



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

NÁSTROJE PRO MĚŘENÍ KVALITY VIDEOSEKVENČÍ BEZ REFERENCE

TOOLS FOR MEASURING VIDEO QUALITY WITHOUT REFERENCE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. ONDŘEJ ZACH

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MARTIN SLANINA, Ph.D.

BRNO 2013



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Diplomová práce

magisterský navazující studijní obor
Elektronika a sdělovací technika

Student: Bc. Ondřej Zach

ID: 119679

Ročník: 2

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Nástroje pro měření kvality videosekvencí bez reference

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s metodami, používanými pro objektivní měření kvality videosekvencí bez možnosti porovnání s originálem. Srovnajte principy, předpoklady a náročnost implementace jednotlivých algoritmů. Uvažujte videosekvence různých rozlišení a snímkových rychlostí. Navrhněte grafické rozhraní programu, do kterého budete přidávat algoritmy pro bezreferenční měření kvality formou přídatných modulů. Implementujte několik jednoduchých kvalitativních metrik. S využitím navrženého programu dále proveďte měření kvality na souboru videosekvencí. Dosažené výsledky srovnajte s výsledky subjektivních testů.

DOPORUČENÁ LITERATURA:

[1] UZAIR, M., FAYEK, D. An efficient no-reference blockiness metric for intra-coded video frames. In 14th International Symposium on Wireless Personal Multimedia Communications. Brest (France), 2011, p. 1-5.

[2] LIU, H., WANG, J., REDİ, J., LE CALLET, P., HEYNDERICKX, I. An efficient no-reference metric for perceived blur. In 3rd European Workshop on Visual Information Processing (EUVIP). Paris (France), 2011, p. 174 - 179.

Termín zadání: 11.2.2013

Termín odevzdání: 24.5.2013

Vedoucí práce: Ing. Martin Slanina, Ph.D.

Konzultanti diplomové práce:

prof. Dr. Ing. Zbyněk Raida

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Výzkum realizovaný v rámci této diplomové práce byl finančně podpořen projektem
CZ.1.07/2.3.00/20.0007 **Wireless Communication Teams**
operačního programu **Vzdělávání pro konkurenceschopnost**.



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Finanční podpora byla poskytnuta Evropským sociálním fondem
a státním rozpočtem České republiky.

ABSTRAKT

Diplomová práce se zabývá objektivními metrikami pro hodnocení kvality videa bez reference. Jsou zde krátce popsány všeobecné základy hodnocení kvality videosekvencí, dále jsou uvedeny základní předpoklady objektivních metrik. Hlavním zaměřením práce jsou no-reference přístupy. Práce se snaží popsat základní metody pro hledání rušení ve videosekvencích. Je uveden rozdíl mezi metrikami pracujícími v prostorové a spektrální oblasti. Dále je popsán návrh aplikace v prostředí Matlab pro objektivní testy kvality videosekvencí. Byla navržena a implementována metrika pro odhad PSNR pro H.264 kódovaná videa na základě informací z bitové toku. Závěrem jsme sestavili databázi videosekvencí a provedli objektivní testy, které byly srovnány s výsledky testů subjektivních.

KLÍČOVÁ SLOVA

kvalita videa, objektivní hodnocení, no-reference, NR, PSNR, metrika pro hodnocení kvality, GBIM, Matlab, NR VQT, H.264

ABSTRACT

This diploma thesis deals with objective video quality assessments without reference. Some of the basics of video quality evaluation are described. Also some basic conditions for objective video quality metrics are introduced. The main focus of the thesis are no-reference approaches. Thesis tries to describe basic methods for seeking distortion in video. The difference between spatial-domain oriented and spectral-domain oriented metrics is analyzed. We also describe design of tool for measuring objective video quality in Matlab environment. We then designed and implemented a bit-stream oriented metric for estimation of the PSNR of H.264 coded sequences. Finally, we created a database of video sequences and we held objective tests. Results were compared with results from subjective measurements.

KEYWORDS

video quality, objective assessment, no-reference, NR, PSNR, quality metric, GBIM, Matlab, NR VQT, H.264

ZACH, O. *Nástroje pro měření kvality videosekvencí bez reference*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 50 s., 6 příloh. Vedoucí diplomové práce Ing. Martin Slanina, Ph.D..

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma Nástroje pro měření kvality videosekvencí bez reference jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne 24. května 2013

.....

(podpis autora)

PODĚKOVÁNÍ

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

V Brně dne 24. května 2013

.....

(podpis autora)

OBSAH

Seznam obrázků	viii
Seznam tabulek	ix
Úvod	1
1 Hodnocení kvality videosekvencí	2
1.1 Subjektivní hodnocení	2
1.1.1 Metody s použitím dvou sekvencí	2
1.1.2 Metoda s použitím jedné sekvence	4
1.2 Objektivní hodnocení.....	4
1.2.1 Metody s plnou referencí	4
1.2.2 Metody s částečnou referencí	5
1.2.3 Metody bez reference.....	5
2 Metriky založené na bezreferenčním přístupu	7
2.1 Metriky pro blokové artefakty	7
2.2 Metriky pro měření rozmazání obrazu.....	8
2.3 Srovnání náročnosti implementace	8
2.3.1 Generalized Block-Edge Impairment Metric (GBIM).....	9
2.3.2 No-reference Perceptual Blur Metric (Marziliano)	10
2.3.3 Odhad PSNR na základě koeficientů DCT.....	11
3 Popis vytvořené aplikace No-Reference Video Quality Tool	13
3.1 Grafické rozhraní	14
3.2 Ověření typu souboru a zjištění informací o videu.....	15
3.3 Volba počtu snímků, získání jasového signálu.....	16
3.4 Použité metriky a zobrazení výsledků	19
3.4.1 Implementace GBIM	19
3.4.2 Implementace NR Perceptual Blur Metric – Marziliano	20
3.4.3 Průběh výpočtu a zobrazení výsledků	21
3.5 Optimalizace a výkon aplikace	23
4 Odhad PSNR pro H.264 kódované sekvence	25
4.1 H.264/AVC	25

4.2	Získání informací z bitového toku	25
4.3	Úprava dekodéru JEG JM H.264/AVC	26
4.4	Návrh algoritmu pro odhad PSNR.....	27
4.4.1	Sestavení algoritmu.....	28
4.5	Ověření a ladění algoritmu	29
4.6	Implementace navržené metriky v prostředí Matlab	30
5	Databáze videosekvencí	33
5.1	Databáze VQEG	33
5.2	Databáze LIVE	34
5.3	Výsledná databáze sekvencí	35
6	Objektivní testy	36
6.1	Pixelově-orientované metriky, databáze LIVE.....	36
6.2	Pixelově-orientované metriky, databáze VQEG.....	40
6.3	Odhad PSNR H.264 kódovaných sekvencí	44
6.4	Srovnání přístupů temporal pooling	45
6.5	Shrnutí.....	48
	Závěr	50
	Literatura	51
	Seznam symbolů, veličin a zkratk	54
	Seznam příloh	55

