislosti PSNR na procentu vložených bodů. Procenta jsou volena proto, abychom se vyvarovali závislosti na velikosti snímku. Procentní interval, pro který PSNR vyjadřujeme, se v části zaměřené na intra kódování odvíjí od použitelnosti reprezentace, kdežto v části zaměřené na inter kódování jde o pevné procento, pro které bylo vzhledem k vysoké časové složitosti testování reálné.

$$MSE = \frac{1}{M \cdot N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \|\mathfrak{S}(i, j, t) - \mathfrak{S}'(i, j, t)\|^2$$
$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \quad (9.1)$$

Kapitola 9 | Testy a srovnání s dostupnými metodami

Pro porovnávání navržené metody s jinými metodami byl zpočátku používán volně šiřitelný program MSU Video Quality Measurement Tool [MSU], který je schopen vyhodnotit odlišnost původního a zakódovaného videa na bázi běžně používaných metrik PSNR, SSIM, VQM, Delta, MSAD, MSE a k tomu přidává pět vlastních, jako například vyhodnocení blokovosti nebo rozmazání. Zajímavý je rovněž mód, ve kterém lze provést komparativní analýzu dvou zakódovaných videosekvencí vzhledem k původní. Porovnávat lze podle libovolného počtu složek videa s formátem barev RGB nebo YUV. Výsledky program poskytuje formou spojnicového grafu, jako video s barevným vyznačením chyby anebo lze vypsat hodnotu chyby pro každý snímek do souboru formátu CSV. Náhled aplikace je na obr. 9.1. Veškerou její potřebnou funkcionalitu ale během testování nahradila vlastní aplikace.



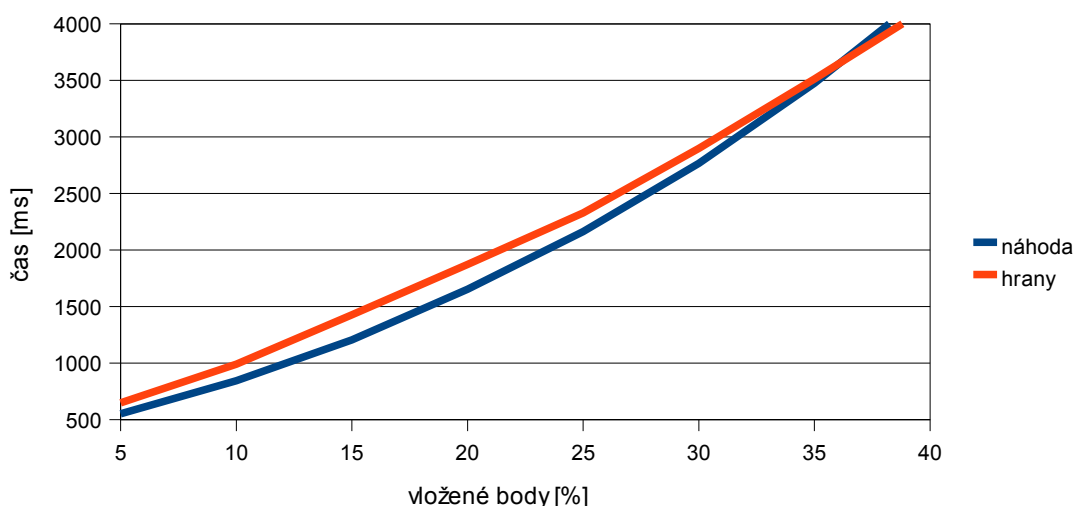
Obr. 9.1.: MSU Video Quality Measurement Tool [MSU]

9.2 Intra kódování

Tato kapitola je zaměřena na případ, kdy je videosekvence reprezentována posloupností absolutně kódovaných statických snímků. Tedy žádný odhad ani kompenzace pohybu se pro výpočet snímků nepoužívá. V první části se zaměříme na časovou složitost výsledného řešení a jednotlivé její složky. Ve druhé části zjistíme, jaký datový tok a kompresní poměr můžeme od intra kódování čekat. V části třetí vyzkoušíme různé hranové detektory popsané v kap. 7.1.1 a pokusíme se najít ten, co se nejlépe hodí pro konkrétní data. Zároveň se pokusíme vyvodit vhodnost celé metody pro různé typy vstupních dat. V poslední části porovnáme výsledky s decimální ADT a s kodekem M-JPEG, který stejně jako vlastní metoda používá pouze intra kódování.

9.2.1 Časová složitost

Časová složitost odpovídající rychlosti komprese intra kódovaných snímků závisí pouze na velikosti vstupních dat, respektive na procentu takových bodů, které je z nich vkládáno. Z tohoto důvodu byla pouze pro jedno video (Foreman) srovnávána časová složitost náhodného výběru s průměrem časových složitostí hranové detekce, které se ukázaly být velice podobné. Čas pro každé procento vstupních bodů se v prvním případě sestává z generování náhodných souřadnic, vložení do triangulace, interpolace a bezeztrátového kódování. V případě hranové detekce jde o čas samotné hranové detekce, automatického prahování, vložení do triangulace, interpolace a bezeztrátového kódování. Interpolace je součástí časové složitosti intra kódování proto, že její výsledek může být využit jako referenční obraz pro pohybovou kompenzaci predikovaných snímků za pomoci warpingu. Hranová detekce¹⁴, jak je vidět na grafu 9.1, přispívá k časové složitosti oproti čistě náhodnému výběru 5-30% bodů mírným zpomalením. Nad 30% vložených bodů se pak tento rozdíl stírá, neboť časová složka, která je zodpovědná za zpomalení se s počtem bodů téměř nemění.



Graf 9.1: Časová složitost v závislosti na procentu vložených bodů z obrazu 176x144px

V tabulce 9.1 je možno vidět průměrný podíl jednotlivých výpočetních celků na výsledné časové složitosti snímků videosekvence při intra kódování s použitím hranové detekce. První sloupec z každého páru popisuje časovou složitost v milisekundách a druhý procento, které daný čas představuje v celkové časové složitosti. Vkládání hranových bodů je s přibližně 80% podílem časově nejnáročnější operace následovaná interpolací s přibližně 14% podílem. Konvoluce prováděná při hranové detekci následovaná automatickým prahováním oproti předpokladům v časové složitosti nefiguruje. Samotné bezeztrátové kódování je s 1% nepodstatná téměř konstantní složka celkové časové složitosti v řádu jednotek milisekund. Ohledně reakce na zvýšený počet bodů se z tabulky dá vyvodit, že čas vkládání roste ze všech složek nejstrměji. Čas detekce roste jen mírně, poněvadž automatické prahování musí pro dosažení většího počtu bodů vyzkoušet větší množství prahů. Čas interpolace roste s vkládanými body přibližně lineárně.

14 Jde o aritmetický průměr časových složitostí všech implementovaných hranových detektorů.

Kapitola 9 | Testy a srovnání s dostupnými metodami

V tabulkách 9.2 a 9.3 můžeme vidět zpracování videosekvencí Miss America a Toasts. Podstatnou odlišnost od poměrné časové složitosti videa Foreman (viz tab. 9.1) lze spatřit v detekci hran, konkrétně v automatickém prahování. Tyto dvě videosekvence totiž obsahují podstatně méně hran než Foreman a proto z podstaty automatického prahování je podniknuto více kroků na nalezení takového prahu, který by vyhovoval počtu požadovaných bodů. Ve všech ostatních aspektech je poměrná časová složitost pro všechna videa téměř shodná.

	detekce [ms]	detekce [%]	vkládání [ms]	vkládání [%]	interpolace [ms]	interpolace [%]	kódování [ms]	kódování [%]
5% bodů	34	5,26	502	77,59	106	16,38	5	0,77
10% bodů	73	7,37	740	74,67	160	16,15	18	1,82
15% bodů	99	6,94	1087	76,23	225	15,78	15	1,05
20% bodů	126	6,73	1440	76,96	271	14,48	34	1,82
25% bodů	138	5,93	1831	78,65	329	14,13	30	1,29
30% bodů	153	5,28	2311	79,74	395	13,63	39	1,35
35% bodů	163	4,64	2817	80,21	480	13,67	52	1,48
40% bodů	167	4,02	3442	82,88	503	12,11	41	0,99

	detekce [ms]	detekce [%]	vkládání [ms]	vkládání [%]	interpolace [ms]	interpolace [%]	kódování [ms]	kódování [%]
5% bodů	138	18,8	475	64,71	113	15,4	8	1,09
10% bodů	175	15,45	778	68,67	166	14,65	14	1,24
15% bodů	187	12,6	1046	70,49	225	15,16	26	1,75
20% bodů	200	10,38	1425	73,95	269	13,96	33	1,71
25% bodů	202	8,77	1762	76,48	310	13,45	30	1,3
30% bodů	208	7,21	2256	78,17	385	13,34	37	1,28
35% bodů	210	6,12	2738	79,78	425	12,38	59	1,72
40% bodů	214	5,18	3383	81,93	478	11,58	54	1,31

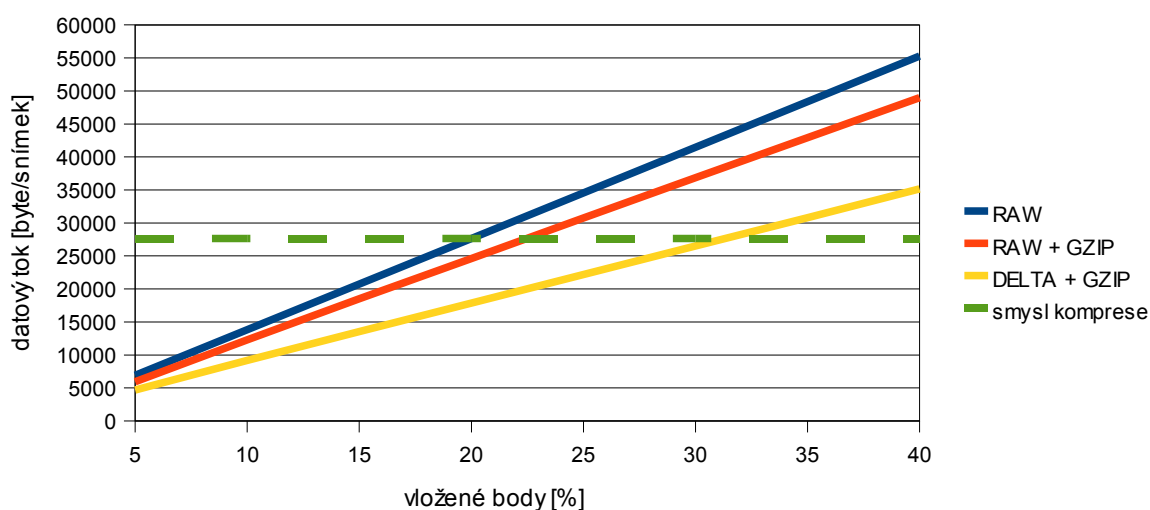
Tabulka 9.2: Poměrná časová složitost pro Miss America 176x144px; hranová detekce; 20% bodů

	detekce [ms]	detekce [%]	vkládání [ms]	vkládání [%]	interpolace [ms]	interpolace [%]	kódování [ms]	kódování [%]
5% bodů	93	11,6	587	73,19	117	14,59	5	0,62
10% bodů	154	12,4	888	71,5	184	14,81	16	1,29
15% bodů	191	10,65	1298	72,39	281	15,67	23	1,28
20% bodů	196	9,02	1635	75,28	313	14,41	28	1,29
25% bodů	202	7,39	2127	77,83	377	13,79	27	0,99
30% bodů	204	6,01	2684	79,1	451	13,29	54	1,59
35% bodů	209	5,09	3332	81,19	524	12,77	39	0,95
40% bodů	226	4,53	4148	83,18	569	11,41	44	0,88

Obdobná tabulka popisující náhodný výběr by nedokázala nic jiného než opět dominantní časovou složitost vkládání bodů. Neobsahovala by sloupce související s detekcí. Rozdíl mezi křivkami v grafu 9.1 je tedy dán částečně absencí detekce hran a částečně absencí automatického prahování u náhodného výběru obecných bodů. Hlavní podíl na uvedeném rozdílu ale má tedy vkládání bodů, resp. skutečnost, že při náhodném výběru hranových bodů je větší problém najít bod na hraně než při náhodném výběru obecných bodů, kdy je k dispozici celý snímek vyjma již vložených bodů.

9.2.2 Datový tok a kompresní poměr

S ohledem na úroveň komprese je dobré vědět, kolik bodů je možné do triangulace vložit, aniž by byla překročena velikost původních snímků. Zároveň je pro potřeby videa nutné omezit datový tok nějakou maximální hodnotou a té se ve všech snímcích držet. Pro video Toasts s velikostí originálního snímku 27648B bylo otestováno použití bezeztrátové komprese RAW, RAW+GZIP a DELTA+GZIP (viz kap. 7.4). Z výsledku v grafu 9.2 je vidět, že smysl komprese je pro RAW metodu kódování pod 20% vložených bodů, pro RAW+GZIP pod zhruba 24% a konečně pro DELTA+GZIP je možno jít až na 33% vložených bodů. Komprese je pro nejlepší kódování DELTA+GZIP v případě 5% vložených bodů přibližně 6:1 a pro v testech používaných 20% vložených bodů činí 1,55:1 s průměrnou velikostí jednoho snímku 16481B. Optimální poměr kvality a komprese se evidentně nachází mezi 15 a 20% vložených bodů.



Graf 9.2: Závislost velikosti snímku 192x144px na procentu vložených bodů

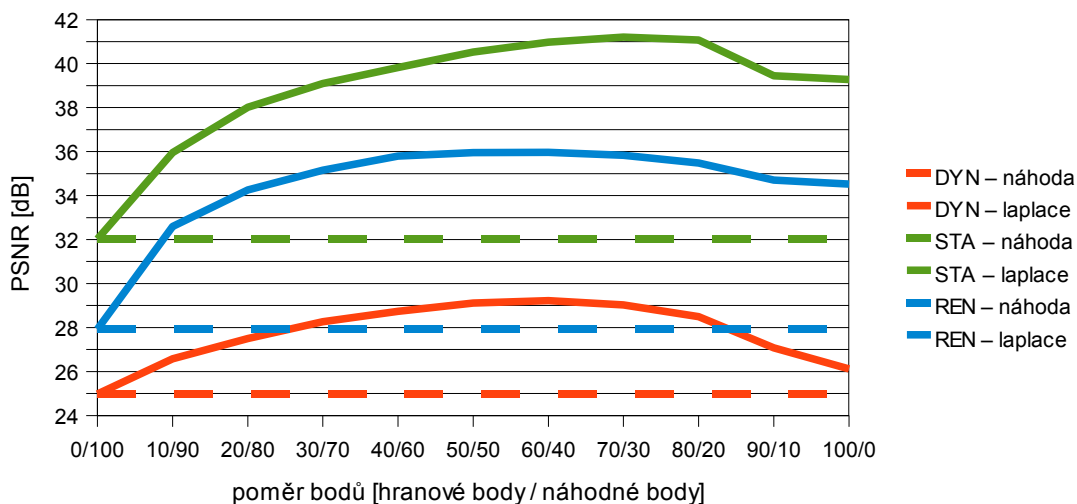
9.2.3 Výběr optimální hranové detekce

První sadou testů je srovnání hranových detektorů s cílem zjistit, jak moc je jejich použití oprávněné oproti výběru čistě náhodných bodů. Na základě výsledků je pak možno vybrat nejlepší detektor. Použity byly hranové detektory z kap. 7.1.1 pro 20% vložených bodů, neboť zhruba toto procento nabízí z hlediska kap. 9.2.2 vyvážený poměr kvality a komprese. V současnosti není kvůli primárnímu zaměření na pohyb DT použita CDT. Hranové ani náhodně vybrané body tedy samy o sobě nezaručí přilnutí triangulace k okrajům obrazu. Proto bylo potřeba přidat všechny body právě z okrajů, aby PSNR podstatně neklesala vlivem toho, že po vytvoření DT nad všemi body nedojde k pokrytí některých oblastí.

Je zřejmé, že výběr pouze náhodných anebo pouze hranových bodů povede k poklesu PSNR i subjektivního dojmu z videa. Proto předpokladem pro smysluplné testování hranových detektorů je zjištění, jaký poměr hranových a náhodných bodů je pro každou z testovacích videosekvencí optimální.

Kapitola 9 | Testy a srovnání s dostupnými metodami

Při pohledu na graf 9.3 můžeme spatřit tři křivky vykreslené plnou čarou, kde každá z nich náleží jedné testovací videosekvenci. DYN představuje videosekvenci Foreman (dynamická kamera), STA je Miss America (statická kamera) a REN značí videosekvenci Toasts (renderované video). Čárkované vodorovné úsečky v grafu znázorňují z důvodů porovnání se zmíněnými křivkami, jak vypadá čistě náhodný výběr stejného počtu bodů pro danou videosekvenci. Z celého obrazu bylo vybráno 20% bodů a ty pak byly dále rozděleny v poměru daném značením osy x na hranové a náhodně generované body.