SEZNAM OBRÁZKŮ

Obr. 1	Struktura testu Double Stimulus [1]	2
Obr. 2	Stupnice vnímaného rušení pro metodu DSIS [1]	3
Obr. 3	Stupnice rušení pro metodu DSCQE, [3]	3
Obr. 4	Váhovací funkce pro metriku GBIM, [11]	9
Obr. 5	Vývojový diagram algoritmu Marziliano, [9]	10
Obr. 6	Diagram aplikace NR VQT	13
Obr. 7	Hlavní okno programu NR VQT	14
Obr. 8	Hláška při volbě neplatného souboru	15
Obr. 9	Příklad chybového hlášení při zadání počtu snímků	16
Obr. 10	Hlášení o úspěšném načtení souboru	18
Obr. 11	Ověření metriky Marziliano, testovací sekvence a hodnoty kvality	20
Obr. 12	Zobrazení ukazatele průběhu výpočtu	21
Obr. 13	Zobrazení výsledků	23
Obr. 14	Srovnání skutečného a odhadnutého PSNR, ověření algoritmu	30
Obr. 15	Schéma použitých prostředků pro odhad PSNR	30
Obr. 16	Grafické rozhraní NR VQT, změny pro H.264 metriku	31
Obr. 17	Zobrazení výsledků metriky pro odhad PSNR H.264 kódovaných sekvencí	32
Obr. 18	Srovnání GBIM a DMOS, databáze LIVE, sekvence (pa), (rb) a (rh)	37
Obr. 19	Srovnání Marziliano a DMOS, databáze LIVE, sekvence (pa), (rb) a (rh)	37
Obr. 20	Srovnání GBIM a DMOS, databáze LIVE, sekvence (sf), (st) a (tr)	38
Obr. 21	Srovnání Marziliano a DMOS, databáze LIVE, sekvence (sf), (st) a (tr)	38
Obr. 22	Srovnání GBIM a DMOS, databáze LIVE, sekvence (pr) a (sh)	39
Obr. 23	Srovnání Marziliano a DMOS, databáze LIVE, sekvence (pr) a (sh)	39
Obr. 24	Srovnání hodnoty GBIM v závislosti na HRC, databáze VQEG	41
Obr. 25	Detaily některých sekvencí, databáze VQEG	41
Obr. 26	Srovnání hodnoty Marziliano v závislosti na HRC, databáze VQEG	42
Obr. 27	Srovnání DMOS v závislosti na HRC, databáze VQEG	43
Obr. 28	Srovnání skutečného a odhadnutého PSNR, databáze LIVE	44
Obr. 29	Zobrazení dalších výsledků, temporal pooling	46
Obr. 30	Srovnání TP přístupů, metrika GBIM	47
Obr. 31	Srovnání TP přístupů, metrika Marziliano	48

SEZNAM TABULEK

Tab. 1 Vliv úprav zdrojového kódu na rychlost běhu dekodéru ldecod.....	27
Tab. 2 Metoda výběru proměnných pro různé intervaly QP	29
Tab. 3 Popis vybraných HRC databáze VQEG, [32]	33
Tab. 4 Popis vybraných sekvencí VQEG, [32].....	34
Tab. 5 Popis vybraných sekvencí databáze LIVE, [30].....	34
Tab. 6 Hodnoty kvality, databáze LIVE, sekvence (pa), (rb) a (rh)	37
Tab. 7 Hodnoty kvality, databáze LIVE, sekvence (sf), (st) a (tr).....	38
Tab. 8 Hodnoty kvality, databáze LIVE, sekvence (pr) a (sh)	39
Tab. 9 Hodnoty metriky GBIM v závislosti na HRC, databáze VQEG	40
Tab. 10 Hodnoty metriky Marziliano v závislosti na HRC, databáze VQEG.....	42
Tab. 11 Hodnoty DMOS v závislosti na HRC, databáze VQEG	43
Tab. 12 Skutečná a odhadnutá PSNR, databáze LIVE	44
Tab. 13 Relativní chyba odhadu PSNR	45

ÚVOD

V současné moderní době plné multimediálních aplikací a vzrůstajícího objemu různých video přenosů, ať již jde o videozáznamy, živě vysílání či např. i videotelefonie, se stále více projevuje požadavek na možnost, jak snadno, rychle a co nejpřesněji zjistit obrazovou kvalitu dané videosekvence. Možností, jak měřit kvalitu videosekvencí, je několik. Dostupné metody lze rozdělit na dvě hlavní skupiny: subjektivní a objektivní. Zatímco metody subjektivní jsou založeny na reálném vnímání kvality sekvence skutečným pozorovatelem, metody objektivní se snaží odstranit tento článek a odhadují kvalitu pouze na základě jistých vzorců, vztahů a očekávaných vlastností obrazu. Subjektivní metody byly důležité zejména v minulosti, kdy se na základě výsledků např. definovaly parametry digitalizace obrazových dat. V současné době je kladen důraz na automatizaci procesu hodnocení obrazu a je snaha o co nejmenší zásah člověka do celého procesu. Proto jsou objektivní metricky žhavým tématem posledního desetiletí.

Objektivní metody lze dále rozdělit na metody založené na srovnání kvality s referenčním snímkem či metody bez použití reference. Na rozhraní mezi těmito dvěma skupinami stojí metody tzv. Reduced Reference, kdy máme jen jisté omezené informace o zkoumaném obraze. Tato práce se bude zabývat metodami objektivními bez použití referenčních snímků. Ty mají tu výhodu, že nepotřebují nic vědět o zpracovávaném snímku; povědomí o tom, co je či není rušením si vytvoří samy na základě jistých definovaných postupů. To však klade velké požadavky na samotný algoritmus metricky, který by měl umět odhadnout kvalitu snímku jak by ji vnímal člověk.

V první kapitole práce jsou krátce rozebrány subjektivní metody hodnocení kvality obrazu. Na jejich základě jsou poté vysvětleny některé základní principy metod objektivních. Je zde popsán rozdíl mezi metrikami typu full-reference, reduced-reference a no-reference.

Druhá kapitola se již detailně věnuje no-reference přístupu. Jsou rozebrány metody používané k detekci jednotlivých typů rušení. Dále jsou rozebrány přístupy pixelově-orientované a založené na analýze bitového toku. Na základě toho jsou srovnány nároky na implementaci.

Třetí kapitola pojednává o samotném vývoji aplikace pro objektivní bezreferenční hodnocení kvality videosekvencí. Popisuje jednotlivé části programu s názvem No-Reference Video Quality Tool, který byl napsán v prostředí Matlab. Program byl testován pro použití s různými sekvencemi a s různými verzemi prostředí Matlab.

Kapitola čtvrtá se zabývá návrhem algoritmu pro odhad PSNR sekvencí kódovaných kodekem H.264 na základě informací z bitového toku. Metrika byla dále implementována do již vytvořené aplikace.

Pátá kapitola objasňuje sestavení databáze videosekvencí pro potřeby následných objektivních testů. Byly vybrány sekvence kódované pomocí MPEG-2 a H.264. Následující kapitola poté pojednává o výsledcích těchto objektivních testů a srovnání s testy subjektivními.

1 HODNOCENÍ KVALITY VIDEOSEKVENČÍ

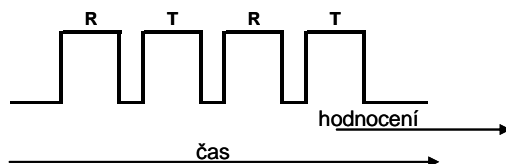
Pro zjišťování kvality obrazu či videa lze užít dvou přístupů, objektivních či subjektivních. Zatímco subjektivní metriky se spoléhají na reálné pozorovatele a jejich hodnocení, metriky objektivní jsou založeny pouze na výpočtech jistých parametrů obrazu na základě předem definovaných vztahů. Oba tyto přístupy mohou mít své výhody i nevýhody, z časových, finančních a dalších důvodů se však stále více uplatňují metriky objektivní. V následující kapitole si ve zkratce popíšeme subjektivní přístup a rozebereme možnosti, které nabízejí metriky objektivní.

1.1 Subjektivní hodnocení

Jak bylo již zmíněno, při subjektivním hodnocení kvality obrazu vyhodnocujeme vliv scény na reálného pozorovatele. Už z principu jsou tedy tyto metody značně časově náročné a nelze je užít v reálném čase. Základem je vybraný panel diváků, kteří na základě předem definovaných požadavků, které jsou dané konkrétní metodou, subjektivně vyhodnocují vnímanou kvalitu obrazu. Výsledky jsou poté statisticky zpracovány. Velmi důležitou roli hraje i samotné složení panelu. Rozdílných výsledků lze například dosáhnout, je-li panel složen z odborníků nebo z laiků. Odborník může v obraze cíleně hledat známé projevy rušení, a proto výsledky takové měření jsou spíše vhodné pro hodnocení např. kompresních metod. Panel složený z neodborníků naopak poskytne velmi dobrý výsledek co se všeobecného vjemu týče. Mezinárodně přijímaných metod pro subjektivní hodnocení kvality obrazu je několik a lze je nalézt např. v doporučení ITU-R BT. 500 pro hodnocení kvality SDTV, či v doporučení ITU-R BT. 710 pro HDTV a dále např. v doporučení ITU-T P.910 a příbuzných. Hlavními metodami jsou Double-stimulus impairment scale (DSIS), Double-stimulus continuous quality evaluation (DSCQE) a Single stimulus continuous quality evaluation (SSCQE) [1],[2].

1.1.1 Metody s použitím dvou sekvencí

Metody Double Stimulus (DSIS a DSCQE) jsou založeny na tom, že pozorovateli je promítán referenční i testovaný snímek, přičemž je známo, který snímek je který. Strukturu testu zobrazuje obrázek 1.



Obr. 1 Struktura testu Double Stimulus [1]

Z obrázku je patrné, že referenční (R) a testovaný (T) snímek jsou promítány dvakrát po

sobě a to po stejný čas (obvykle 10 s). Mezi nimi je promítán šedý obraz po dobu přibližně 3 vteřiny. Dále se metriky DSIS a DSCQE liší ve stylu, kterým respondent kvalitu obrazu hodnotí [1],[2].

Double Stimulus Impairment Scale

Metoda DSIS je založena na tom, že hodnotící subjekt hodnotí vnímanou kvalitu videa na pětibodové stupnici. Stupnice zahrnuje stavy, které mohou při vnímání kvality nastat. Pozorovatel si musí vybrat mezi nimi, nelze tedy volit hodnotu, která by ležela mezi dvěma danými úrovněmi kvality. Jak může vypadat stupnice pro hodnocení DSIS je zobrazeno na následujícím obrázku.

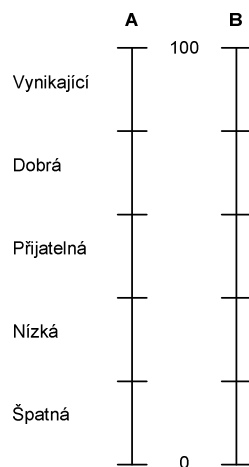
Nepozorovatelné	<input type="checkbox"/>
Viditelné, ne však obtěžující	<input type="checkbox"/>
Mírně obtěžující	<input type="checkbox"/>
Obtěžující	<input type="checkbox"/>
Velmi obtěžující	<input type="checkbox"/>

Obr. 2 Stupnice vnímaného rušení pro metodu DSIS [1]

Zde zobrazená stupnice je mezinárodně uznávaným standardem pro tuto metodu. Nejčastějšího použití nalezne tato metoda v situaci, kdy měníme nějaké parametry obrazu a zajímá nás, kdy nastane zlom ve vnímání kvality. Tedy například měníme-li datový tok videosekvence a zajímá nás, kdy budou případné artefakty rušivé[1].

Double Stimulus Continuous Quality Evaluation

Metoda DSCQE nám naopak dává možnost měřit kvalitu pozorovanou subjektem na kontinuální stupnici. Stupnice pro tuto metodu je zobrazena na obrázku 3, [3].



Obr. 3 Stupnice rušení pro metodu DSCQE, [3]

Respondent tedy určuje kvalitu na stupnici od 0 do 100. Místo referenčního a testovacího snímku zde máme obraz A a B, přičemž nemusí být jasně daný referenční snímek [1], [8].

1.1.2 Metoda s použitím jedné sekvence

Metoda Single Stimulus Continuous Quality Evaluation (SSCQE) je založena na tom, že pozorovateli je kontinuálně promítán obraz, jehož kvalita se mění s časem. Subjekt je vybaven elektronickým zařízením, pomocí něhož určuje vnímanou kvalitu obrazu, přičemž stupnice pro určení kvality je tatáž jako v případě metody DSCQE. Pro naši práci je na této metodě zajímavé to, že metoda nevyžaduje referenční snímek. Pořadatel testu může sice předem respondentům poskytnout referenční obraz, není to ale nutné. Referenční obraz si pak vytvoří každý pozorovatel sám na základě vlastního vnímání obrazu. Velkou výhodou této metody je právě kontinuita měření, lze tedy sledovat vliv změny parametrů obrazu na pozorovatele prakticky v reálném čase [1].

Metody subjektivního hodnocení byly uvedeny jen ve zkratce pro ilustraci a více lze o nich nalézt v literatuře, např. [1], [2] a [23].

1.2 Objektivní hodnocení

Jak již bylo řečeno, v současné době, ať už z důvodů časových, ekonomických a dalších, se více prosazují metody objektivní. Z počátku poskytovaly spíše jen přibližný odhad kvality obrazu, nyní máme však k dispozici metody, které jsou schopny spočítat velmi přesně kvalitu, která by byla vnímána i reálným pozorovatelem. Základními skupinami, na které lze metriky pro objektivní hodnocení kvality obrazu dělit, jsou metody s plnou referencí (full-reference, FR), částečnou referencí (reduced-reference, RR) a bez reference (no-reference, NR). V následující podkapitole budou krátce popsány principy full-reference a reduced-reference metod, které jsou výchozími pro metody no-reference, které jsou hlavním tématem práce.

1.2.1 Metody s plnou referencí

Jak vyplývá ze samotného názvu, metody s plnou referencí (full-reference, FR) potřebují ke svému fungování mít k dispozici originální (nezarušený) snímek. Lze je tedy srovnat s metodami Double Stimulus, které jsou často brány jako srovnání pro nové FR metody. U metod FR můžeme vysledovat dvojí přístup, buď pixelový nebo přístup při kterém se zohledňují vlastnosti lidského zraku (human visual system, HVS) [1].