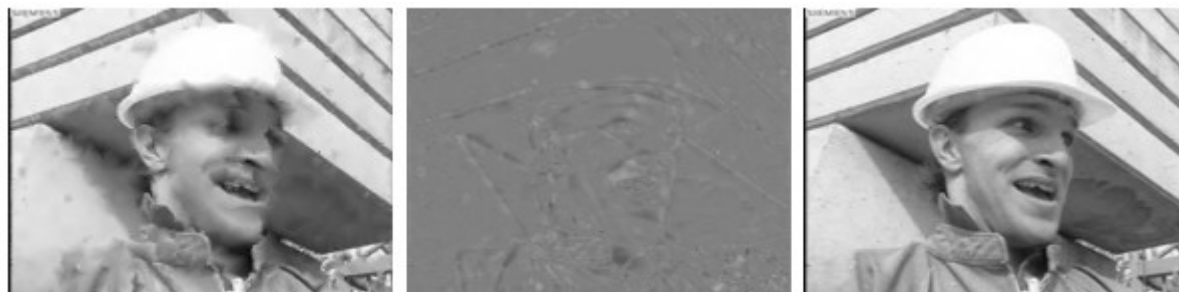
Průměrné hodnoty PSNR v grafu se pro jednotlivá videa vzájemně značně liší, což prozkoumáme podrobně hned v následujících testech. Nyní se zaměříme na polohy maxim, která udávají, pro jaký poměr je videosekvence aproximována nejlépe. V případě videa Foreman (DYN) má PSNR křivka maximum pro poměr hranových bodů k náhodným 60:40. Maximum pro video Toasts nastane při poměru 50:50. Pro video Miss America pak maximum můžeme najít až při poměru 80:20. V intervalu od nejnižšího poměru k optimálnímu dochází ve všech případech u PSNR k téměř logaritmické odezvě, zatímco na zbylém intervalu mírně klesá. Tento efekt je podle pozorování dán tím, v jakém poměru jsou ve videosekvenci plochy s vyšší hranovostí a plochy téměř homogenní. V případě, že jsme daným poměrem hranových bodů dosud nepokryli maximum relevantní informace v plochách prvního typu, křivka má stále, kam stoupat. Pokud je ale veškerá relevantní informace vyčerpána a my nemáme, odkud hranové body brát, prahujeme s nižším prahem. To nám sice vrátí více hranových bodů, ale ty budou obsahovat stále více šumu, který jednak není pro aproximaci tak relevantní jako skutečné hrany a jednak nemusí být rozptýlen po celé ploše obrazu tak jako body, které získáme náhodným výběrem.

Odstup od čistě náhodného výběru, který si Miss America oproti ostatním drží pro poměr 100:0 (náhodně generované body nejsou přítomny), je způsoben vysokou homogenitou pozadí. Ta je tak zjevná, že křivka má ke konci intervalu dokonce konstantní průběh.

Kapitola 9 | Testy a srovnání s dostupnými metodami

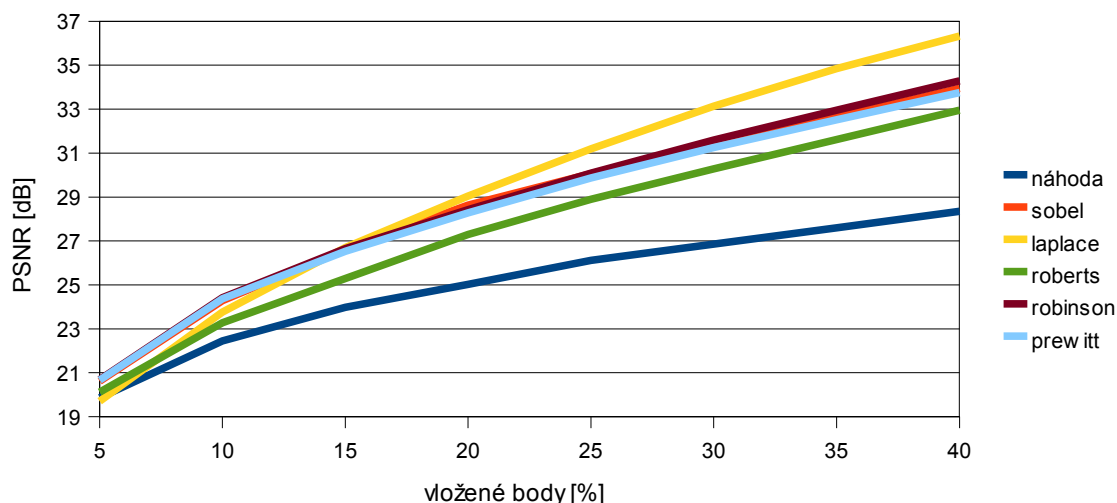
Pohyblivá kamera: „Foreman“

Video Foreman o velikosti 176x144px bylo podrobně testováno pro 20% vložených bodů. První snímek videosekvence včetně porovnání s odpovídajícím původním snímkem lze vidět na obr. 9.2. Ilustrace postupu od vytvoření triangulace až po vykreslení snímku je na obr. A.4.



Obr. 9.2: Zleva: aproximace vložení 20% bodů, residuum, originální snímek

Při testování kvality aproximace u videa typu pohyblivá kamera byl nejprve zkoumán interval $\langle 5\%;40\% \rangle$ vložených bodů při poměru hranových bodů k náhodným 50:50. Jak je vidět z grafu 9.4, závislost PSNR na množství vložených bodů je logaritmická. Všechny hranové detektory překonávají náhodný výběr počínaje 5% vložených bodů. Křivky by se sbíhaly až ve zhruba 2,5% vložených bodů, protože toto procento představuje vložené body z okrajů obrazu. Nejlepší hrany produkuje Laplaceův operátor, který při 40% vložených dosahuje téměř 37dB a rozdíl od čistě náhodného výběru je tak 8dB. Robertsův operátor kvůli menší velikosti konvolučního jádra drží od ostatních detektorů znatelný odstup.

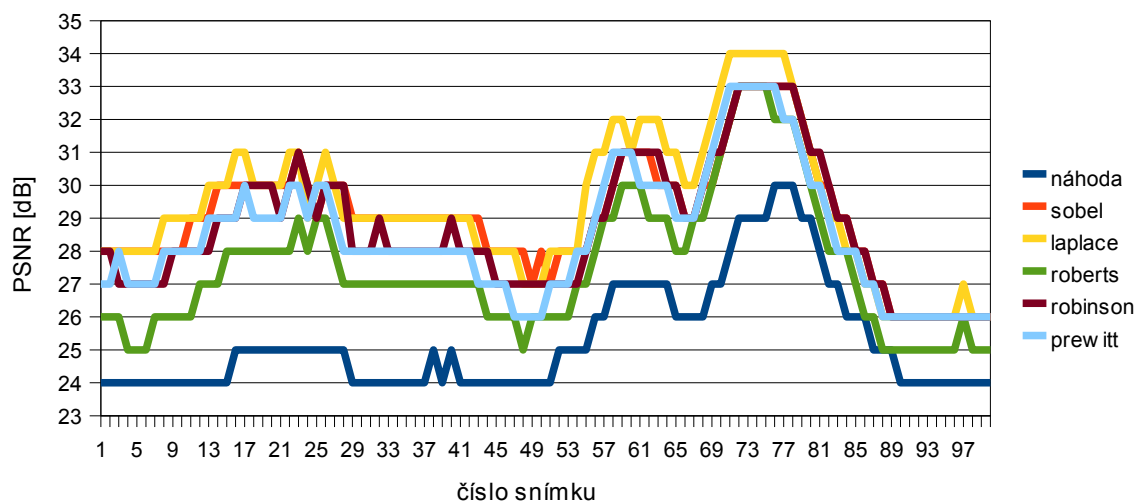


Graf 9.4: Závislost PSNR na procentu vložených bodů

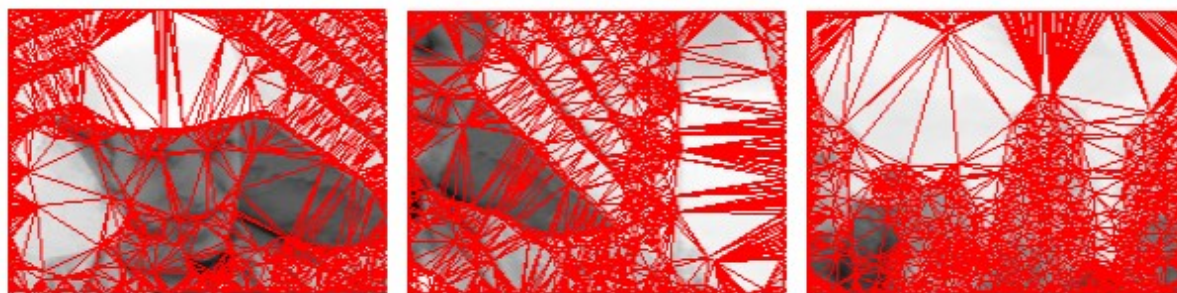
Podrobný průběh PSNR pro 20% vložených bodů ukazuje, že kvalita během trvání celé videosekvence významně kolísá. V grafu 9.5 lze najít minimálně tři výrazná lokální maxima (viz obr. 9.3) a jedno lokální minimum (viz obr. 9.4). Za maxima jsou zodpovědné velké, co do intenzity téměř homogenní, plochy, které se buď v obraze objeví anebo se zvětší.

Kapitola 9 | Testy a srovnání s dostupnými metodami

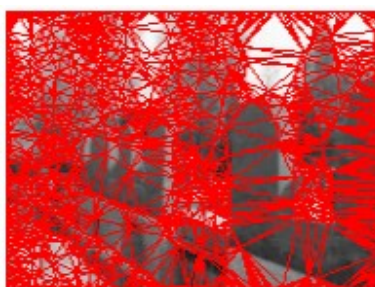
Mezi snímky 13-29 se do popředí dostane helma, kterou má na hlavě hlavní aktér. Ve snímcích 56-63 je na vině omítka domu na jedné straně jeřábu spolu s nebem na straně druhé. A konečně největší lokální maximum je zaznamenáno pro snímky 71-77, kdy jsou snímky extrémně přeexponované a téměř celé nebe je možno aproximovat jednou intenzitou. Lokální minimum ve snímcích 88-100 je způsobeno závěrečným pohledem kamery na torzo stavby společně s množstvím stromů. Vzhledem k tomu, že ve scéně je množství relevantních hran, není polovina z 20% vložených obrazových bodů k jejich pokrytí postačující a hrany trojúhelníků evidentně kříží hrany v obraze.



Graf 9.5: Podrobný průběh PSNR pro 20% vložených bodů



Obr. 9.3: K ilustraci lokálních maxim ve videosekvenci Foreman (řazeno chronologicky)



Obr. 9.4: Významné lokální minimum

Kapitola 9 | Testy a srovnání s dostupnými metodami

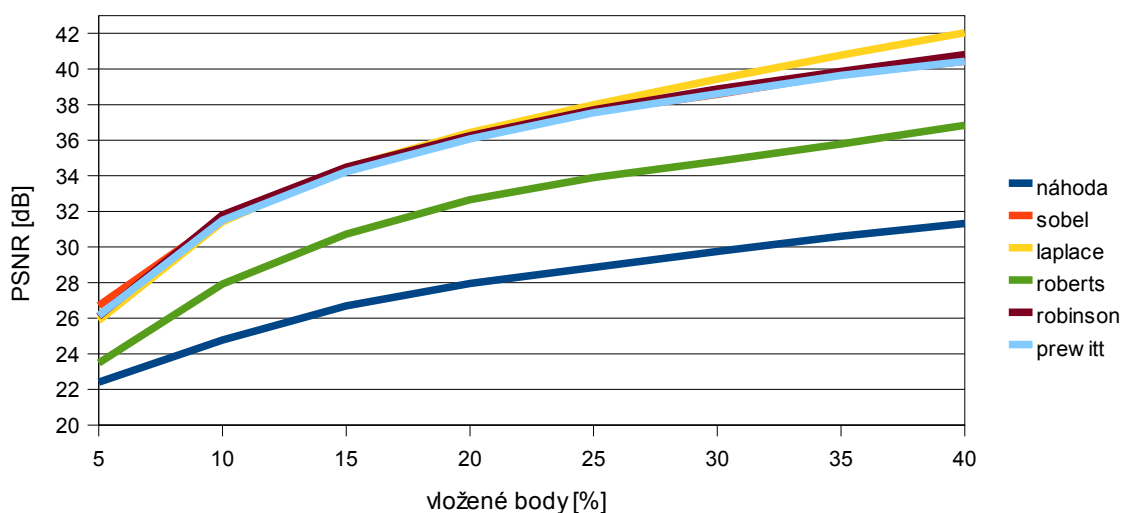
Renderované video: „Toasts“

Video Toasts o velikosti 192x144px bylo podrobně testováno pro 20% vložených bodů. První snímek videosekvence včetně porovnání s odpovídajícím původním snímkem lze vidět na obr. 9.5. Ilustrace postupu od vytvoření triangulace až po vykreslení snímku je na obr. A.5.



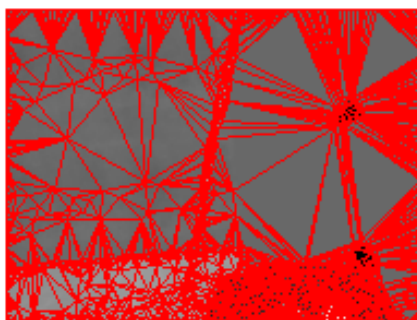
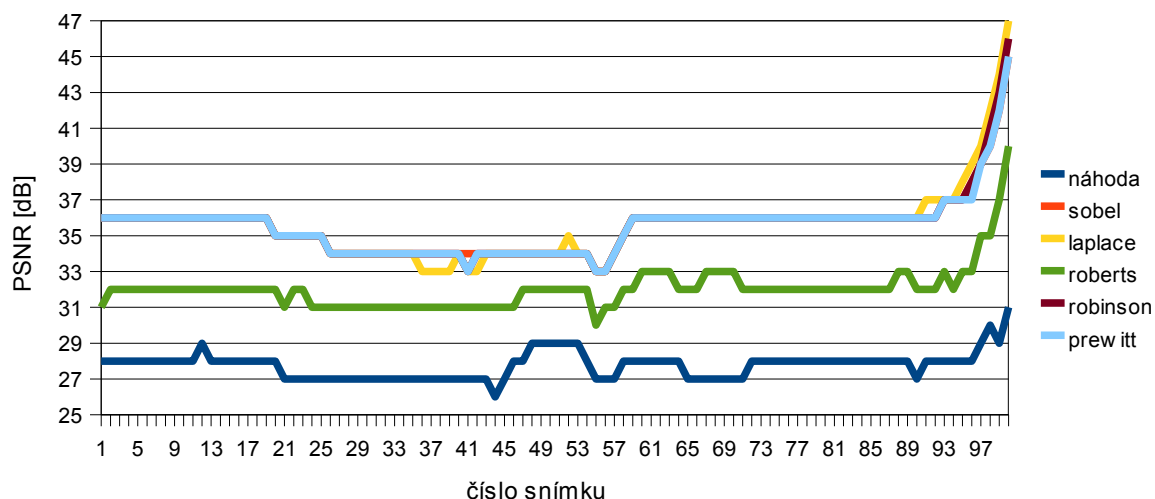
Obr. 9.5: Zleva: aproximace vložením 20% bodů, residuum, originální snímek

Pro renderované video je typické, že neobsahuje šum. Proto se dá očekávat, že naprostá většina nalezených hran budou skutečná ohraničení objektů ve scéně. Pro interval $\langle 5\% ; 40\% \rangle$ vložených bodů při poměru hranových bodů k náhodným 50:50 vyšlo z grafu najevo, že závislost PSNR na množství vložených bodů je logaritmická. Všechny detektory překonávají v aproximaci náhodný výběr už při 5% vložených bodů. Nejvyšší hodnoty PSNR poskytuje skupina operátorů Laplace, Sobel, Robinson a Prewitt. Laplace sice nad 30% poskytuje lepší výsledky než ostatní, ale jak ukázala kap. 9.2.2, vkládání bodů nad 33% se vzhledem k použité bezztrátové kompresi nevyplatí. Odchylka všech operátorů od čistě náhodného výběru dosahuje kromě Robertse zhruba 9dB. Robertsův operátor drží stejně jako v předchozím testu kvůli menší velikosti konvolučního jádra od ostatních detektorů odstup.



Kapitola 9 | Testy a srovnání s dostupnými metodami

Podrobný průběh PSNR pro 20% vložených bodů v grafu 9.7 ukazuje, že kvalita pro většinu snímků videosekvence je téměř konstantní. V závěru však lze najít jedno výrazné lokální maximum ve snímku 100 (viz obr. 9.6). Scéna se v té části výrazně změní tak, že z většiny zachycuje pouze jednu velkou homogenní plochu. Nicméně tato plocha je v obraze viditelně ohraničena, takže čistě náhodný výběr bodů nedosáhne stejného nárůstu PSNR jako při použití jakéhokoliv z hranových detektorů. Rozdíl v PSNR mezi čistě náhodným výběrem a hranovými detektory s výjimkou Robertse je v průběhu celé videosekvence přibližně 8dB.



Obr. 9.6: Významné lokální maximum

Kapitola 9 | Testy a srovnání s dostupnými metodami

Statická kamera: „Miss America“

Video Miss America o velikosti 176x144px bylo podrobně testováno pro 20% vložených bodů. První snímek videosekvence včetně porovnání s odpovídajícím původním snímkem lze vidět na obr. 9.7. Ilustrace postupu od vytvoření triangulace až po vykreslení snímku je na obr. A.6.



Obr. 9.7: Zleva: aproximace vložením 20% bodů, residuum, originální snímek

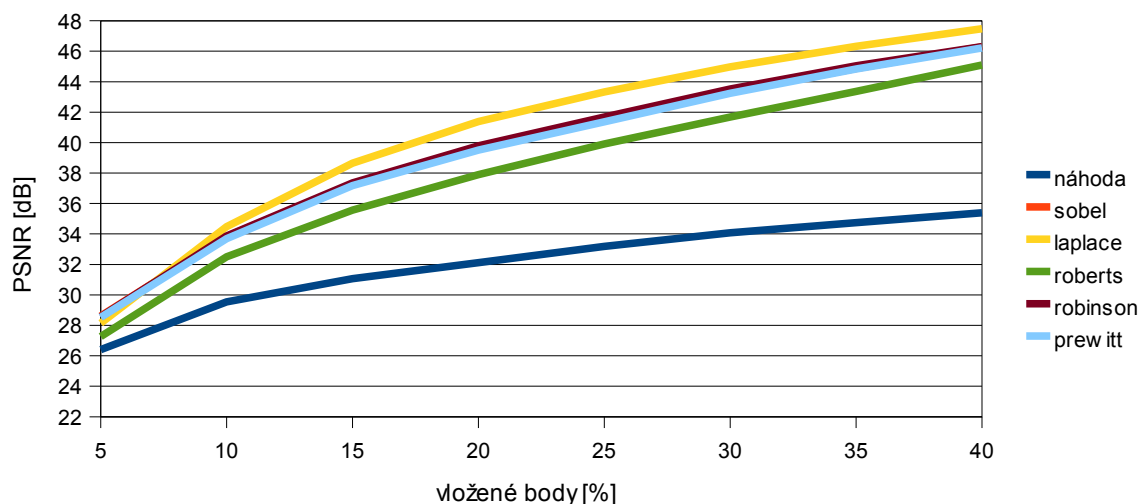
U testovaného videa typu talking head je pozadí téměř homogenní. V důsledku toho je v ní oproti předchozím dvěma zkoumaným videosekvencím podstatně méně hran. Práh při automatickém prahování proto klesal až k hodnotám intenzity kolem 30. V tu chvíli už byla polovina z 20% bodů obrazu na relevantní hranové body vyčerpána a v prahovaném obraze se začal objevovat šum (viz obr. 9.8).



Obr. 9.8: Detekce hran a následné prahování

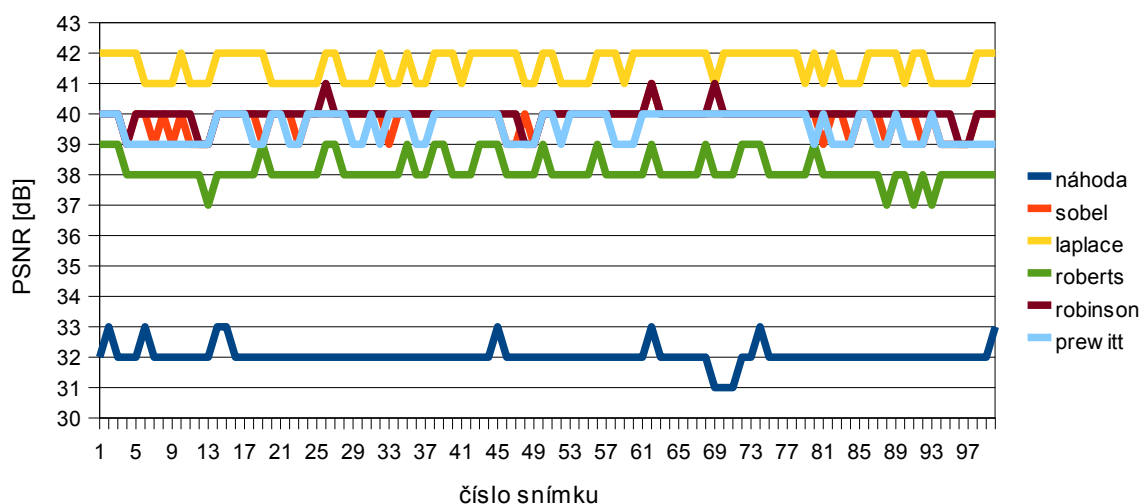
Při vynesení PSNR v závislosti na počtu vložených bodů v intervalu $\langle 5\%;40\% \rangle$ s poměrem hranových bodů k náhodným 50:50 vyšlo z grafu 9.8 najevo, že závislost je logaritmická. Lepší aproximaci než čistě náhodný výběr poskytují všechny detektory během celého intervalu vložených bodů. Laplaceův operátor poskytuje znatelně lepší výsledky nad 10% vložených bodů, ale ty oproti skupině hranových operátorů Sobel, Prewitt a Robinson představují rozdíl pouhých 2dB. Odchylna všech hranových operátorů od čistě náhodného výběru dosahuje kromě Robertse 8-10dB. Robertsův operátor drží stejně jako v předchozím testu kvůli menší velikosti konvolučního jádra od ostatních detektorů odstup. Laplaceův detektor je jasně nejlepší v celém testu.

Kapitola 9 | Testy a srovnání s dostupnými metodami



Graf 9.8: Závislost PSNR na procentu vložených bodů

Podrobný průběh PSNR pro 20% vložených bodů v grafu 9.9 ukazuje, že kvalita pro většinu snímků videosekvence je téměř konstantní. Rozdíl v PSNR mezi čistě náhodným výběrem a hranovými detektory s výjimkou Robertse je v průběhu celé videosekvence mezi 8 a 10dB.



Graf 9.9: Podrobný průběh PSNR pro 20% vložených bodů

9.2.4 Srovnání s decimační ADT a s kodekem M-JPEG

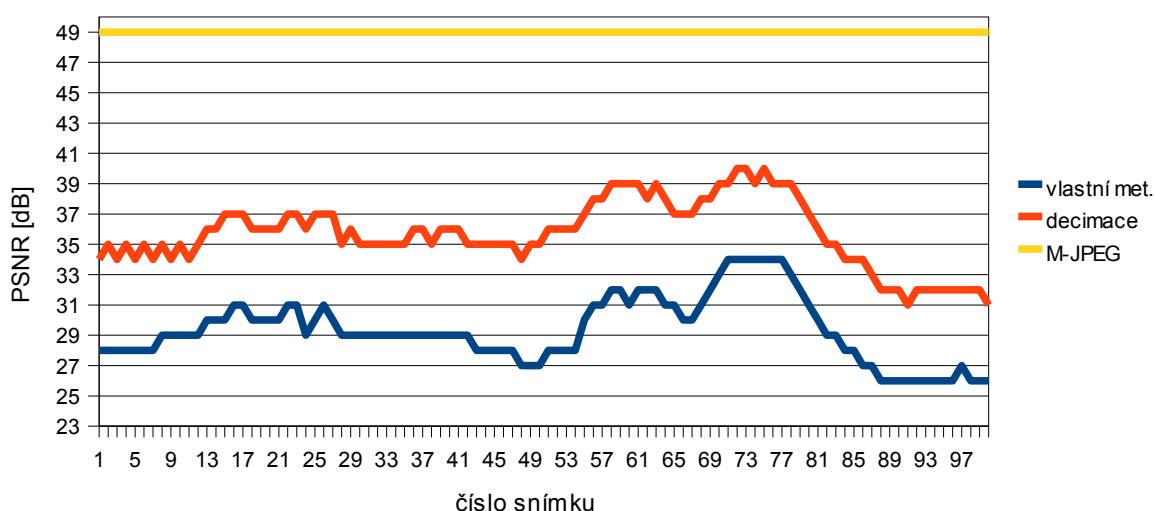
Video Foreman o velikosti 176x144px bylo podrobně provnáváno s existujícími metodami pro 20% vložených bodů. První snímek videosekvence zakódovaný jednotlivými metodami lze vidět na obr. 9.9.



Obr. 9.9: Porovnání výstupů vlastní metody, decimační metody a M-JPEG

Pro porovnání implementované hranové detekce¹⁵ (dále nazývána vlastní metoda) s metodou pro decimaci kompletní triangulace a s kodekem M-JPEG byly vyhotoveny grafy srovnávající vývoj PSNR (viz graf 9.10) a časové složitosti (viz graf 9.11). Pro zbylé dvě videosekvence byla celková odlišnost porovnávaných metod velmi malá, proto odpovídající grafy z důvodů zjednodušení nejsou uvedeny. Zaručení shodných podmínek pro všechny metody bylo zajištěno 20% vkládaných, resp. 80% odebíraných bodů, pro vlastní metodu, resp. pro metodu decimační. Kodér M-JPEG však není založen na vkládání určitého počtu bodů. Jeho parametry musely být v programu VirtualDub [VirtDB] před kompresí experimentálně nastaveny tak, aby velikost jím produkovaného videosouboru odpovídala vypočtené velikosti videosekvence vytvořené vlastní metodou (viz kap. 9.2.2). Ta při průměrné velikosti jednoho uloženého snímku 16481B činí pro všech 100 snímků součtem 1,648MB.

V grafu 9.10 lze vidět přibližně 6dB nárůst v PSNR, který má decimace oproti vlastní metodě. Na rozdíl od M-JPEG s nárůstem průměrně 14dB na decimaci a 20dB na vlastní



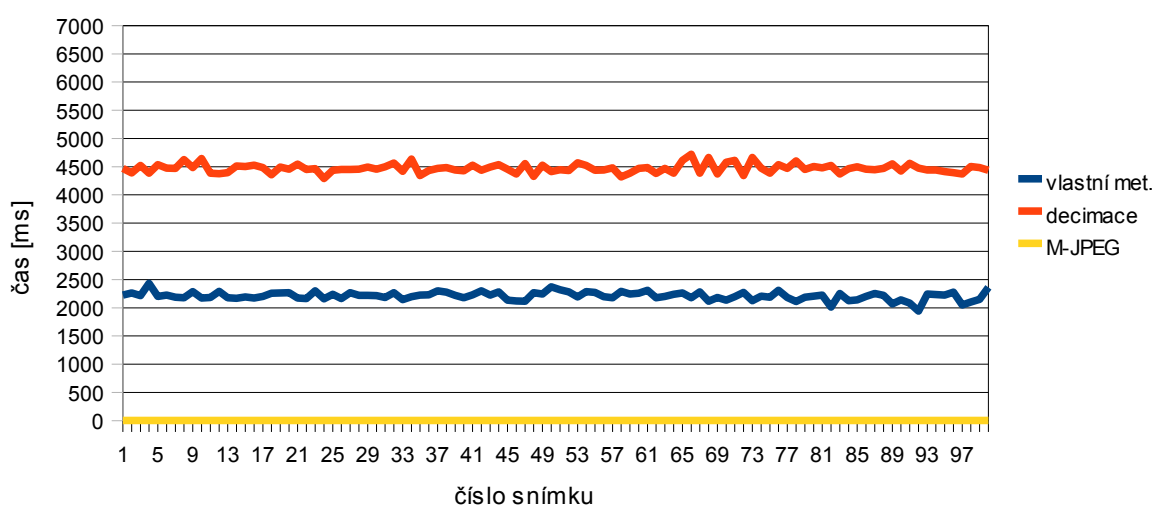
metodu není v metodách použita kontrola stavu kvality, což se v obou projevuje právě

15 Konkrétně jde o Laplaceův operátor, který si vedl nejlépe v předchozích testech.

Kapitola 9 | Testy a srovnání s dostupnými metodami

viditelným nekonstantním průběhem PSNR. Subjektivní porovnání na prvních snímcích videa lze provést nad obr. 9.9.

Jiné výsledky přineslo srovnání časové složitosti všech tří metod (viz graf 9.11). S nejnižším průměrným časem 5ms na snímek si poradil M-JPEG. Tato hodnota musela být určena jako procento z času zpracování celkového počtu 100 snímků, neboť podrobná statistika nebyla při procesu kódování k dispozici. Přibližně o dvě vteřiny více na každý snímek potřebuje vlastní metoda, která poskytuje rovněž poměrně vyrovnanou časovou složitost. Decimační metoda však pro konkrétní počet bodů vrací výsledky ve více jak dvakrát delším čase než vlastní metoda. Důvod vychází z její podstaty, kdy jsou odebírány body, jejichž nízká relevance musí být zjištěna algoritmem typu brutální síla.



Graf 9.11: Srovnání podrobného průběhu časové složitosti pro 20% vložených bodů

9.3 Intra + inter kódování

Tato kapitola je zaměřena na případ, kdy je videosekvence reprezentována jednak absolutně kódovanými snímky a jednak predikovanými snímky. Tyto snímky jsou v testech kombinovány v poměru 1:5 (GOP = 6) z důvodu přehlednosti a faktu, že pro větší GOP se kvalita velmi výrazně propadá. V první části se zaměříme na otestování datového toku a s ním souvisejícího kompresního poměru. Ve druhé části zjistíme, jaká je kvalita výstupu a časová složitost pro jednotlivé snímky testovacích videosekvencí. Při těchto testech uvažujeme počet vkládaných bodů 5%, neboť pro vyšší hodnoty je algoritmus již příliš časově náročný. Dále jako algoritmus pro tvorbu intra kódovaných snímků budeme používat decimaci, neboť ta nám oproti hranové detekci poskytne v intra kódovaných snímcích přesnější aproximaci, a my tak budeme moci lépe prozkoumat dopad kompenzace pohybu, která podle předpokladu bude kvalitu aproximace znatelně snižovat. V poslední části porovnáme výsledky s kodekem XviD založeným na MPEG-4 standardu, který stejně jako vlastní metoda používá intra i inter kódování.

Volba velikosti bloku

Nyní se zaměříme na to, jaké parametry a proč budeme volit v souvislosti s metodou block matching pro odhad pohybu. Prvním volitelným parametrem je velikost bloku. Obecně platí, že příliš malé bloky jsou extrémně citlivé na šum v obraze, zatímco příliš velké bloky produkují méně kvalitní korespondenci. K ověření této teorie bylo otestováno první GOP (šest snímků) všech videosekvencí pro velikosti bloku 4x4px, 8x8px, 16x16px a 24x24px. Výsledky ukazuje tab. 9.4, kde DYN označuje videosekvenci Foreman, REN je videosekvence Toasts a STA je Miss America. První snímek je vždy intra kódovaný, dalších pět snímků je z něj odvozeno. V první a třetí části tabulky se pro videosekvence Foreman a Miss America kvalita s velikostí bloku zvyšuje až do velikosti 16x16px, poté klesá. Pro video Toasts je dále pro 24x24px kvalita shodná s 16x16px. Na základě těchto testů budeme dále používat velikost bloku 16x16px.

	DYN 4	DYN 8	DYN 16	DYN 24	REN 4	REN 8	REN 16	REN 24	STA 4	STA 8	STA 16	STA 24
Snímek 1	28	28	28	28	34	34	34	34	39	39	39	39
Snímek 2	23	24	25	25	28	29	31	31	35	37	37	37
Snímek 3	22	23	24	24	26	27	29	29	33	35	35	35
Snímek 4	21	22	23	23	26	26	27	27	32	33	34	33
Snímek 5	20	21	22	22	25	25	26	26	31	32	32	32
Snímek 6	20	21	22	21	25	24	25	25	30	31	31	30

Volba velikosti vyhledávacího okénka

Dalším parametrem je velikost vyhledávacího okénka, která má přímou souvislost s maximálním detekovaným posunem ve scéně. Vzhledem k odhadované vysoké časové složitosti a s ohledem na míru pohybu ve videosekvencích zvolíme velikost 15x15px.