Pixelově-orientované metriky

Základem pixelově-orientovaných metrik je střední kvadratická chyba MSE (mean squared error) a z něj pak odvozená nejjednodušší metrika PSNR (peak signal to noise ratio). MSE je definována jako střední hodnota druhých mocnin rozdílů dvou šedotónových hodnot pixelů obrazu. MSE lze vypočítat podle následujícího vztahu [4]

$$MSE = \frac{1}{TXY} \sum_t \sum_x \sum_y [I(t, x, y) - \tilde{I}(t, x, y)]^2, \quad (1)$$

kde I, \tilde{I} jsou černobílé obrazy o rozměrech X a Y a T snímcích. Hodnota PSNR v decibelech je poté dána vztahem

$$PSNR_{dB} = 10 \cdot \log_{10} \left(\frac{m^2}{MSE} \right), \quad (2)$$

kde m je maximální hodnota, kterou může pixel nabývat (např. 255 pro 8bitový obrázek). Typická hodnota PSNR pro kvalitní obraz by měla být kolem 50 dB. Metrika PSNR je sice velmi snadná na výpočet, nemusí však vždy poskytovat relevantní výsledky. Typickým příkladem je šum ve členité oblasti obrazu (např. obraz se skalami), který sice nebude pozorovatelem téměř vnímán, PSNR jej však vypočítá a výsledná objektivně určená kvalita obrazu jím určená bude proto nižší [4], [5].

Metody založené na modelu systému lidského vidění

Metriky typu PSNR nedovedou zcela postihnout vliv na reálného pozorovatele. Metody založené na modelu lidského zraku se toto snaží vyřešit. Základem pro modelování systému lidského vidění (human visual system, HVS) je určení citlivostní funkce kontrastu (contrast sensitivity function, CSF). Ta vlastně postihuje vlastnost lidského zraku, kdy se s rostoucí prostorovou frekvencí snižuje citlivost. V jednoduchých metrikách je tedy HVS brán jako jednoduchý prostorový filtr, který je definován právě pomocí CSF. Sofistikovanější metody ještě tuto charakteristiku dělí do dílčích pásem. V každém z těchto pásem je potom simulována další vlastnost HVS, maskování (na podobném principu, jak jej můžeme znát např. ze zpracování audia). Na základě těchto vlastností je poté počítán výsledný faktor kvality [6], [7].

1.2.2 Metody s částečnou referencí

Oproti metodám full-reference zde není zapotřebí mít k dispozici kompletní originální snímek. Metody reduced-reference se spokojí s částečnou informací extrahovanou z obrazu. Tou může být například informace o hranách získaná na základě hranových operátorů, informace o míře pohybu a jiné. Odkazy na RR metriky lze nalézt např. v [6].

1.2.3 Metody bez reference

Poslední skupinou objektivních metrik jsou bezreferenční metody. Ty tedy nepotřebují žádné apriorní informace o testovaném obraze. Problém tohoto přístupu však je, že metrika musí umět rozlišit mezi tím, co je rušením a co je obsahem snímku. Často se uvádí příklad, kdy v limitním případě může být obraz šachovnice identifikován jako blokové artefakty. Další nevýhodou může být například to, že nelze zcela dobře navrhnout nějakou univerzální metriku. NR metody se vždy specializují na konkrétní typ rušení a to poté v obraze hledají. Musíme tedy předem vědět, co je pro nás nejdůležitější (zda např. rozostření, blokové artefakty, šum) a podle toho se daná metrika navrhuje. Na druhé straně nespornou výhodou toho přístupu je, že lze na přijímací straně ohodnotit kvalitu obrazu, aniž bychom spolu s testovacím snímkem museli přenášet další informace o originálním snímku, navíc originální snímek nemusí

být vždy dostupný. Na problematiku NR metod lze pohlížet z několik stran, detailněji budou různé metody osvětleny v následující kapitole [6],[7].

Metrikami reduced-reference a no-reference se zabývala v roce 2009 i expertní skupina VQEG. Metody NR však byly z výsledného hodnocení nakonec všechny vyřazeny [15].

2 METRIKY ZALOŽENÉ NA BEZREFERENČNÍM PŘÍSTUPU

V předchozí kapitole bylo nastíněno, jak přibližně metriky bez použití referenčního snímku fungují. V této kapitole se podíváme konkrétně na metody, jak lze no-reference metriky navrhnout a seznámíme se s některými z nich. Jak již bylo zmíněno, metriky bez možnosti reference jsou obvykle specializovány na konkrétní typ rušení. Na základě jednotlivých typů rušení si uvedeme jak mohou jednotlivé algoritmy pracovat.

2.1 Metriky pro blokové artefakty

Základním typem rušení, které se objevuje u digitálně kódovaného videa, jsou blokové artefakty. To je jev daný už samotným principem kódování videa, běžným u kodeků rodiny MPEG a podobných. Tyto kodeky vytvářejí nejčastěji bloky o velikosti 8x8 pixelů, které jsou poté vstupem pro transformační kódování, nejčastěji DCT. Blokové artefakty jsou obvykle tím výraznější, čím nižších bitových rychlostí kodek využívá. Tento typ rušení se nejvíce podílí na vnímané kvalitě a proto je hlavním zájmem většiny metrik [6].

Předpokládá se, že blokové artefakty vytvoří v obraze další přidané hrany, které nemají informační hodnotu (nejsou tedy součástí obrazu). Aby bylo možné tyto „nové“ hrany rozeznat od skutečných hranic objektů, předpokládá se dále, že přirozené hrany jsou silnější než hrany způsobené blokovou strukturou. Ty se navíc vyskytují v konstantních vzdálenostech, které jsou dány právě velikostí bloků [10].

První NR metrikou pro blokové artefakty je metrika GBIM [11]. Ta měří vertikální a horizontální diference mezi řádky a sloupci. Váhovací koeficienty jsou stanoveny na základě středních a standardních odchylek bloků sousedících s danou hranicí. Výsledný koeficient kvality se stanoví normalizováním pomocí týchž hodnot ale určených mimo okrajové oblasti bloků. Nevýhodou tohoto relativně jednoduchého přístupu však je, že za jistých okolností může být normalizační koeficient roven nule, což vede k nestabilitě algoritmu. Dalšími možnostmi je například uvažovat obraz s blokovými artefakty jako součet obrazu nezarušeného a obrazu s čistě blokovou strukturou. Zajímavou se jeví metrika navržená v [12], kterou budeme označovat jako NR-SVD. Tato metrika je založena na principu singulární dekompozice (Singular Value Decomposition, SVD). Autoři zde počítají hodnoty SVD podél svislých a podélných hranic bloků. Jelikož je každý makroblok kódován jinak (ve smyslu kvantizace), projeví se tato změna i v hodnotách SVD. Na základě změn v těchto hodnotách SVD sousedících bloků lze pak definovat blokové artefakty v obraze [6], [12].

Do teď jsme uvažovali, že pro analýzu kvality přijímaného obrazu používáme dekódovaný obraz, tedy že známe hodnoty jednotlivých pixelů. Ne vždy je však možné obraz dekódovat, ať už z důvodu výpočetní náročnosti nebo z důvodu nedostupnosti dekóderu. Pokud jde o videosekvence kódované na základě DCT, je možné využít přímo spektrálních koeficientů ze samotného datového toku. Výpočet ve spektrální oblasti má zároveň tu výhodu, že spektrální koeficienty přímo reprezentují blokovou

strukturu. Základem je však důkladný rozbor datového toku a je možné, že bude třeba alespoň částečného dekódování. Podobně orientované metriky často dále využívají umělých neuronových sítí v úlohách deskriptorů [6], [16].