Výběr metriky pro korespondenci dvou bloků

Kapitola 9 | Testy a srovnání s dostupnými metodami

Posledním parametrem je volba vyhledávacího kritéria z množiny MAD, MSD a PDC (viz kap. 3.1). Protože nemáme možnost analyticky testovat, zda jsou nalezené vektory získané při odhadu pohybu věrnou aproximací optického toku, zaměříme se na kvalitu výsledné aproximace a časovou složitost pro jednotlivé testovací videosekvence, jak ukazují následující tabulky. Připomeňme, že první snímek GOP je vždy intra kódován a tedy odhad pohybu není třeba. U metriky PDC byl práh nastaven na hodnotu 5, což se v jejím případě ukázalo být optimální. V tabulce 9.5, 9.6 a 9.7 podává nejlepší výsledky ohledně PSNR metrika MAD. Naproti tomu nejlepší časovou složitost má ve všech testovacích videosekvencích metrika MSD. Na rozdíl od ostatních metrik pracuje v každém testu pouze s druhou mocninou rozdílů intenzit, takže není potřeba počítat absolutní hodnotu tvořenou jedním testem a jedním součinem jako v případě MAD nebo dvěma testy a jedním součinem jako v případě PDC. Průchod s metrikou PDC je nejpomalejší a zároveň ani výsledná PSNR nesevčí o vhodnosti jejího použití. V dalších měřeních budeme používat metriku MAD, protože poskytuje nejkvalitnější výsledky a časově se příliš od MSD neliší.

	MAD PSNR [dB]	MSD PSNR [dB]	PDC PSNR [dB]		MAD čas [ms]	MSD čas [ms]	PDC čas [ms]
Snímek 1	28	28	28		-	-	-
Snímek 2	25	25	25		1480	1072	2229
Snímek 3	24	24	23		1492	1072	2271
Snímek 4	23	22	22		1454	1064	2137
Snímek 5	22	22	21		1497	1069	2293
Snímek 6	22	21	20		1501	1082	2261

Tabulka 9.5: Srovnání metrik pro odhad pohybu pro 1. GOP videa Foreman

	MAD PSNR [dB]	MSD PSNR [dB]	PDC PSNR [dB]		MAD čas [ms]	MSD čas [ms]	PDC čas [ms]
Snímek 1	34	34	34		-	-	-
Snímek 2	31	30	31		2068	1533	2571
Snímek 3	29	29	29		2101	1540	2616
Snímek 4	27	27	27		2102	1543	2593
Snímek 5	26	26	26		2089	1542	2579
Snímek 6	25	25	25		2081	1543	2549

Tabulka 9.6: Srovnání metrik pro odhad pohybu pro 1. GOP videa Toasts

	MAD PSNR [dB]	MSD PSNR [dB]	PDC PSNR [dB]		MAD čas [ms]	MSD čas [ms]	PDC čas [ms]
Snímek 1	39	39	39		-	-	-
Snímek 2	37	37	37		1910	1386	2331
Snímek 3	35	35	35		1913	1378	2316
Snímek 4	34	34	34		1937	1386	2365
Snímek 5	32	32	32		1940	1401	2340
Snímek 6	31	31	30		1939	1411	2294

Tabulka 9.7: Srovnání metrik pro odhad pohybu pro 1. GOP videa Miss America

9.3.1 Časová složitost

Nyní se podíváme na to, z jakých dílčích celků se skládá časová složitost výsledného řešení. Následující tabulka 9.8 ukazuje pro oba typy vykreslení, jak se na výpočtu predikovaného snímku jednotlivých videosekvencí podílí odhad pohybu, posun DT, vykreslení a kódování. DYN označuje video Foreman, REN je Toasts a STA je Miss America.

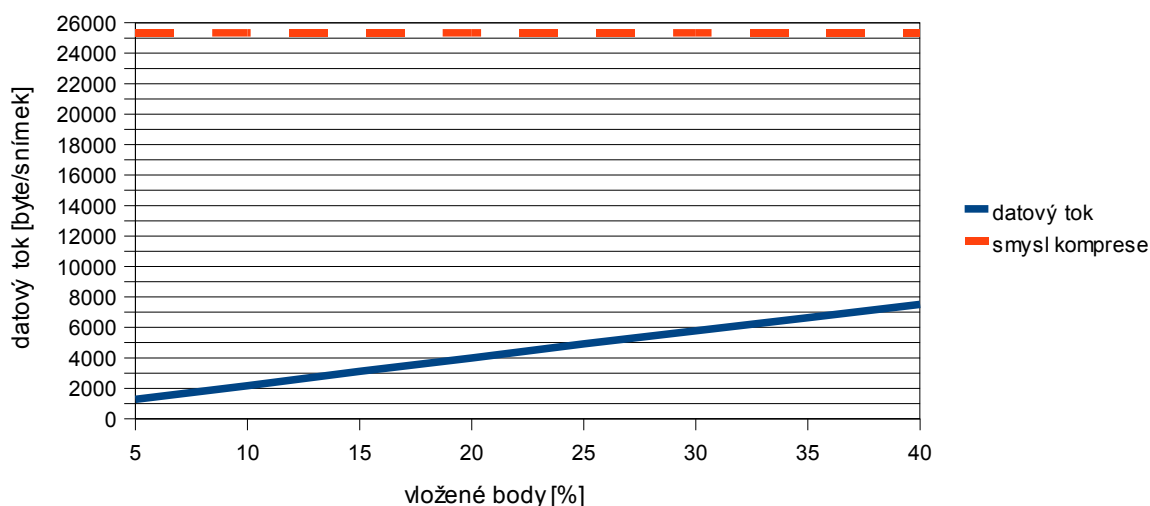
Kapitola 9 | Testy a srovnání s dostupnými metodami

	odhad pohybu [ms]	posun DT [ms]	vykreslení [ms]	kódování [ms]
DYN – interp	3824	4248	115	14
DYN – warping	3788	4411	515	19
REN – interp	4584	2162	123	18
REN – warping	4613	2061	403	14
STA – interp	4396	128	110	12
STA – warping	4378	128	350	13

Tabulka dokazuje, že nejvyšší časovou složitost má na svědomí odhad pohybu a jeho kompenzace posunutím DT. Případ Miss America tvoří výjimku v časové složitosti posunutí, neboť zde dochází k pohybu jen v minimální míře. Čas potřebný pro vykreslení je u jednotlivých videosekvencí v případě warpingu o 300-500% vyšší než v případě interpolace. V kontextu celkové časové složitosti se však jedná o zanedbatelný rozdíl.

9.3.2 Datový tok a kompresní poměr

Vzhledem k vysoké časové složitosti odhadu a kompenzace pohybu byl datový tok zjišťován pouze pro první GOP videa Foreman s velikostí původního snímku 25344B. Graf 9.12 poskytuje informaci o průměrné velikosti snímku při konkrétním procentu vložených bodů. Užitá byla bezztrátová komprese (viz kap. 7.4) DELTA+GZIP na první intra snímek v GOP a pak vždy RAW+GZIP na ostatních 5 predikovaných snímků. Z výsledku v grafu je vidět, že smysl komprese je dostatečně vzdálen i při 40% vložených bodů. Jaký podíl mají na datovém toku intra a inter kódované snímky, ukáže další kapitola.



Pro 5% vložených bodů, které budeme používat v dalších testech, je průměrná velikost jednoho snímku 1280B a kompresní poměr tedy je zhruba 20:1. Pro 30% vložených bodů je jeden snímek veliký 5779B s kompresním poměrem přibližně 4,5:1.

9.3.3 Podrobné testy

Hlavním cílem podrobných testů je zjistit, jak moc se mění PSNR, datový tok a časová složitost v inter kódovaných snímcích v závislosti jejich pozici v GOP. K vykreslení dat z triangulace je použita jednak interpolace za pomoci barycentrických souřadnic a jednak warping se třemi páry úseček. Podružným cílem testů je posouzení, který z těchto dvou způsobů je pro testovací videosekvence lepší použít. K těmto účelům budeme v případě testů kvality používat metriku PSNR i subjektivní porovnání. V případě datového toku si vystačíme se zjištěním, kolik bytů je potřeba pro reprezentaci jednoho snímku při bezeztrátové kompresi DELTA+GZIP pro intra a RAW+GZIP pro inter kódované snímky. Časovou složitost měříme v každém ze snímků odděleně pro každou metodu vykreslení.

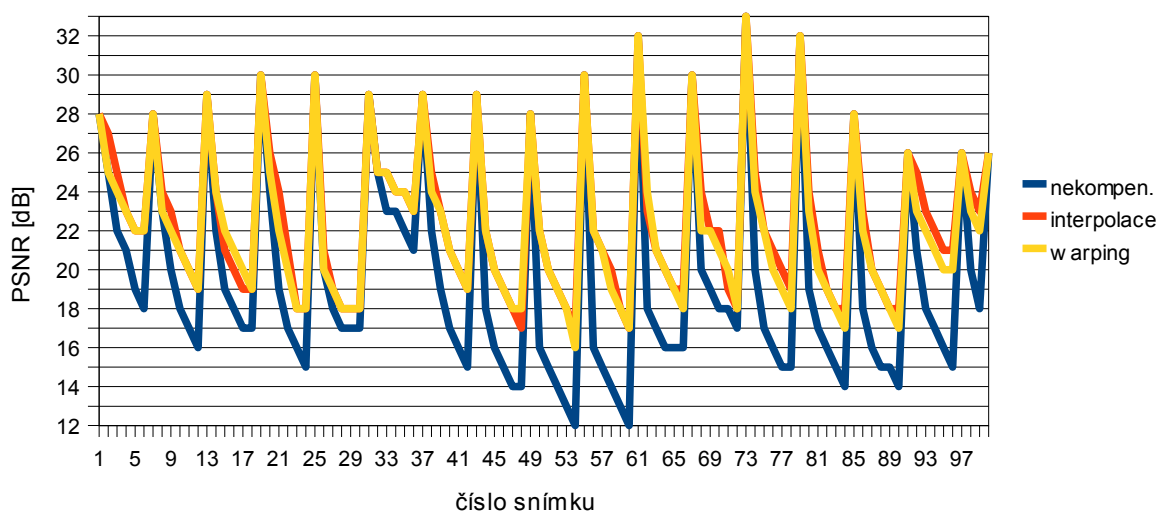
Podrobný průběh PSNR pro 5% vložených bodů v následujících grafech ukazuje, jaká je kvalita snímků videosekvencí získaných interpolací vzorků a snímků videosekvencí získaných warpingem po odhadu pohybu algoritmem block matching a kompenzací pohybem DT. Každý šestý snímek je intra kódovaný a je vytvořený decimační triangulací. Protože obsahuje veškeré údaje o poloze bodů získané metodou pro statický snímek, je v něm aproximace poměrně dobrá. Následujících pět snímků v každé šestici bylo z tohoto a odpovídajících vstupních snímků dopočteno.

I když o kvalitě každého snímku rozhoduje hlavně sama velikost PSNR, pro určení kvality aproximace zavedeme ještě dodatečnou referenční videosekvenci, kterou nazveme nekompenzované video. Její zavedení je motivováno potřebou zjistit, jak velký vliv na kvalitu má samotná kompenzace pohybu. Pokud by ke kompenzaci pohybu v inter kódovaných snímcích nedocházelo, byl by výstupem kódování všech snímků v GOP vždy jen nezměněný intra kódovaný snímek, který je v GOP na první pozici. Přesně podle této úvahy bylo vytvořeno a vyhodnoceno nekompenzované video, jehož kvalita je pod zkratkou „nekompen.“ vynesena v následujících grafech. Je-li tato kvalita v celé GOP vysoká, pak v daných snímcích nedochází k většímu pohybu a naopak. Hlavní ukazatel, který nám nekompenzované video poskytuje, spočívá v rozdílu jeho kvality a kvality pohybově kompenzovaných videosekvencí. V případě, že je v dané GOP tento rozdíl velký, znamená to, že odhad a kompenzace pohybu ve vyšetřovaných snímcích zafungovaly velmi dobře. Je-li tento rozdíl naopak malý, pak mohou nastat dva případy podle velikosti propadu kvality nekompenzovaného videa. V případě malého propadu je scéna v dané GOP téměř statická a je na místě zvážit, zda má zde kompenzace pohybu smysl. Jestliže je zároveň s malou hodnotou rozdílu kvality kompenzovaného a nekompenzovaného videa propad kvality nekompenzovaného videa velký, pak odhad a kompenzace pohybu zafungovaly velice špatně, neboť nereflektují optický tok v původní videosekvenci.

Kapitola 9 | Testy a srovnání s dostupnými metodami

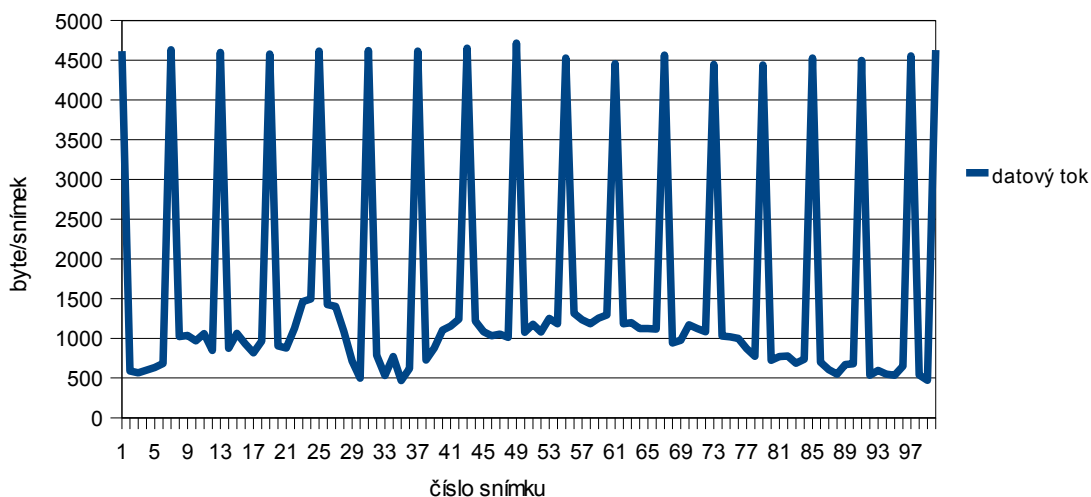
Pohyblivá kamera: „Foreman“

Při pohledu na graf 9.13 se kvalita snímků vzniklých po kompenzaci pohybu zřetelně liší od kvality, kterou dávají snímky nekompensovaného videa. Úroveň propadu nekompensovaného videa je evidentně dána množstvím pohybu ve scéně. Oproti předchozímu intra kódovanému snímku se PSNR kompenzovaných snímků block matchingem snižuje zhruba o 70-80%, kde 100% je kvalita nekompensovaného videa vynesena v grafu. Nejhorší aproximace je ve snímcích 18-29, kdy hlavní aktér provede velmi rychlé gesto rukou. Nejlepší je naopak uprostřed videosekvence ve snímcích 42-59, kdy dochází k posunu a rotaci budovy v pozadí scény. Tato budova obsahuje poměrně homogenní stěny co do intenzity a po detekci hran je velmi dobře ohraničena. Porovnání kvality snímků vykreslených interpolací a snímků vykreslených warpingem vyznívá nepatrně ve prospěch interpolace. Subjektivní dojem z obou aproximací ale může být podstatně jiný, protože při warpingu nedochází ke vzniku trojúhelníkových artefaktů (viz obr. A.7 a obr. A.8).



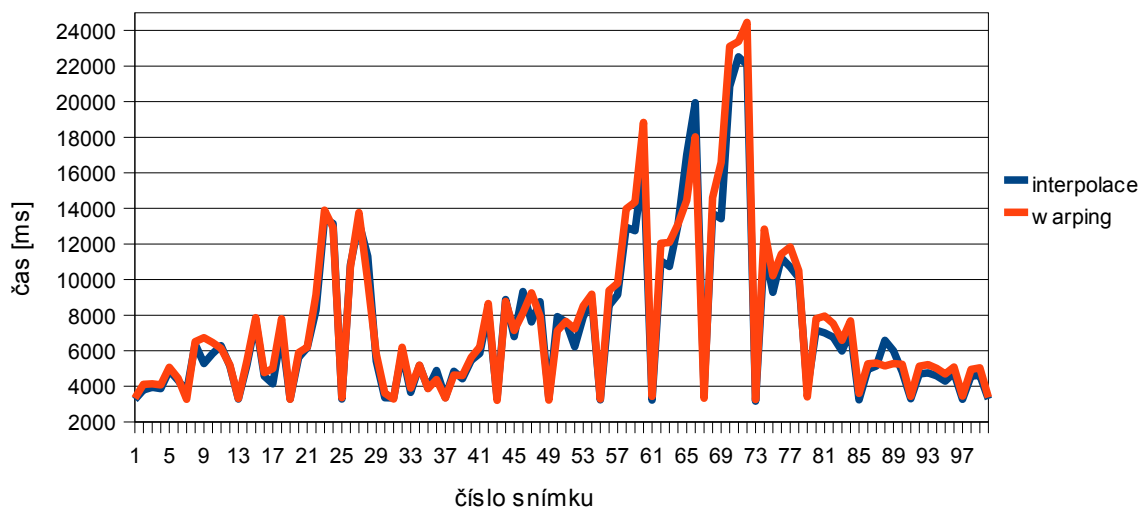
V grafu 9.14 je vynesena podrobný datový tok pro všechny snímky videosekvence. Protože nezávisí na typu vykreslení, je uvedena pouze jedna křivka. V intra kódovaných snímcích je datový tok poměrně stálý s průměrnými 4600B, zatímco predikované snímky se pohybují mezi 500 a 1500B. V nejlepším případě je tedy velikost predikovaných snímků v poměru 1:9 k intra kódovaným snímkům, v nejhorším pak 1:3. Průměrná velikost predikovaných snímků je 925B, což značí poměr 1:5 k intra kódovaným snímkům. Velikost predikovaných snímků záleží na množství pohybu ve scéně, přesněji na počtu nenulových vektorů posunu, které jsou vstupem bezetrátového kódování. Důkazem jsou snímky 6-23, kdy aktér pomalu kýve hlavou a snímky 18-29, kdy aktér provede velmi rychlé gesto rukou. Viditelně se na datovém toku projeví celá skupina snímků 36-95, kdy se kamera nerovnoměrně otáčí na jiné místo ve scéně.

Kapitola 9 | Testy a srovnání s dostupnými metodami



Graf 9.14: Podrobný průběh datového toku pro 5% vložených bodů

Graf 9.15 vypovídá o průběžné časové složitosti. Ta na rozdíl od datového toku nezávisí ani tak na množství pohybujících se bodů ve scéně jako na velikosti jednotlivých vektorů posunu. Dvě výrazná maxima to potvrzují. První případ ve snímcích 18-29 představuje zmíněné rychlé gesto a druhý případ ve snímcích 54-71 má za následek dočasně zrychlené otáčení kamery, které se ve zbylých snímcích videosekvence ustálí.

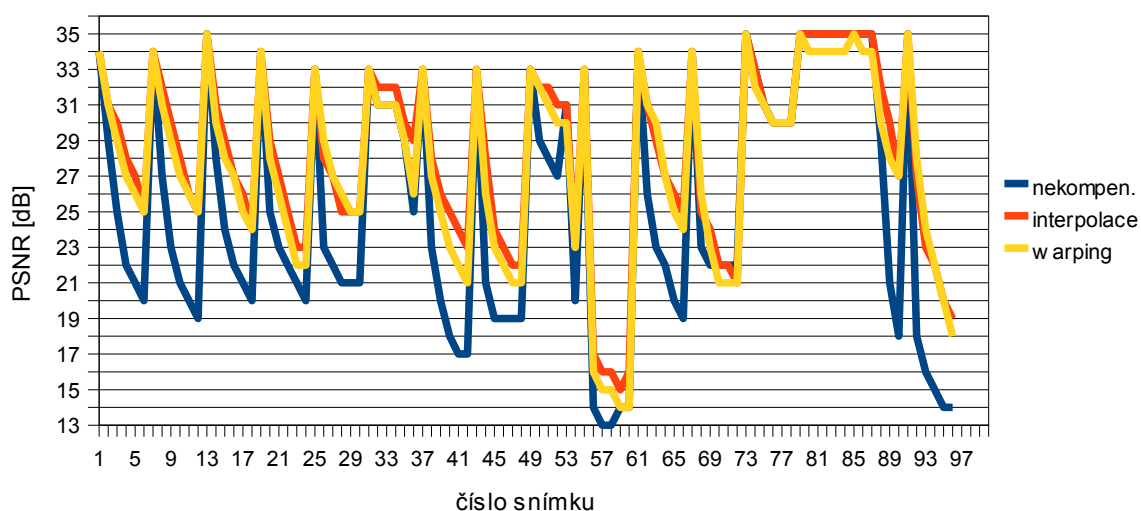


Graf 9.15: Podrobný průběh časové složitosti pro 5% vložených bodů

Kapitola 9 | Testy a srovnání s dostupnými metodami

Renderované video: „Toasts“

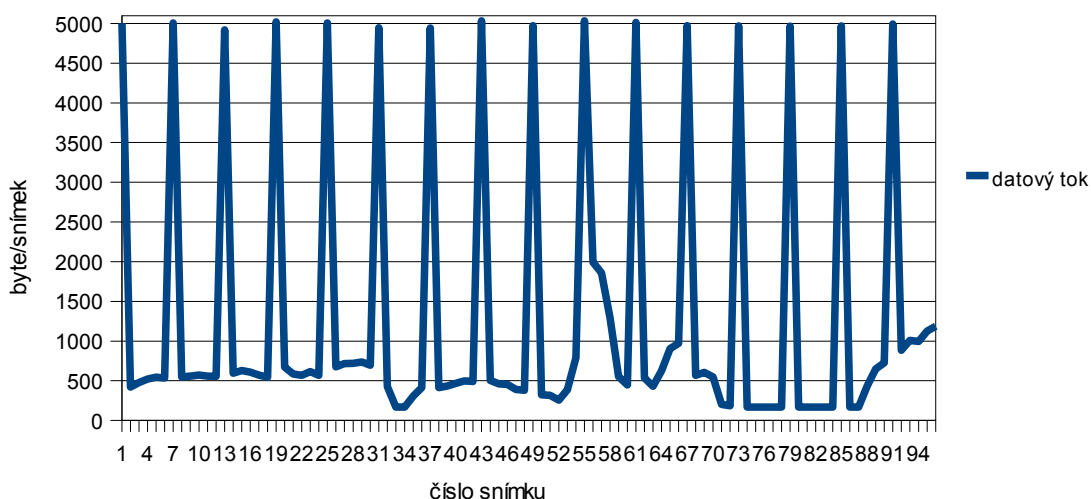
První polovina grafu 9.16 je význačná tím, že se kvalita snímků vzniklých po kompenzaci pohybu zřetelně liší od kvality, kterou dávají snímky nekompenzovaného videa. Úroveň propadu kvality nekompenzovaného videa je evidentně dána množstvím pohybu ve scéně. V druhé polovině grafu je pak odlišnost kvality kompenzovaných snímků často totožná s kvalitou pro nekompenzované video a celkově kvalita obou reprezentací velmi kolísá. Konec grafu je nekompletní, protože metoda vytvářející decimální triangulaci v intra kódovaném snímku při zpracování snímku 97 způsobí pád programu. Oproti předchozímu intra kódovanému snímku se PSNR kompenzovaných snímků block matchingem snižuje zhruba o 60-100%, kde 100% je kvalita nekompenzovaného videa vynesena v grafu. Nejhorší aproximace je ve snímcích 54-59, kdy dojde k velmi rychlé katapultaci jedné topinky topinkou druhou. V této jediné části videosekvece evidentně nedostačuje zvolená velikost vyhledávacího okenka 15x15px a některé body zůstanou statické i v případě, že se v odpovídajících původních snímcích pohybují. Dále je aproximace velmi špatná ve snímcích 90-95, kdy se kamera rychle pohybuje do míst, kde se nacházejí nové objekty a hrany, které budou zohledněny až v následujícím intra kódovaném snímku. V grafu lze rovněž nalézt minimálně čtyři GOP, kde se kvalita kompenzovaných snímků ani nekompenzovaného videa výrazněji nepropadá. Konkrétně jde o skupiny snímků 31-36, 49-54, 73-78 a 79-84. Ve všech těchto případech dojde k tomu, že scéna je v daném intervalu téměř statická. Porovnání kvality snímků vykreslených interpolací a snímků vykreslených warpingem vyznívá nepatrně ve prospěch interpolace. Subjektivní dojem z obou aproximací ale může být podstatně jiný, protože při warpingu nedochází ke vzniku trojúhelníkových artefaktů, které působí velmi rušivě (viz obr. A.11 a obr.A.12).



V grafu 9.17 je vynesena podrobný datový tok pro všechny snímky videosekvece. Protože nezávisí na typu vykreslení, je uvedena pouze jedna křivka. V intra kódovaných snímcích je datový tok poměrně stálý s průměrnými 5000B, zatímco predikované snímky se pohybují mezi 150 a 2000B. V nejlepší případě je tedy velikost predikovaných snímků v poměru 1:33

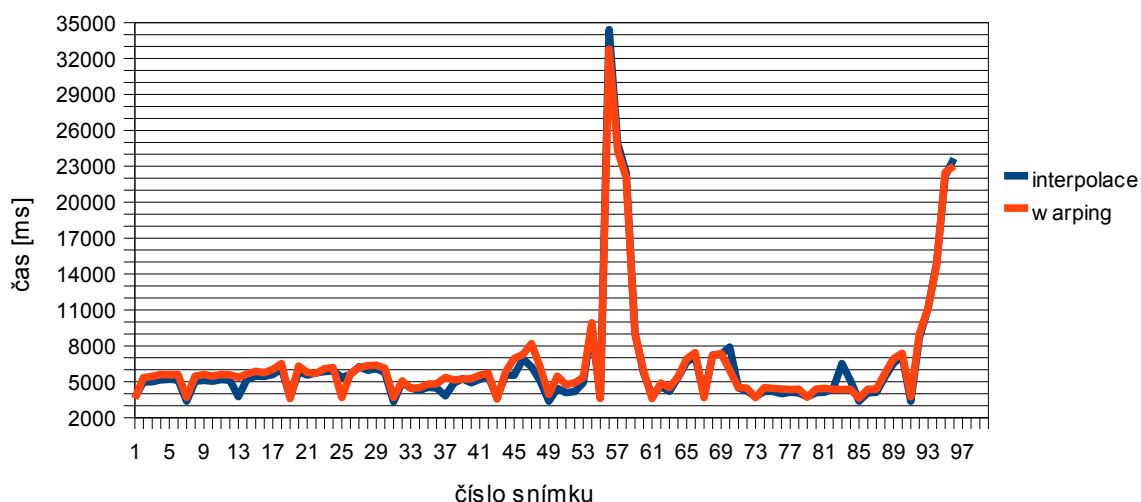
Kapitola 9 | Testy a srovnání s dostupnými metodami

k intra kódovaným snímkům, v nejhorším pak 1:2,5. Průměrná velikost predikovaných snímků je 545B, což značí poměr 1:9 k intra kódovaným snímkům. Velikost predikovaných snímků závisí na množství pohybu ve scéně, přesněji na počtu nenulových vektorů posunu, které jdou do bezztrátového kódování. Důkazem je skupina snímků 55-60, kdy se hýbou obě topinky a vidlička mezi nimi. Dále je poměrně markantní vzestup datového toku ve dvou skupinách snímků 85-96, kdy se kamera velmi rychle otáčí vzhůru a všechny body ve scéně se pohybují.



Graf 9.17: Podrobný průběh datového toku pro 5% vložených bodů

Graf 9.18 vypovídá o průběžné časové složitosti. Ta na rozdíl od datového toku nezávisí ani tak na množství pohybujících se bodů ve scéně jako na velikosti jednotlivých vektorů posunu. Dvě výrazná maxima to potvrzují. Jsou totožná s maximy popisovanými u grafu 9.17 a popisují nejrychlejší posuny ve scéně. Rozdíl časové složitosti při vykreslení dat z triangulace interpolací a warpingem je evidentně statisticky zanedbatelný.

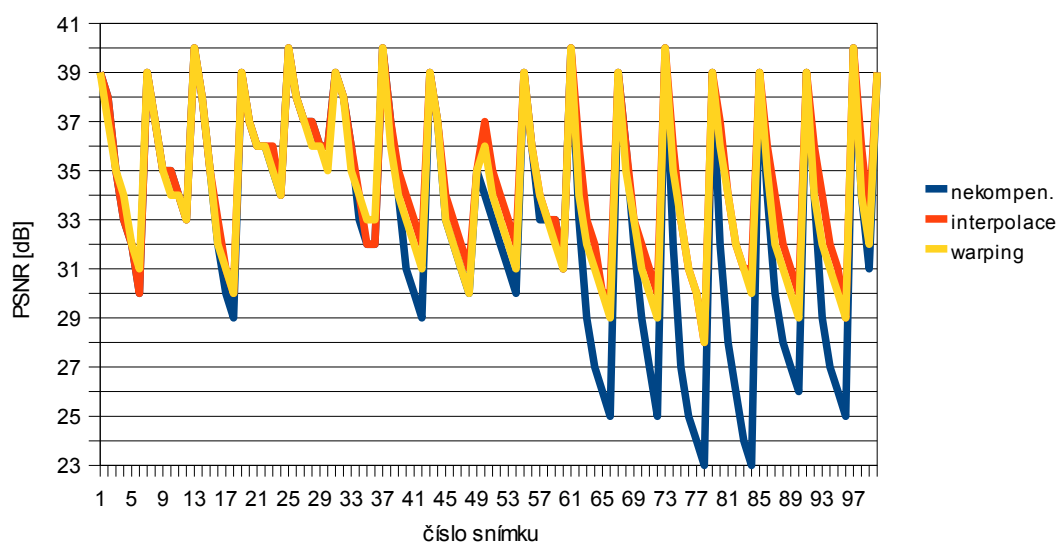


Graf 9.18: Podrobný průběh časové složitosti pro 5% vložených bodů

Kapitola 9 | Testy a srovnání s dostupnými metodami

Statická kamera: „Miss America“

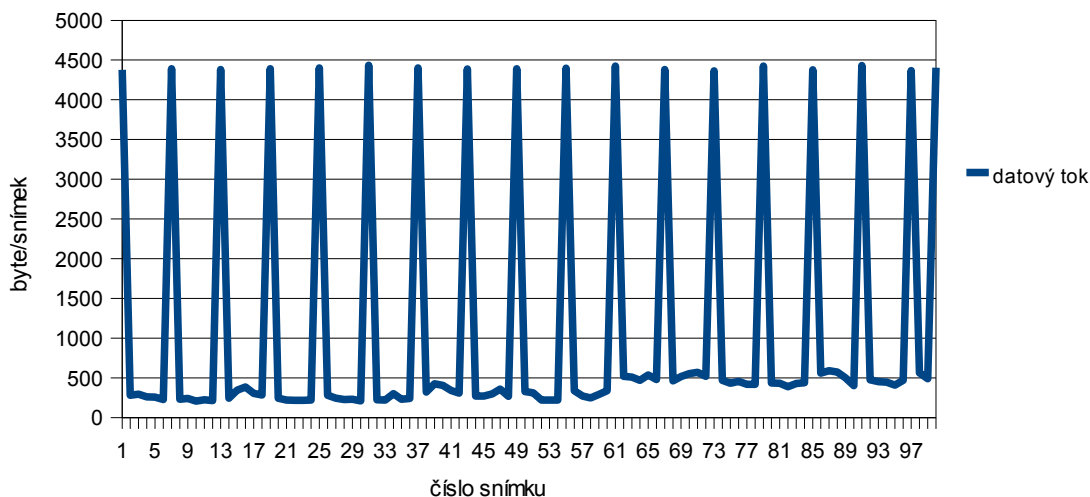
V grafu 9.19 je vynesena PSNR pro 5% vložených bodů. V první polovině grafu se kvalita snímků vzniklých po kompenzaci pohybu od kvality, kterou dávají snímky nekompensovaného videa, téměř vůbec neliší. Úroveň propadu kvality nekompensovaného videa je evidentně dána množstvím pohybu ve scéně. Ten je nejvíce znát ve druhé polovině grafu odpovídající snímkům, které jsou obecně bohatší na pohyb než snímky odpovídající první polovině grafu. Důvodem, proč v první části není reakce kompenzace pohybu na odhad pohybu adekvátní, je skutečnost, že k pohybu dochází z větší části na místech (jedná se hlavně o oči a ústa hlavní aktérky), která jsou velmi hustě vzorkována body z triangulace. Při takové koncentraci bodů dochází velmi často ke snaze o přesunutí dvou a více bodů na stejnou souřadnici ve výstupním obraze. To ale algoritmus pro posun bodů triangulace při současné implementaci neumožňuje. Ve druhé části grafu se oproti předchozímu intra kódovanému snímku PSNR kompenzovaných snímků block matchingem snižuje zhruba o 60-80%, kde 100% je kvalita nekompensovaného videa vynesena v grafu. Pohyb, který má v této části za následek to, že se PSNR v predikovaných snímcích tolik nesnižuje, je pohyb těla aktérky, který implementovaný odhad a kompenzace pohybu velice dobře reflektují. Porovnání kvality snímků vykreslených interpolací a snímků vykreslených warpingem vyznívá nepatrně ve prospěch interpolace. Subjektivní dojem z obou aproximací ale může být podstatně jiný, protože při warpingu nedochází ke vzniku trojúhelníkových artefaktů (viz obr. A.9 a obr. A.10).