2.2 Metriky pro měření rozmazání obrazu

Základ pro určení rozmazání či ostrosti obrazu je podobný, jako v případě hledání blokových artefaktů – hrany. V tomto případě se ale hledají hrany samotného obrazu, nejen okraje jednotlivých bloků. Tím se stávají tyto metriky univerzálnějšími. Nejčastějším důvodem, jak může rozmazání nastat, je aplikace filtrů pro potlačení blokové struktury na přijímací straně. Ty se právě snaží odstranit nežádoucí blokové artefakty, zmíněné v předchozí podkapitole, ale za cenu ztráty všeobecné ostrosti celého snímku. Tím dojde ke snížení množství vnímaných detailů. Metriky hodnotící ostrost či neostrost scény mohou najít též uplatnění např. v samozaostřovacích systémech snímacích zařízení [6], [9].

Zástupcem NR metrik měřících rozmazání obrazu může být například metrika autorů Marziliano, Winklera a dalších, uvedená v [9]. Tato metrika pracuje v prostorové oblasti. Základem je užití hranového detektoru, autoři používali Sobelových masek. Na obraz byl aplikován Sobelův vertikální operátor. Poté byly procházeny jednotlivé řádky obrazu a stanoveny začátky a konce hran. Na základě šířek hran je poté definováno lokální rozmazání. Výsledný koeficient kvality je stanoven průměrováním lokálních rozmazání přes všechny oblasti hran. Metrika je sice prvotně navržena pro statické obrazy, lze ji však velmi jednoduše upravit pro použití na videosekvencích [9].

Další metrikou na podobném principu, je poměrně nová metrika uvedená v [13]. Zde autoři na začátku postupují podobně jako v [9]. Používají Sobelova maskového operátoru a následně prahování pro eliminace případného šumu a falešných hran. Následuje určení lokálního rozmazání. Zde se však již metriky rozcházejí. Tato metrika definuje lokální rozmazání na základě gradientu. Dále využívá globálních deskriptorů, jejich výstupy jsou vstupem umělé neuronové sítě. Metrika dosahuje lepších výsledků, než metrika [9], aniž by výrazně vzrostly výpočetní nároky. Vyšší jsou ovšem nároky na její implementaci [13].

Také problematiku rozostření lze řešit ve spektrální oblasti. Rozmazání obrazu je ve spektru definováno jako tlumení vyšších frekvenčních složek. Vyšší frekvence totiž odpovídají detailům a ostrým hranám v obraze. Metrikou používající spektrální oblast je např. metrika NROQM uvedená v [10]. Ta dokonce používá zároveň informace z prostorové oblasti a umožňuje tak lepší popis rušení. Dále tato metrika zahrnuje i vlivy dalších typů rušení a stává se tak velmi všestrannou metrikou [10], [14].

2.3 Srovnání náročnosti implementace

Následující podkapitola bude řešit problém náročnosti implementace metrik pracujících v prostorové oblasti a metrik využívajících pouze informací z datového toku. Jako zástupce pixelově-orientovaných metrik byly zvoleny metriky GBIM [11] a metrika publikovaná autory Marziliano, Winkler a dalšími [9]. Jako zástupce metrik, které pracují s informacemi z datového toku, zvolíme metriku pro MPEG-2 autorů Ichigaya,

Kurozumi a dalších [17], která používá DCT koeficienty k odhadu PSNR. Metriku pro kodek MPEG-2 jsme zvolili důvodu, že u tohoto kodeku lze, na rozdíl od modernějších, lépe pozorovat blokovou strukturu při nižších bitových rychlostech.

2.3.1 Generalized Block-Edge Impairment Metric (GBIM)

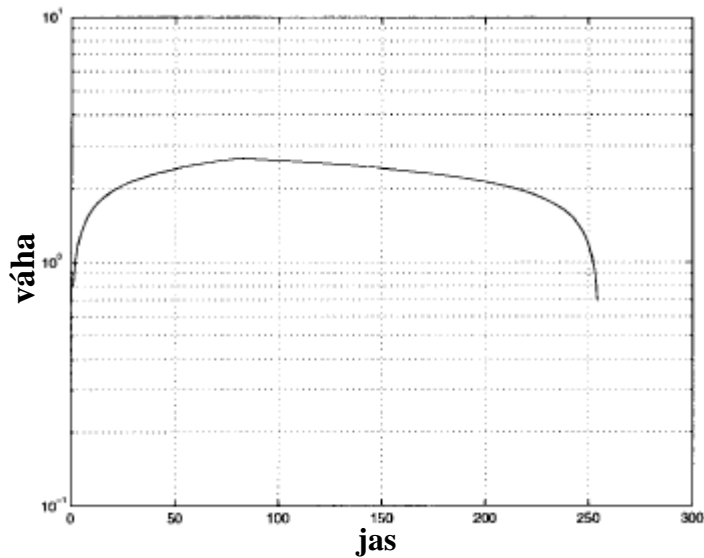
Základním předpokladem pro GBIM je obraz f definovaný jako $f = \{f_{c1}, f_{c2} \dots f_{cN_c}\}$, kde f_c jsou sloupce obrazu jehož šířka je N_c . Dále se definuje sloupcová diference jako

$$D_C f = \begin{bmatrix} f_{c8} - f_{c9} \\ f_{c16} - f_{c17} \\ \vdots \\ f_{c(N_c-8)} - f_{c(N_c-7)} \end{bmatrix}. \quad (3)$$

Předpokládá se, že byl použit kodek, který vytváří bloky 8x8 pixelů. Poté můžeme definovat metriku pro měření horizontální blokové struktury jako

$$M_h = \|W D_C f\| = \sqrt{\sum_{k=1}^{N_c/8-1} \|w_k [f_{c(8k)} - f_{c(8k-1)}]\|^2}. \quad (4)$$

Zde W je diagonální váhovací matice a $w_k = \text{diag}[w_{1,j}, w_{2,j}, \dots, w_{N_r,j}]$, kde N_r je výška obrazu a j je definováno $j = 8 \times k$, přičemž $k = 1, 2, \dots, N_c/8-1$. Přesnou definici váhovacích koeficientů zde již nebudeme uvádět, lze ji nalézt v [11]. Průběh váhovací funkce vidíme na následujícím obrázku.



Obr. 4 Váhovací funkce pro metriku GBIM, [11]

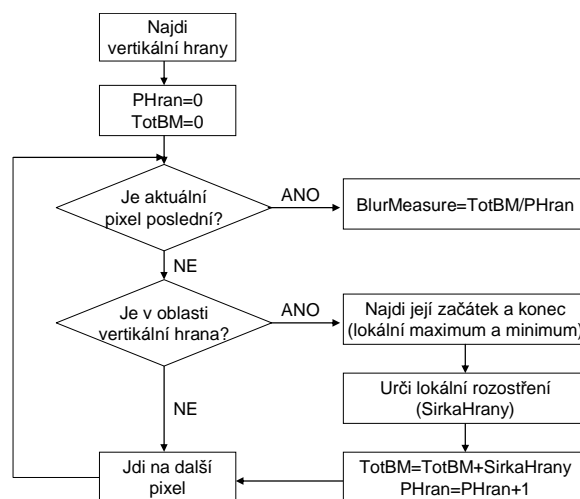
Vidíme, že váhovací funkce má své maximum v hodnotě 81. To je dáno tím, že rušení je nejvíce pozorovatelné, jestliže je hodnota jasu mezi 70 a 90 v 8bitové škále šedi. Hodnota 81 je poté přibližný průměr. Algoritmus výpočtu váhovací funkce je též relativně jednoduchý. Zahrnuje pouze klasické matematické operace, viz [11].