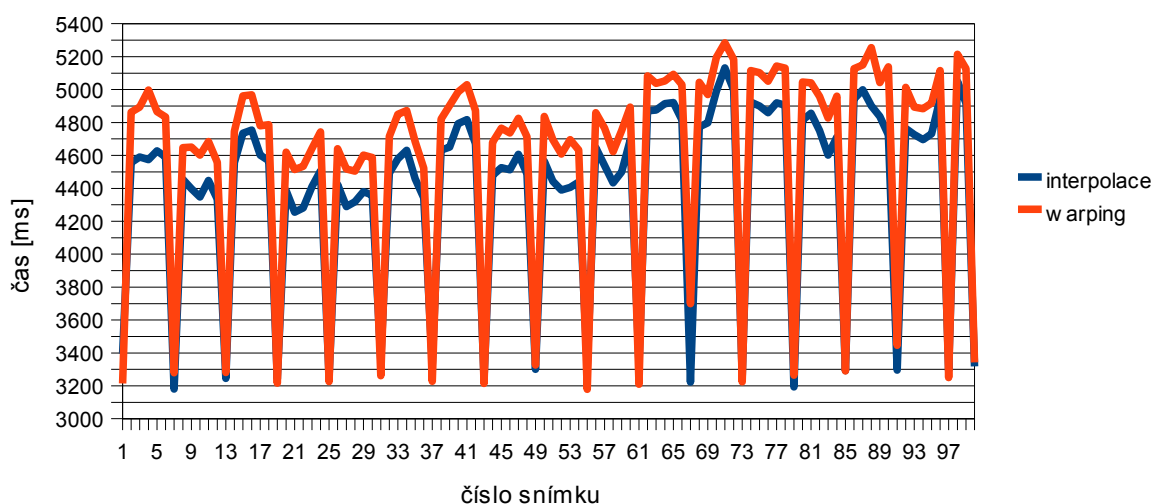
V grafu 9.20 je vynesena podrobný datový tok pro všechny snímky videosekvence. Protože nezávisí na typu vykreslení, je uvedena pouze jedna křivka. V intra kódovaných snímcích je datový tok poměrně stálý s průměrnými 4500B, zatímco predikované snímky se pohybují mezi 200-600B. V nejlepším případě je tedy velikost predikovaných snímků v poměru 1:22 k intra kódovaným snímkům, v nejhorším pak 1:7,5. Průměrná velikost predikovaných

Kapitola 9 | Testy a srovnání s dostupnými metodami

snímků je 352B, což značí poměr 1:13 k intra kódovaným snímkům. Velikost predikovaných snímků závisí na množství pohybu ve scéně, přesněji na počtu nenulových vektorů posunu, které jdou do bezetrátového kódování. Zmíněná druhá část videosekvence, kde dochází k mírnému pohybu hlavní aktérky, dokazuje, že příspěvek k vyššímu datovému toku je v grafu znát.



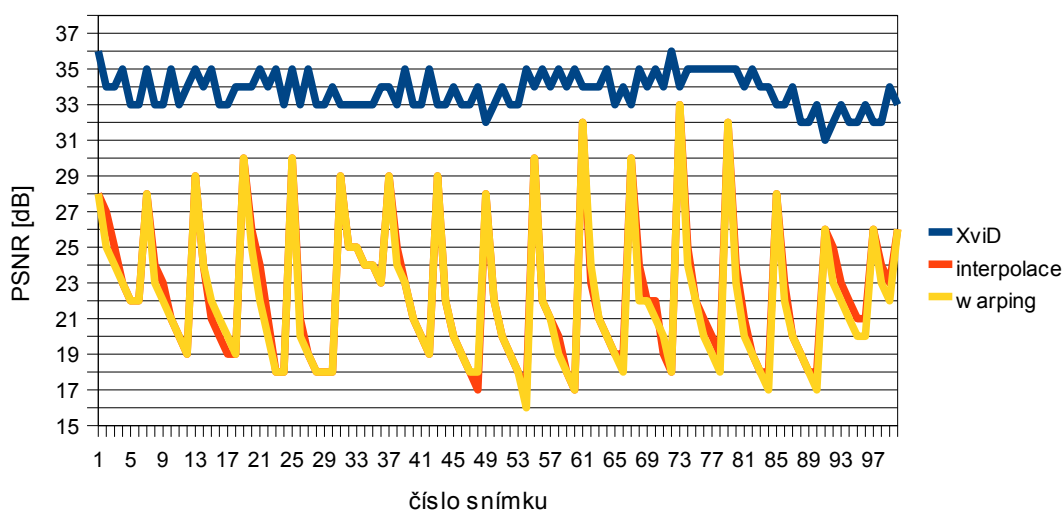
Graf 9.18 vypovídá o průběžné časové složitosti. Ta na rozdíl od datového toku nezávisí ani tak na množství pohybujících se bodů ve scéně jako na velikosti jednotlivých vektorů posunu. Vzhledem k tomu, že ve scéně k pohybu téměř nedochází, je rozdíl v časové složitosti výpočtu predikovaných snímků minimální. Rozdíl časové složitosti pro oba typy vykreslení ukazuje na vyšší časovou náročnost warpingu, která v tomto případě představuje zhruba 200ms. Konkrétní podíl dílčích celků na celkové složitosti můžeme najít v kap. 9.3. Drobné odchylky lze přičíst nepřesnosti při měření.



Graf 9.21: Podrobný průběh časové složitosti pro 5% vložených bodů

9.3.4 Srovnání s kodekem XviD

Pro porovnání implementované hranové detekce s kodekem XviD vycházejícím z MPEG-4 byl vyhotoven graf srovnávající vývoj PSNR (viz graf 9.22). Představa vytvořit graf pro srovnání časové složitosti se ukázala být jako lichá, neboť XviD zpracuje jeden snímek v řádu jednotek milisekund, zatímco vlastní metoda k tomu potřebuje jednotky sekund. Porovnání bylo provedeno pro jednotlivé snímky videosekvence Foreman. Pro zbylé dvě videosekvence byla celková odlišnost porovnávaných metod velmi malá, proto odpovídající grafy z důvodů zjednodušení nejsou uvedeny. Kodér XviD však není založen na vkládání určitého počtu bodů. Jeho parametry musely být v programu VirtualDub [VirtDB] před kompresí experimentálně nastaveny tak, aby velikost jím produkovaného videosouboru odpovídala vypočtené velikosti videosekvence vytvořené vlastní metodou (viz kap. 9.3.2). Ta při průměrné velikosti jednoho uloženého snímku 1280B činí pro všech 100 snímků součtem 128kB.



XviD dosahuje poměrně stabilní kvality, která se vyjádřeno v normě PSNR liší od vlastní metody v klíčových snímcích průměrně o 5dB a oproti predikovaným snímkům se odstup zvyšuje až na 15dB. Vzhledem k opravdu nízkému datovému toku jsou v obraze přítomny blokové artefakty, které PSNR podstatně snižují (porovnání viz obr. 9.10).



Obr. 9.10: Porovnání výstupu vlastní metody a XviD pro první snímek videa Foreman

10 Závěr

Na reprezentaci videa za pomoci triangulací jsou kladeny požadavky zahrnující vytvoření alternativy ke stávajícím reprezentacím takové, že je možné provádět morfologické změny pouze s triangulací a očekávat odpovídající změny v matici obrazu. Sekundárními požadavky jsou minimalizace časové složitosti, minimalizace datového toku a maximalizace kvality. Pro tyto účely byla úspěšně navržena a implementována metoda, která kóduje snímky do geometrické podoby tak, že je lze použít pro cíle dané prioritním požadavkem. V případě sekundárních požadavků rozlišuje řešení založené na intra kódování k maximalizaci kvality a minimalizaci časové složitosti. Dále přichází s řešením na bázi inter kódování vyznačující se snímkovou predikcí k minimalizaci datového toku.

Při reprezentaci celé videosekvence intra kódovanými snímky vyšel z testů nejlépe Laplaceův operátor těsně následovaný operátory Robinson, Prewittová a Sobel. Odezva kvality (měřeno v PSNR) byla v případě všech operátorů logaritmická. Podrobný průběh testovacích videosekvencí pak ukázal na skutečnost, že nejlépe jsou aproximovány snímky s velkým podílem co do intenzity homogenních ploch, velkým podílem relevantních hranových bodů a nízkým šumem. Z tohoto hlediska jsou nejlépe reprezentována videa typu talking head a renderovaná videa, neboť v nich jsou velmi často přítomny právě homogenní plochy a existence rozmazání pohybem je v nich buď malá a nebo se s ní neseťkáme vůbec. O poměru hranových bodů k náhodným nelze vyslovit jednoznačný verdikt s výjimkou poznatku, že kvalita aproximace obecně nejlépe vychází pro nadpoloviční počet hranových bodů. Z hlediska nejlepší navržené bezztrátové komprese se metoda vyplatí při procentu vkládaných bodů menším než 30%. Při srovnání s kodekem M-JPEG bylo patrné, že jde o příliš zdatného konkurenta. Ovšem při srovnání s decimální triangulací vyšla najevo nižší časová složitost vlastního řešení za cenu mírného propadu kvality.

Použití inter kódování ukázalo svoji hlavní přednost v obecně velmi nízkém datovém toku shora omezeném intra kódovanými snímky. Pro interval 5-30% vložených bodů lze dosáhnout kompresního poměru 20:1-4,5:1. Z metrik pro vyhodnocení korespondence dvou bloků byla v testech zvolena metrika MAD, neboť poskytovala nejlepší výsledky a zároveň nejkratší čas zpracování. Pro samotný odhad pohybu se jako optimální se ukázala velikost bloku 16x16px, pro jiné velikosti šla kvalita výrazně dolů. Poměrná časová složitost vyzdvihla odhad a kompenzaci pohybu jako hlavní příčinu zpomalení. Čas zpracování snímku je v případě odhadu pohybu úměrný procentu vložených bodů. V případě kompenzace pohybu je zde patrná závislost na velikosti vektorů pohybu. Aproximace při subjektivním srovnávání vychází lépe pro warping než pro interpolaci z důvodu neexistence trojúhelníkových artefaktů. Podle metriky PSNR je naopak horší, protože zde dochází k mírnému „roztřepení“ aproximovaného obrazu vlivem neexistence převzorkování vstupního obrazu. Výsledky testů kvality byly nejlepší pro dynamickou kameru, kde se kvalita aproximace během celé videosekvence držela dostatečně nad kvalitou nekompensovaného videa. Srovnání kvality s kodekem XviD ukázalo malý rozdíl při velmi nízkém datovém toku. Vlastní metoda může v tomto případě být vhodnou alternativou, když ošetří propad kvality predikovaných snímků.

Kapitola 10 | Závěr

Při dalším vývoji navržené metody v rámci projektu by bylo vhodné se zaměřit pouze na inter kódování a vylepšit kvalitu a časovou složitost predikce snímků. V případě časové složitosti je nejvíce problémovým elementem algoritmus block matching, který pro konkurenceschopnost s komerčními metodami vyžaduje modifikace typu tříkrokové nebo binární vyhledávání. Další úsilí je nutno vynaložit na urychlení kompenzace pohybu posunem DT tak, aby bylo možné v přijatelném čase zpracovávat pohyby řádově desetitisíců bodů získaných z videí se standardním rozlišením. Ohledně kvality je největší prostor k vylepšení na straně odhadu pohybu ve smyslu vyloučení vířivých vektorů, které buď velmi hrubě aproximují optický tok v daném místě anebo ho nerespektují vůbec. K tomuto účelu je vhodné použít buď nějakou dodatečnou optimalizaci v podobě algoritmu podobného UMA [Yaop98], který lokálně minimalizuje chybu vzniklou po interpolaci okolí vyšetřovaného bodu, anebo zavést subpixelovou přesnost jako v případě kodeků rodiny MPEG. Dalším významným faktorem udávajícím míru propadu kvality je kompenzace pohybu. Zde je potřeba přehodnotit stávající přístup s pevným počtem bodů v jedné GOP. Případy, kdy se body nemohou posunout dle odhadnutých vektorů, totiž narůstají spolu s množstvím bodů ve scéně, a proto je na místě zvážení lokálních úprav v místech, kde je pohyb částečně nebo zcela zablokovan. Proměnný počet bodů by také mohl přispět k zohlednění podstatných změn ve scéně, které se udály během doby trvání jedné GOP. Pro tento účel by zřejmě bylo vhodné vylepšení popisu inter kódovaných snímků tak, aby oblasti s nejhorší aproximací byly kódovány intra jako v MPEG. Poslední úpravu si vyžaduje warping snímků, kdy by převzorkování vstupního obrazu znatelně vylepšilo kvalitu pohybově kompenzovaného snímku.

11 Použité zdroje

- [Boug99] Bouguet J. Y.: Pyramidal implementation of the Lucas Kanade Feature Tracker: Description of the Algorithm. Intel Research Laboratory, 1999.
- [Bowy93] Bowyer A., Woodwark J.: Introduction to Computing with Geometry. , 1993, ISBN 1-874728-03-8.
- [Bres65] Bresenham J. E.: Algorithm for Computer Control of a Digital Plotter. IBM Systems Journal, Vol. 4, No. 1, str. 25-30, 1965.
- [Burg00] Burgos B.X.V: Approximation and Geometric Processing of Digital Images with Adaptive Triangular Meshes. Universidad Politécnic de Cataluña, 2000.
- [Cine01] Cinepak (CVID) stream format for AVI and QT[online]. URL: <<http://www.csse.monash.edu.au/%7etimf/videocodec/cinepak.txt>>.
- [Dey06] Dey T.K.: Curve and Surface Reconstruction: Algorithms with Mathematical Analysis. Cambridge University Press, 2006, ISBN 0521863708.
- [Ding95] Ding-Zhu D., Hwang F.: Computing in Euclidian Geometry. World Scientific, 1995, ISBN 9810218761.
- [Fuks06] Fuksa M.: Delaunayova triangulace s omezením (CDT) v E2 a E3. Diplomová práce, Západočeská univerzita v Plzni, 2006.
- [Garc99] García M.A., Vintimilla B. X., Sappa A. D.: Efficient Approximation of Gray-scale Images Through Bounded Error Triangular Meshes. Proceedings of IEEE International Conference on Image Processing, Kobe, Japan, 1999.
- [Hjel06] Hjelle O., Daehlen M.: Triangulations and Applications. Springer, 2006, ISBN 3-540-33260-X.
- [Horn81] Horn B.K.P, Schunck B.G.: Determining Optical Flow. Artificial Inteligence, Vol. 17, No. 1, str. 185-203, 1981.
- [Huff] Huffyuv[online]. URL: <<http://neuron2.net/www.math.berkeley.edu/benrg/huffyuv.html>>.
- [Chen99] Chen C. W., Zhang YA. Q.: Visual Information Representation, Communication, and Image Processing. CRC Press, 1999, ISBN 082471928X.
- [ISO01] ISO/IEC 14496-2:2001. Coding of Audio-Visual Objects: Visual. 2nd ed, 2001.
- [Jack05] Keith J.: Video Demystified: A Handbook for the Digital Engineer, Fourth Edition. Elsevier, 2005, ISBN 0-7506-7822-4.
- [Jana07] Janák T.: Interpolation Methods for Triangulation-Represented Digital Image. CESCg, Bratislava – Budmerice, str. 51-58, 2008.
- [Kell04] Kelly L., Worboys M., Duckham M.: GIS: A Computing Perspective, second ed.. CRC Press, 2004, ISBN 0415283752.
- [Koho07] Kohout J.: On Digital Image Representation by the Delaunay Triangulation. Západočeská univerzita v Plzni, 2007.
- [Krey01] Kreylos O., Hamann B.: On Simulated Annealing and the Construction of Linear Spline Approximations for Scattered Data. IEEE Transactions on Visualization and Computer Graphics, Vol. 7, No. 1, str. 17-31, 2001.
- [Laga] Lagarith Lossless Video Codec[online]. URL: <<http://lags.leetcode.net/codec.html>>.

Kapitola 11 | Použité zdroje

- [Lowe03] Lowe D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, Vol. 20, No. 1, str. 91-110, 2003.
- [Luca81] Lucas B.D., Kanade T.: An Iterative Image Registration Technique with an Application to Stereo Vision. Proceedings of Imaging understanding Workshop, str. 121-130, 1981.
- [Mach] SIFT: Scale Invariant Feature Transform[online]. URL: <<http://cgg.ms.mff.cuni.cz/~pepca/ref/SIFT.pdf>>.
- [Marq00] Marquant G.: Représentation par maillage adaptatif déformable pour la manipulation et la communication d'objets vidéo. Université de Rennes, 2000.
- [Marr80] Marr D., Hildrith E.: Theory of Edge Detection. Proc. Royal Society of London, Vol. 207, str. 187-217, 1980.
- [MSU] MSU Video Quality Measurement Tool[online]. URL: <compression.ru/video/quality_measure/video_measurement_tool_en.html>.
- [OpenCV] OpenCV Library[online]. URL: <<http://opencvlibrary.sourceforge.net/>>.
- [Pras06] Prasad L., Skourikhine A. N.: Vectorized Image Segmentation via Trexel Agglomeration. Pattern Recognition, Vol. 39, str. 501-514, 2006.
- [RFC] RFC 1951 - DEFLATE Compressed Data Format Specification version 1.3[online]. URL: <<http://tools.ietf.org/html/rfc1951>>.
- [Rich03] Richardson I. E. G.: H.264 and MPEG-4 Video Compression. Wiley, 2003, ISBN 0-470-84837-5.
- [Rila98] Rila L.: Image Coding Using Irregular Subsampling and Delaunay Triangulation. Proceedings of SIBGRAPI, 1998.
- [SharCV] SharperCV Project[online]. URL: <<http://www.cs.ru.ac.za/research/groups/SharperCV/>>.
- [Scha06] Schaar M., Turaga D. S., Stockhammer T.: MPEG-4 Beyond Conventional Video Coding. Morgan & Claypool Publishers, 2006, ISBN 1598290428.
- [Srik05] Srikanth R., Ramakrishnan A.G.: Contextual Encoding in Uniform and Adaptive Mesh-Based Lossless Compression of MR Images. IEEE Transactions on Medical Imaging, Vol. 24, No. 9, str. 1199-1206, 2005.
- [Varg07] Varga M.: Využití tetradřizace jako alternativy k objemovým datům. Diplomová práce, Univerzita Karlova v Praze, 2007.
- [VirtDB] VirtualDub[online]. URL: <<http://virtualdub.org/>>.
- [Voma08] Vomáčka T.: Delaunayova triangulace pohybujících se bodů v rovině. Diplomová práce, Západočeská univerzita v Plzni, 2008.
- [Wiki01] Lucas-Kanade method[online]. URL: <http://en.wikipedia.org/wiki/Lucas_Kanade_method>.
- [Xiph] Xiph.org :: Test Media[online]. URL: <<http://media.xiph.org/video/derf/>>.
- [Yaop98] Yaoping Y., Chengke W.: A Novel Video Coding Scheme Using Delaunay Triangulation. Journal of Visual Communication and Image Representation, Vol. 9, No. 1, str. 80-86, 1998.
- [Žára98] Žára J., Beneš B., Felkel P.: Moderní počítačová grafika. Computer Press, 1998, ISBN 80-7226-049-9.

12 Zkratky a pojmy

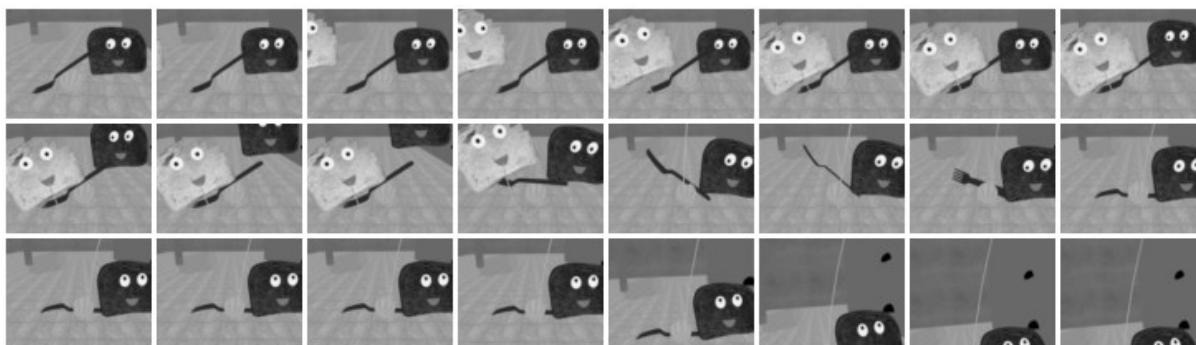
2D	dvourozměrný
3D	třírozměrný
BM	Block Matching; metoda hledající korespondenci mezi bloky v jednotlivých snímcích
CDT	Constrained Delaunay Triangulation; triangulace s omezením
DCT	Discrete Cosine Transformation; diskrétní kosinová transformace
DFD	Displaced Frame Difference
DOG, DoG	Difference of Gaussians; rozdíl Gaussianů
DT	Delaunayova triangulace
DWT	Discrete Wavelet Transform; diskrétní vlnková transformace
fps	frames per second; jednotka snímkové frekvence
GOP, GoP	Group Of Pictures; skupina snímků počínaje a konče intra snímkem
GPL	General Public Licence; všeobecná veřejná licence
KLT	Kanade-Lucas-Thomasi; diferenciální metoda odhadující optický tok
kodek	kompresor-dekompresor
LGPL	Lesser GPL; GPL umožňující spojení s nesvobodným kódem
MAD	Mean Absolute Difference; střední absolutní odchylka
makroblok	sružení více přilehlých bloků (jednotky) do jednoho
MPEG	Motion Pictures Expert Group
MR	Magnetic Resonance
MSD	Mean Square Difference, střední kvadratická odchylka
MSE	Mean Square Error
NTSC	National Television System(s) Comitee
PAL	Phase Alternating Line
PDC	Pel Difference Classification
PSNR	Peak Signal to Noise Ratio; maximální poměr signálu k šumu
QXGA	Quad eXtended Graphics Array
RGB	Red, Green, Blue
RLE	Run Length Encoding
RMSE	Root Mean Square Error
SDTV	Standard Definition Television
SIF	Source Input Format
SIFT	Scale Invariant Feature Transform
SSE	Sum of Squared Errors
SVCD	Super VideoCD
SVGA	Super Video Graphics Array
SXGA	Super eXtended Graphics Array
UMA	Umbrella Matching Algorithm
UXGA	Ultra eXtended Graphics Array
VD	Voronoi diagram
VGA	Video Graphics Array
videosekvence	sled souvisejících snímků
VO	Video Object
VOL	Video Object Layer
VOP	Video Object Plane
XGA	eXtended Graphics Array

Příloha A

Testovací videosekvence



Obr. A.1: Náhled původního videa “Foreman”

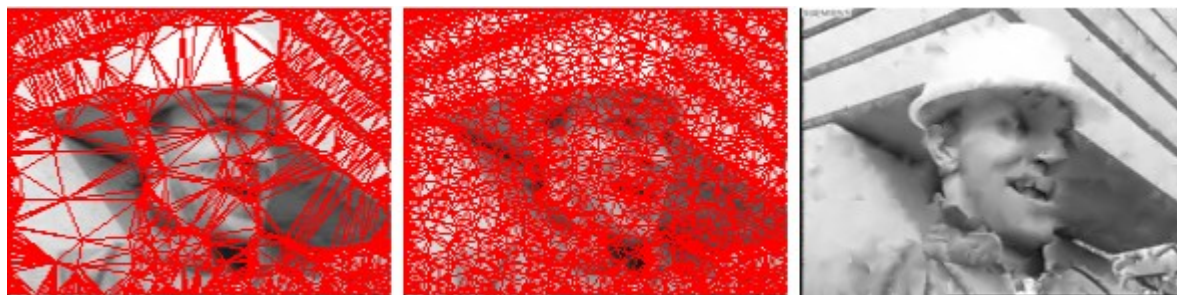


Obr. A.2: Náhled původního videa “Toasts”

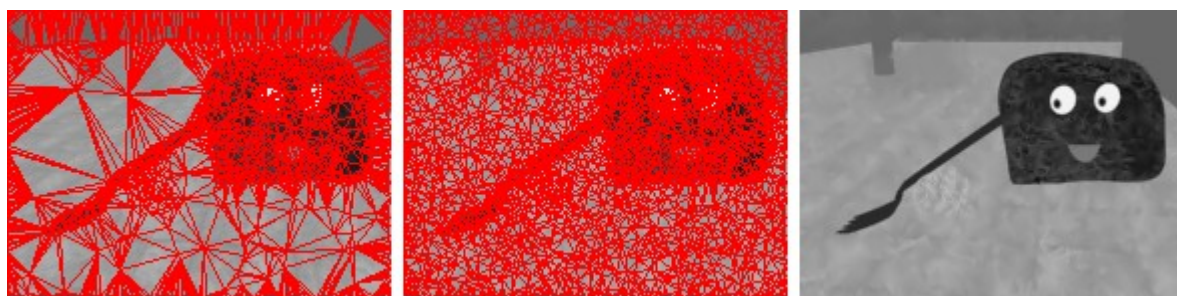


Obr. A.3: Náhled původního videa “Miss America”

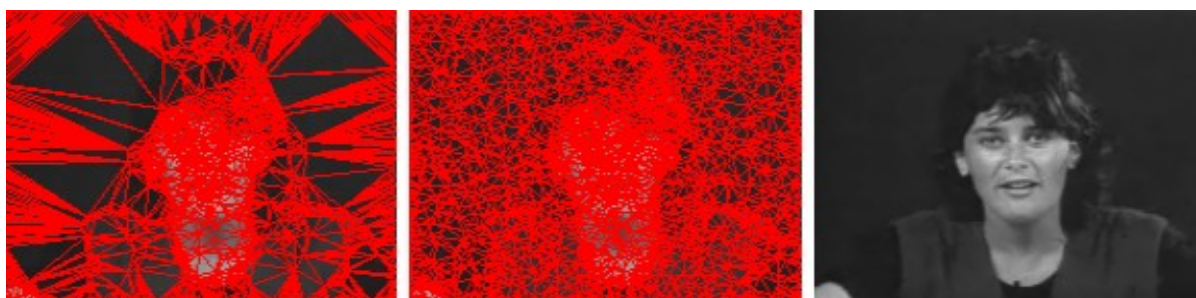
Ukázky reprezentace triangulacemi pro vybrané I-snímky



Obr. A.4: Zpracování intra kódovaného snímku videa “Foreman” velikého 176x144px pro 20% bodů (Zleva: body získané Laplaceovým operátorem, přidané náhodné body, aproximace po interpolaci sítě)



Obr. A.5: Zpracování intra kódovaného snímku “Toasts” velikého 192x144px pro 20% bodů (Zleva: body získané Laplaceovým operátorem, přidané náhodné body, aproximace po interpolaci sítě)



Obr. A.6: Zpracování intra kódovaného snímku videa “Miss America” velikého 176x144px pro 20% bodů (Zleva: body získané Laplaceovým operátorem, přidané náhodné body, aproximace po interpolaci sítě)

Porovnání interpolace a warpingu pro vybrané GOP



Obr. A.7: Interpolace snímků videa Foreman pro 1. GOP (snímky 1-6)



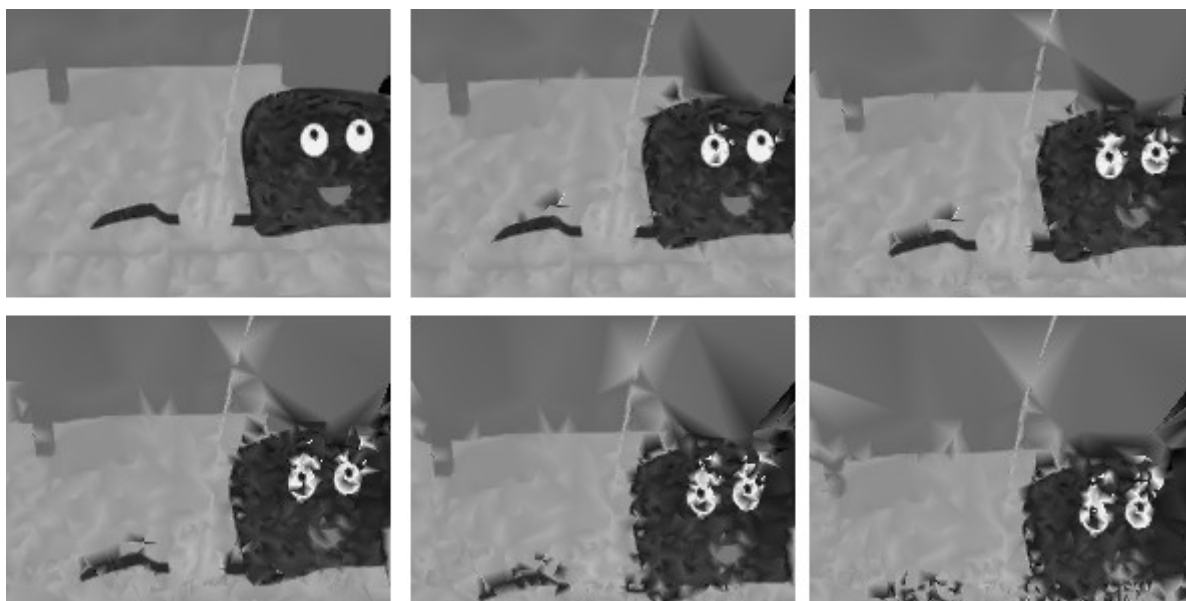
Obr. A.8: Warping snímků videa Foreman pro 1. GOP (snímky 1-6)



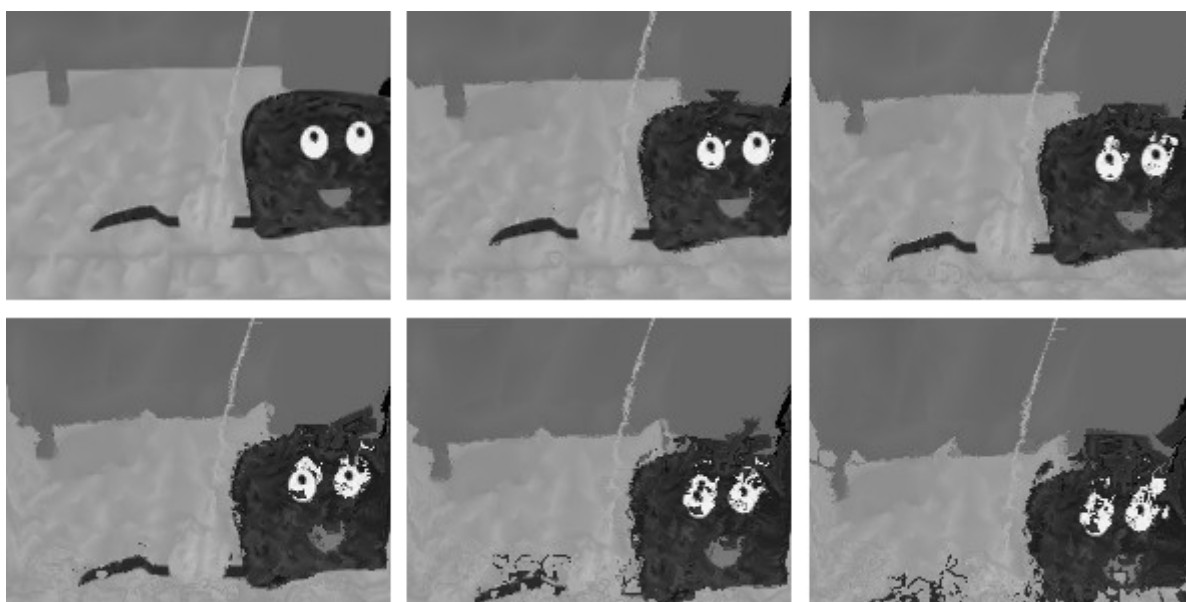
Obr. A.9: Interpolace snímků videa Miss America pro 15. GOP (snímky 85-90)



Obr. A.10: Warping snímků videa Miss America pro 15. GOP (snímky 85-90)



Obr. A.11: Interpolace snímků videa Toasts pro 16. GOP (snímky 91-96)