Výsledný koeficient M_{GBIM} je poté dán váhovaným součtem normalizovaného horizontálního a vertikálního koeficientu M_h a M_v , přičemž M_v se získá podobně jak je uvedeno výše, jen sloupce se vymění za řádky. Implementace této metriky je poměrně jednoduchá. Jednotlivé dílčí výpočty jsou definovány jednoduchými vztahy, které jsou vhodné pro implementaci v prostředí typu Matlab. Pro použití u videosekvencí pak stačí koeficient jistým způsobem váhovat přes jednotlivé snímky. Nevýhodou metriky je její rychle vzrůstající výpočetní náročnost pro videosekvence s vyšším rozlišením. Zbývá již jen poznamenat, že podobně jako většina ostatních, se GBIM počítá pouze na základě jasové složky videa [11].

2.3.2 No-reference Perceptual Blur Metric (Marziliano)

Druhou metrikou, jejíž implementaci si rozebereme více, je metrika autorů Marziliano, Winklera a dalších [9]. Metrika nedostala od autorů oficiální název, dále je budeme uvádět podle jména autorky, tedy Marziliano. Tato metrika také pracuje v prostorové oblasti, zaměřena je pak na hledání rozostření v obraze. To může nastat jako důsledek pohybu v obraze, špatného doostření snímacího systému apod.

Základem pro metriku Marziliano je užití hranového detektoru. Přímo v návrhu autoři zmiňují vertikální Sobelovu masku pro detekci vertikálních hran, lze užít samozřejmě i následnou horizontální pro získání hranové reprezentace v obou směrech. Jako výhodné se zde jeví použití Cannyho detektoru, jehož výstupem je čistá hranová reprezentace. Dále metrika prochází jednotlivé řádky a hledá začátky a konce hran, které definuje jako lokální oblasti extrémů. Šířka hrany, daná rozdílem začátku a konce hrany, je poté lokálním rozmazáním. Globální rozmazání obrazu je poté průměrem těchto lokálních hodnot přes oblasti všech hran. Vývojový diagram algoritmu je na následujícím obrázku.



Obr. 5 Vývojový diagram algoritmu Marziliano, [9]

Zde PHran značí celkový počet nalezených hran, TotBM je celkové rozmazání. Výsledný koeficient BlurMeasure je průměrná hodnota jednotlivých lokální šířek hran.

Aplikace hranového detektoru na obraz je v prostředí typu Matlab otázkou jediného řádku kódu. Dále zde budeme v cyklu počítat šířky hran, což je též v prostředí Matlab jednoduchou záležitostí. Problémem zde však budou výpočetní nároky při aplikaci metriky na videa s vysokým rozlišením. Konvoluce obrazu s maskou, případné hledání hran pomocí převedení obrazu i masky do spektrální oblasti a následné vynásobení, si vyžádá svůj výpočetní čas. Ani aplikace cyklů není z hlediska výpočetní náročnosti příliš vhodná. I zde je výsledný koeficient počítán pro šedotónové statické obrazy, jednoduše jej lze upravit i pro videosekvence. V tom případě budeme koeficient počítat pro každý snímek, výsledek bude jistým váhovaným průměrem vypočítaných hodnot. [9], [14].

2.3.3 Odhad PSNR na základě koeficientů DCT

Další možností, jak získat informace o obraze, na jejichž základě by bylo možné určit kvalitu obrazu, je analýza koeficientů DCT. Ta má oproti předešlým možnostem tu výhodu, že není třeba na přijímací straně video dekódovat, vystačíme si tedy jen s přímým datovým tokem. Zde si rozebereme metodu navrženou v [17]. Zde autoři, na základě Parsevalova teorému o zachování energie navrhuji, že PSNR lze vypočítat i přímo na základě hodnot koeficientů DCT.

$$PSNR = 20 \log \left(\frac{S_{p-p}}{\sqrt{MSQE}} \right), \quad (5)$$

kde $MSQE$ je střední kvadratická kvantizační chyba definovaná jako

$$MSQE = \frac{1}{N} \sum_{i=1}^N (X_i - X'_i)^2, \quad (6)$$

kde X_i a X'_i jsou DCT koeficienty zdrojového resp. odpovídajícího dekódovaného signálu. Na základě tohoto vztahu bychom však potřebovali mít k dispozici i DCT koeficienty zdroje. Pokud však známe rozdělení amplitud DCT koeficientů zdroje, můžeme psát

$$MSQE = \frac{1}{N} \sum_{v=0}^7 \sum_{u=0}^7 \sum_{q_code=0}^{31} \left(n_{(u,v,q_code)} \sum_i \int_{\alpha_i}^{\beta_i} f_i(X) dX \right), \quad (7)$$

kde f_i je definováno vztahem

$$f_i = p_{(u,v,q_code)}(X)(X - q_i)^2, \quad (8)$$

a $p_{(u,v,q_code)}(X)$ je funkce hustoty pravděpodobnosti DCT koeficientu $X(u,v)$ pro každý kvantizační krok a q_i reprezentuje hodnotu kvantizačního intervalu i . Dále $n_{(u,v,q_code)}$ vyjadřuje počet vzorků DCT koeficientů pro (u,v) pro každý kvantizační krok. Kvantizační interval poté označujeme $[\alpha_i, \beta_i]$. Zbývá nám ještě vyjádřit funkci hustoty pravděpodobnosti. Ta je dána vztahem

$$p(x) = \frac{1}{\sqrt{2}\sigma} e^{\left(\frac{-\sqrt{2}|x|}{\sigma} \right)}, \quad (9)$$

kde σ je standardní odchylka [17].

Postup pro výpočet PSNR z DCT koeficientů je poté následující:

1. Extrahování kvantizovaných DCT koeficientů a dekodování velikosti kvantizačního kroku z bitového toku
2. Pro každý DCT koeficient a kvantizační krok se vypočítá standardní odchylka
3. Odhad funkce hustoty pravděpodobnosti na základě vztahu (9)
4. Výpočet MSQE pomocí (7) a (8)
5. Výpočet PSNR vztahem (5).

Zde jsme zjednodušeně popsali, jak daná metrika pracuje. Detailnější informace, zejména pak odvození statistických vlastností DCT koeficientů lze nalézt v [17]. Autoři zde dále např. odlišují výpočet *MSQE* pro intra-kódované bloky a ostatní bloky.

Už na první pohled je jasné, že z hlediska náročnosti implementace je tato metrika složitější, než předešlé. Zásadní zde bude algoritmus pro analýzu bitového toku. Je tedy nutné znát detailně jeho strukturu. Samotné výpočetní vztahy sice nejsou až tak komplikované, setkáme se zde však s velkým množstvím proměnných. Metrika je však bez problémů použitelná i pro HD rozlišení.

Další způsob, který se často využívá, je zapojení umělých neuronových sítí do výpočtu. Příklad takové metriky je např. v [16]. Zde autoři extrahují z datového toku význačné informace, které jsou poté vstupem neuronové sítě se zpětným šířením chyby. Nevýhodou pro nás je však to, že aby byla metrika použitelná, je třeba ji „naučit“ souvislosti mezi hodnotami extrahovaných informací a kvalitou vnímanou skutečným pozorovatelem. Autoři tedy nejdříve vedli test subjektivního hodnocení kvality, na jehož základě se pak neuronová síť učila, což je pro naše potřeby nevhodné.