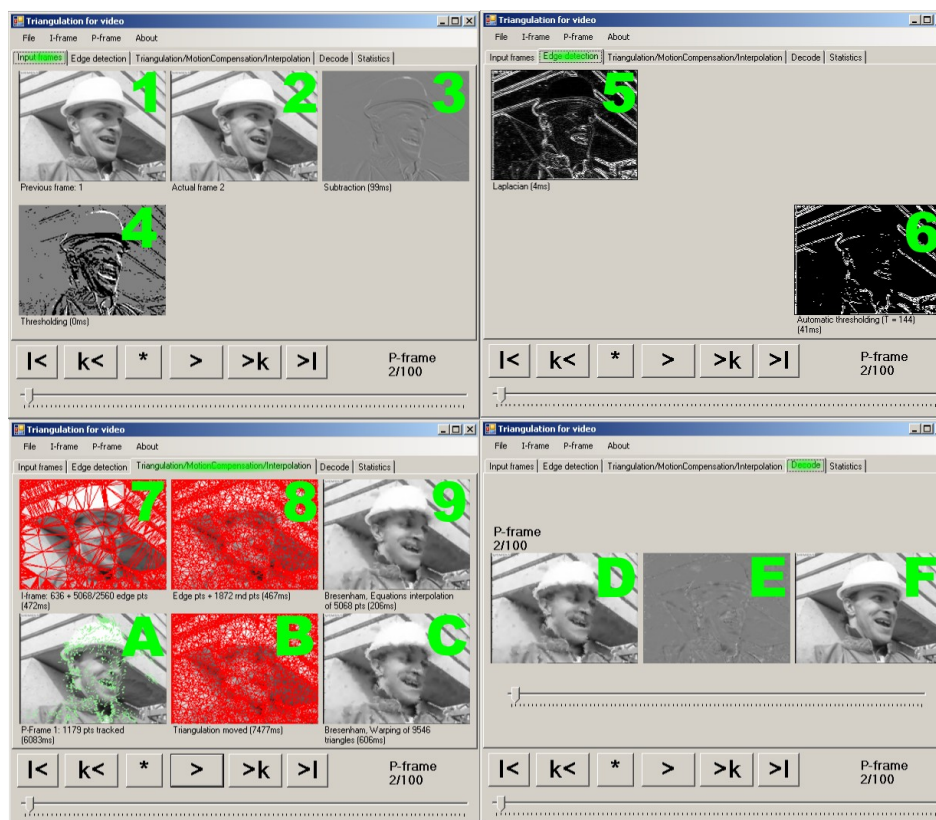
Obr. A.12: Warping snímků videa Toasts pro 16. GOP (snímky 91-96)

Příloha B

Účelem použití aplikace TriCompress je otestování navržené metody ve smyslu funkčnosti kodéru i dekodéru. V jednu chvíli lze v aplikaci zpracovávat pouze jedno video, které se při pokusu o práci s jiným anebo po zadání *Menu>File>Close session* automaticky z programu odstraní. Mezi základní vstupně výstupní schopnosti aplikace patří načítání/ukládání videosouborů, načítání/ukládání sekvence statických snímků a načítání/ukládání sekvence reprezentované geometricky. Speciální vlastností je ukládání jednotlivých snímků bez ohledu na jejich typ jako intra kódované soubory reprezentované geometricky s možností uložení topologie. Vstupem mohou být libovolně velké šedotónové snímky nebo video. V případě barevných vstupů se z podstaty metody zatím pracuje pouze s jedním kanálem.

Načítání a zpracování videosouboru nebo sekvence statických snímků

Kliknutím na *Menu>File>Open AVI...* nebo *Menu>File>Open image sequence...* zadáme cestu k libovolnému souboru s videem ve formátu *.avi anebo k adresáři, který obsahuje soubory formátu *.bmp alfanumericky seřazené podle pořadí v sekvenci. Následně se provedou operace zobrazené na obr. B.1, kde se na jednotlivá označení budeme dále odkazovat. Vyznačená okénka jdou v aplikaci po kliknutí přímo na ně zvětšit.



Obr. B.1: Zpracování videa v aplikaci TriCompress

V panelu *Input frames* se zobrazí předchozí (1) a aktuální (2) snímek načtené sekvence, následuje jejich vzájemný rozdíl (3) a jeho prahování podle dvou prahů.

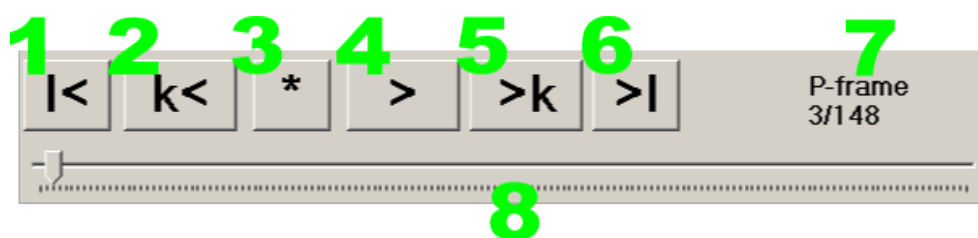
V panelu *Edge detection* se při zapnuté hranové detekci provede v (5) příslušná hranová detekce nad (2) a automaticky prahovaný obraz se zobrazí v (6). Pro různé hranové detektory může být vyžadováno více kroků, které se popřípadě zobrazí v prostoru mezi (5) a (6).

V panelu *Triangulation/MotionCompensation/Interpolation* se v (7,8,9) zobrazuje postup vedoucí k výpočtu intra kódovaného snímku a v (A,B,C) postup tvorby inter kódovaného snímku. V (7) se na základě (2) a (6) postaví Delaunayova triangulace nad vybranými hranovými body. V (8) se podle poměru hranových bodů k náhodným doplní triangulace v (7) o náhodné body, pokud není zvolena tvorba DT decimací. Podle zvolené metody vykreslení se v (9) zkonstruuje intra snímek. Má-li být zpracováván snímek inter kódován, pak je podle výběru metody odhadu pohybu v (A) vytvořen snímek ilustrující odhadnuté vektory pohybu mezi snímky (1) a (2). Kompenzace pohybu posunutím DT je znázorněna v (B) a warpovaný intra snímek (9) anebo interpolovaná data z vrcholů jsou zobrazena v (C).

V panelu *Decode* se nachází dekodér, který je v režimu kódování k dispozici pro náhled již zpracovaných snímků a pro srovnání se vstupní videosekvencí. V (D) je kóděrem zpracovaný snímek, v (F) je příslušný vstupní snímek a v (E) je jejich residuum vyjadřující chybu aproximace. Za pomoci slideru, který je v tomto panelu k dispozici, je možno interaktivně procházet snímky, a to i během komprese.

Posun a zpracování videosekvence

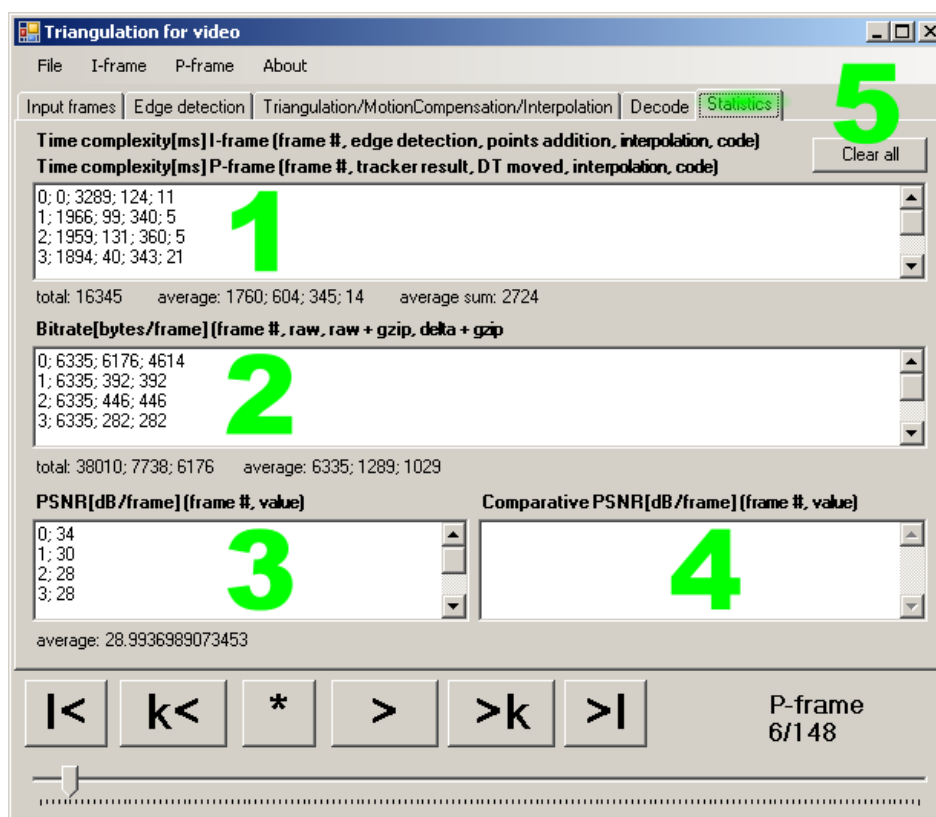
Pro potřeby posunu a zpracování obsahuje TriCompress hned několik ovládacích prvků (viz obr. B.2). Tlačítkem (1) se dostaneme na začátek videosekvence a tlačítkem (6) na její konec. Tlačítka (2) a (5) se přemístíme na jeden z okolních intra kódovaných snímků. Tlačítkem (4) se přemístíme na následující snímek bez rozlišení jeho typu. K automatickému zpracování celé videosekvence počínaje aktuálním snímkem slouží tlačítko (3). Toto tlačítko lze kdykoliv „odmáčknout“ a skončit automatické zpracovávání ihned po dokončení současného snímku. Zde je nutno poznamenat, že všechny výpočty nad snímky probíhají v samostatném vlákne, přičemž lze aplikaci během zpracování snímků používat pouze k náhledu. V opačném případě není stabilita zaručena. Mezi snímky se lze rovněž pohybovat prostřednictvím slideru (8), který nás ovšem pustí na jakýkoliv snímek bez respektování toho, zda je predikovaný či nikoliv. Proto je doporučeno po tažení posuvníku přeskočit na nejbližší intra snímek tlačítky (2,5). Pro orientaci ve videosekvenci slouží informační návěští (7), které zobrazuje zařazení aktuálního snímku mezi ostatními a také typ jeho kódování.



Obr. B.2: Ovládací prvky aplikace TriCompress

Zkoumání statistických údajů o kódování

Statistické údaje o kódování lze v aplikaci zobrazit kliknutím na panel *Statistics* (viz obr. B.3). V seznamu (1) se zobrazují informace o dílčích časech časové složitosti pro I-snímek vzniklý hranovou detekcí ve formátu [číslo snímku, hranová detekce, vkládání bodů, interpolace, kódování]. Pro I-snímek vzniklý decimací je to [číslo snímku, 0, decimace, interpolace, kódování]. Pro P-snímek je formát takovýto: [číslo snímku, odhad pohybu, posun DT, interpolace, kódování]. Pod seznamem (1) jsou informace o celkovém čase všech snímků (*total*), průměrném čase každé složky (*average*) a průměrném čase na jeden snímek (*average sum*). V seznamu (2) jsou údaje o velikosti bezztrátově komprimované triangulace ve formátu: [číslo snímku, datový tok RAW, datový tok RAW+GZIP, datový tok DELTA+GZIP]. Pod seznamem (2) jsou informace o celkovém datovém toku všech dosud zpracovaných snímků ze všech tří metod (*total*) a jejich průměrné hodnoty (*average*). V seznamu (3) se nachází PSNR pro každý zpracovaný snímek vzniklá porovnáním s odpovídajícím snímkem ve vstupní videosekvenci. Pod seznamem (3) je její průměrná hodnota (*average*). Seznam (4) slouží pro porovnání načtené vstupní videosekvence s nějakou další videosekvencí o stejných parametrech, která se nachází ve formátu *.avi a je mimo program. Po načtení takové další videosekvence přes *Menu>File>Compare with AVI...* dojde k porovnání všech snímků a jejich PSNR je vypsána právě do (4). Pro výmaz všech naměřených údajů během práce s videosekvencí slouží tlačítko (5). Nutno dodat, že všechny zmíněné statistiky jsou řešeny formou tabulky, kdy při opakovaném zpracování snímku jsou pro něj záznamy přepočteny.



Obr. B.3: Záložka *Statistics* v aplikaci TriCompress

Parametry intra kódovaných snímků

V souvislosti s intra kódovanými snímky můžeme v *Menu>I-frame* měnit jejich parametry, které se projeví hned při následujícím zpracování I-snímku. Velikost GOP je možno určit v *I-Frame>minimal appearance period*. Číselně nebo procentuelně vyjádřený počet vzorků, přičemž dochází k automatickému přepočtu protějšku, je možno volit v *I-Frame>maximum added samples #* nebo *I-Frame>maximum added samples [%]*. Volba, zda interpolovat pomocí barycentrických souřadnic nebo se soustavou rovnic, je v *I-Frame>interpolation*. Nejrozsáhlejším podmenu voleb parametrů I-snímků je volba metody výběru relevantních bodů v *I-Frame>method*. Zde máme na výběr mezi metodami generickými v podobě náhodného výběru a uniformní mřížky. Nutno poznamenat, že u uniformní mřížky nelze změnit krok, neboť po celou dobu návrhu sloužila pouze jako pomůcka. Další skupinou metod jsou hranové detektory, které kromě možnosti výběru jednotlivých typů nabízejí i možnost stanovit poměr hranových bodů k náhodným v *edge detection>edge/rnd ratio [%]*. Poslední skupinou metod jsou decimace v *I-Frame>method>decimation*.

Parametry inter kódovaných snímků

U inter kódovaných snímků v *Menu>P-Frame* máme na výběr ohledně metody odhadu pohybu formou block matchingu nebo KLT v *P-Frame>method*. Jako parametr těchto metod je možné volit velikost bloku v *P-Frame>kernel size*. Výběrem typu vykreslení v *P-Frame>interpolation* řekneme programu, zda má využít klasické interpolace za pomoci barycentrických souřadnic či soustavy rovnic anebo zda se má zaměřit na předcházející I-snímek a pokusit se jeho obraz warpovat podle odhadu pohybu v tomto a předcházejících I-snímecích. Specifikum metody block matching jsou dva parametry, které se využijí při hledání blokové korespondence. Velikost vyhledávacího okénka v *P-Frame>search window size* zodpovídá za velikost prohledávaného intervalu. Metriky MSD, MAD a PDC v *P-frame>metric* zase udávají jakým způsobem se budou dva bloky vzájemně odlišovat.

Načítání geometricky reprezentované videosekvence

Máme-li v úmyslu dekodovat a zobrazit data, která už program jednou zakódoval, pak potvrdíme položku *Menu>File>Open text sequence...* a vybereme složku, kde se nacházejí textové soubory s příponou *.dat ve formátu [xxxx-i.dat] pro I-snímky a [xxxx-p.dat] pro P-snímky, kde maska xxxx značí čísla v dekadické soustavě zleva dovyplněné nulami tak, aby velikost takového řetězce byla právě 4. Po načtení a dekodování, které neběží ve vlastním vlákně, se dekodovaná sekvence zobrazí v panelu *Decode* v obrazu (D) (viz obr. B.1). V tomto režimu je umožněno pouze scrolování přílehlým sliderem. Ostatní akce mohou vést k pádu aplikace.

Příloha C

Příložené CD v kořenovém adresáři obsahuje:

- text diplomové práce ve formátu PDF (Portable Document Format)
./text-dp/Puncman-dp-2008.pdf
- editovatelný text diplomové práce ve formátu ODT (OpenDocument Text)
./text-dp/Puncman-dp-2008.odt
- OpenOffice.org 2.3.1 (Portable ed.), licence LGPL
./OpenOfficePortable/OpenOfficePortable.exe
- zdrojový kód aplikace TriCompress jako projekt pro prostředí SharpDevelop
./TriCompress/TriCompress.sln
- SharpDevelop 3.0.0, licence LGPL
./SharpDevelop/bin/SharpDevelop.exe
- sestavenou aplikaci TriCompress
./TriCompress/bin/Debug/TriCompress.exe
- volně šířitelná videa
./TriCompress/bin/Debug/data/videos
- volně šířitelné sekvence statických snímků
./TriCompress/bin/Debug/data/im-sequences
- geometricky reprezentované videosekvence
./TriCompress/bin/Debug/data/geom-sequences

V rámci prvních pokusů s aplikací je nejvhodnější se nejprve zaměřit na načtení libovolné z testovacích videosekvencí a otestovat intra i intra+inter kódování. Návod, jak dosáhnout načtení videosekvence Foreman a na co se při tom zaměřit, nalezneme přímo v aplikaci TriCompress v *Menu>About>Tutorial 1*. Další test může být spjat s dekodováním geometricky reprezentované videosekvence Suzie popsáním v *Menu>About>Tutorial 2*.