3 POPIS VYTVOŘENÉ APLIKACE NO-REFERENCE VIDEO QUALITY TOOL

Pro potřeby práce byla v prostředí Matlab navržena aplikace s grafickým rozhraním (GUI), do které bude možné implementovat některé z dříve uvedených metrik pro bezreferenční hodnocení obrazové kvality videosekvencí. Program byl napsán ve verzi Matlab R2010a 64-bit pod OS Windows 7 64-bit. Aplikace dostala na základě její hlavní funkce jednoduchý název No-Reference Video Quality Tool (dále jen NR VQT), tedy Nástroj bezreferenčního hodnocení kvality videa.

Program by měl zastávat následující funkce:

- otevření souboru,
- zjištění parametrů (rozlišení a počet snímků),
- extrahování jasové složky obrazu, případně dat z bitového toku,
- hodnocení kvality,
- zobrazení výsledků.

Vývojový diagram aplikace by poté mohl vypadat, jako na následujícím obrázku.

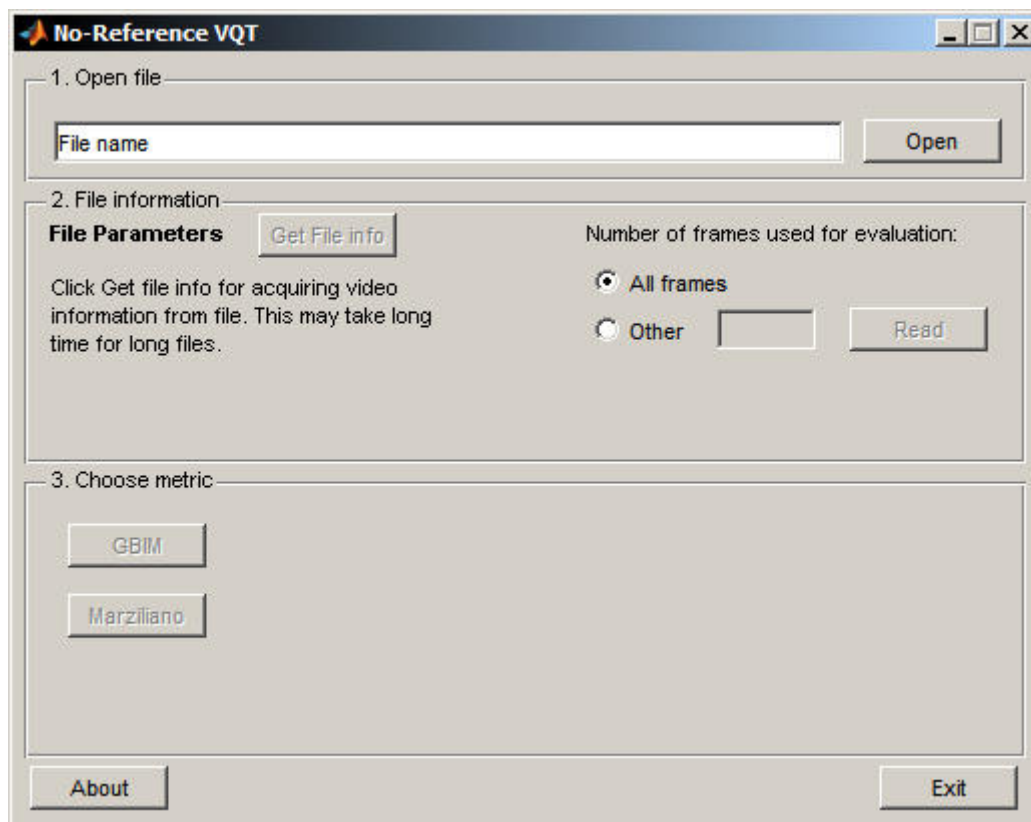


Obr. 6 Diagram aplikace NR VQT

Případné ochrany proti neplatnému zadání (např. výběr špatného souboru apod.) nejsou v diagramu zobrazeny z důvodu jeho následného velkého rozměru. Jednotlivé metody ochrany, které budou implementovány, budou zmíněny dále v textu. Operace výběru souboru, volby počtu snímků pro zpracování a volba metriky budou na uživatelské volbě, ostatní procesy budou plnit samostatné funkce.

3.1 Grafické rozhraní

Pomocí průvodce GUIDE pro návrh uživatelského rozhraní programu Matlab,[19] byl navržen vzhled programu. Rozvržení jednotlivých prvků můžeme vidět na obrázku 7. Snímek je pořízen při startu programu.



Obr. 7 Hlavní okno programu NR VQT

Grafické rozhraní je rozděleno do tří logických oddílů. První slouží k vybrání souboru, který chceme analyzovat. V druhém oddíle jsou ovládací prvky pro zjištění parametrů videosouboru a nastavení volitelného počtu zpracovávaných snímků. Zjištěné parametry (rozlišení a počet snímků) jsou zde zároveň zobrazeny. Nejdůležitějším je poslední třetí oddíl. Zde se nacházejí ovládací tlačítka pro spuštění výpočtu jednotlivých metrik. Výsledky výpočtu jsou poté zobrazeny v prostoru vedle těchto ovládacích prvků. Mimo tyto oddíly je umístěno tlačítko pro zobrazení informací o aplikaci a tlačítko k jejímu ukončení.

Z obrázku je patrné, že všechna tlačítka, kromě tlačítek „Open“, „About“ a „Exit“, jsou po startu aplikace neaktivní. To je z důvodu, aby byl uživatel veden k dodržení přesného sledu akcí. Zároveň to zabezpečí jednoduchou ochranu před tím, že by se uživatel pokusil např. spustit metriku, aniž by načel video, což by vedlo k chybovému hlášení. Pokud uživatel provádí akce postupně, tlačítka jsou také postupně aktivována. Tedy například po správném výběru souboru se aktivuje tlačítko „Get File info“, které slouží k získání informací o souboru. Proběhne-li toto v pořádku, aktivuje se možnost

volby počtu snímků a tlačítko „Read“ pro načtení jasového signálu z videosouboru a podobně. Prakticky v každém kroku jsou zavedeny kontroly správného zadání, více bude zmíněno u jednotlivých částí.

3.2 Ověření typu souboru a zjištění informací o videu

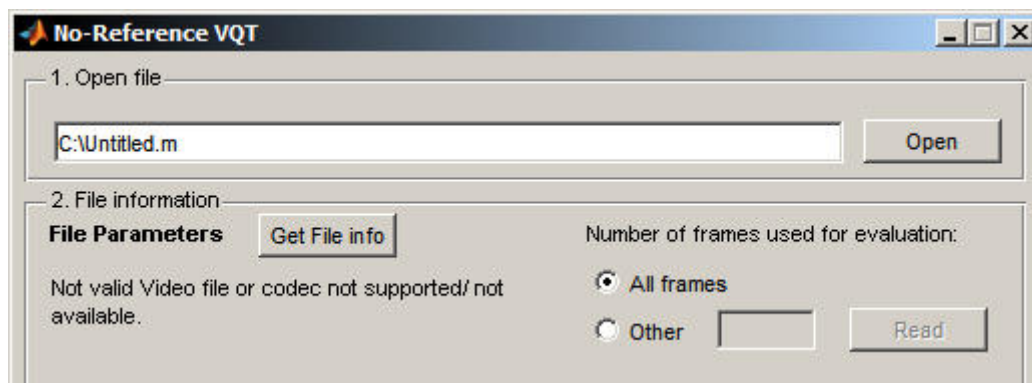
Pro výběr souboru je využito velice jednoduché funkce `uigetfile`. Ta vyvolá klasické okno pro výběr souboru. Zde je také již použita první uživatelská ochrana, která zabezpečí, že pokud uživatel nevybere soubor, nebude aplikace vypisovat chybu.

Nyní přichází na řadu zjištění informací o souboru. První verze programu zde využívala funkci `mmfileinfo`. Ta se nám zpočátku jevila jako velice efektivní. V průběhu testování se však objevily problémy, které by způsobily, že by nebylo možné zaručit kompatibilitu aplikace na ostatních počítačích. Týž problém se objevil i u další funkce, kterou jsme používali k načtení obrazové informace z videa, funkce `mmreader`. Více o tomto problému a jeho vyřešení bude uvedeno dále.

Ke zjištění informací tedy aktuální verze aplikace používá externí program `ffprobe.exe`, který je součástí balíku programů FFmpeg, [18]. Syntaxe pro naše použití je následující:

```
ffprobe -i myvideo.avi -show_streams -count_frames.
```

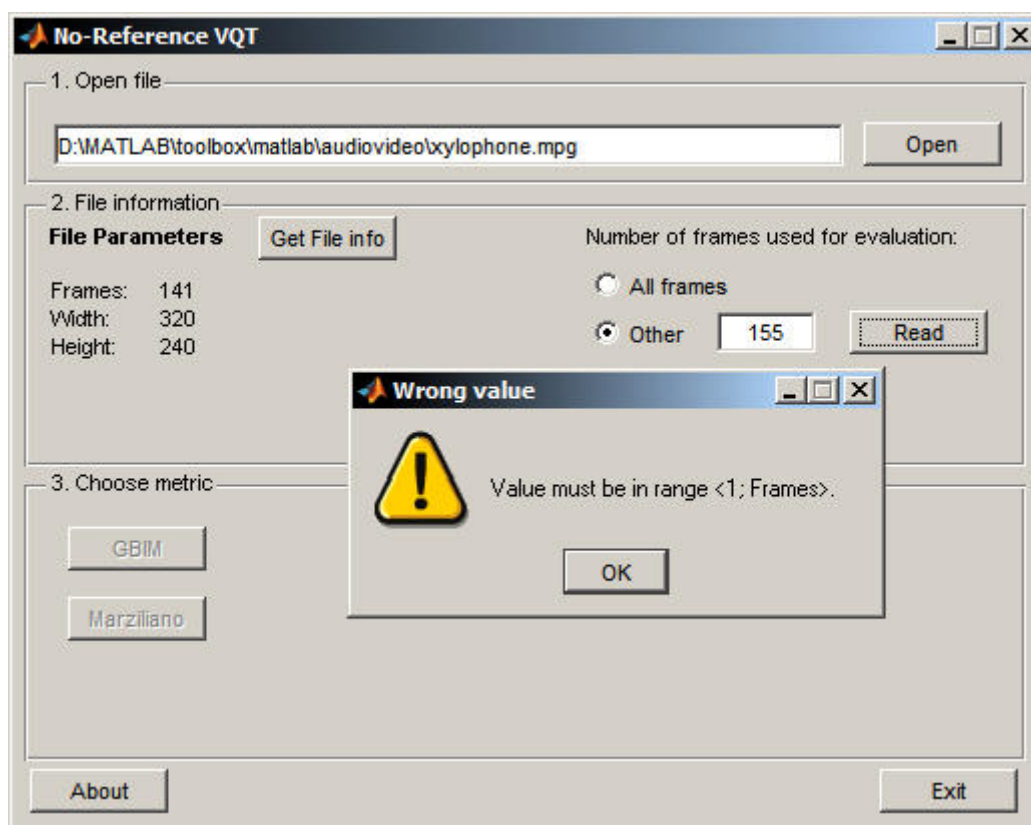
Tento příkaz je volán přímo z prostředí Matlab. Poté je zachytáván výpis odpovědi programu a ten je podroben analýze námi vytvořenou funkcí, která z něj vybere pro nás podstatné informace, tedy rozlišení obrazu a počet snímků. Je zde dále řešen i případ, že se uživatel pokusí načíst nepodporovaný soubor. V tomto případě je toto identifikováno a vypsáno do okna aplikace. Hlášku můžeme vidět na následujícím obrázku. Povšimněme si také, že v případě nepodporovaného souboru je také znemožněna další činnost (tlačítko Read je neaktivní). Zde se jednalo o pokus načíst soubor aplikace Matlab.



Obr. 8 Hláška při volbě neplatného souboru

3.3 Volba počtu snímků, získání jasového signálu

Pokud již máme k dispozici základní parametry videosouboru, je možné z něj extrahovat jasovou složku obrazu pro jednotlivé snímky. Jasový signál je pro nás plně dostačující, protože námi použité metriky pracují pouze s informací v něm obsaženou, navíc nám načítání pouhé jasové složky sníží velikost obsazené paměti. Aplikace dále umožňuje zvolit počet snímků, které budou načteny a podrobeny hodnocení kvality. Jak je vidět na obrázku 7, jednoduchým přepínačem lze volit mezi celým videem (tedy všemi snímky) nebo uživatelem zadanou hodnotou. Program má zavedenu kontrolu správnosti zadání. Pokud uživatel zadá číslo větší než maximální počet snímků, je na to upozorněn varovným hlášením. Stejně tak je upozorněn, zadá-li hodnotu, která není číslem, případně číslo je menší než 1. Pokud je vstupem číslo s desetinnou částí, je číslo zaokrouhлено na nejbližší nižší celé číslo. Příklad chybového hlášení při zvoleném počtu snímků mimo maximální rozsah je uveden na obrázku 9.



Obr. 9 Příklad chybového hlášení při zadání počtu snímků

Nyní můžeme zmínit výše uvedený problém, na který jsme při testování narazili. Původní idea byla, použít k načítání obrazové informace funkci Matlabu `mmreader`. Pomocí ní je možné načíst obrazovou informaci snímek po snímku. Výstupem jsou data RGB, byl proto použit přepočítání na jasový signál. Testování však odhalilo nedokonalost tohoto řešení. Funkce `mmreader`, jakožto i `mmfileinfo`, využívají v prostředí Windows k dekódování videosouborů filtry `DirectShow`. Nelze zaručit, že na daném počítači budou vždy konkrétní filtry dostupné či zda budou správně fungovat. To by způsobilo,

že by aplikace v mezní situaci nebyla schopna načíst žádná video data a stala by se naprosto nepoužitelnou. Při testování bylo dokonce dosaženo případu, kdy aplikace nebyla schopna načíst ani nekomprimované prokládané video (nekomprimované video v kontejneru AVI). To je pro naše potřeby zcela nevhodné. Menším problémem pak bylo, že funkce nebyly vždy schopny správně určit počet snímků. To by sice šlo řešit buď čtením všech snímků a zpětným zjištěním jejich počtu nebo aproximací na základě znalosti doby trvání sekvence a její snímkové rychlosti. První způsob je časově náročný a pro velké soubory jej nelze použít. Druhá metoda je sice jednoduchá, výsledek však nepřesný. Proto bylo od takového řešení ustoupeno.

Situace byla vyřešena pomocí externí aplikace. V tomto případě je to enkodér `ffmpeg.exe`, který je taktéž součástí balíku FFmpeg, [18]. Celý tento balík je poskytován pod licencí GPL, lze jej tedy bez problému použít pro naše potřeby. Řešení je následující. Z prostředí Matlab je volán `ffmpeg`, který video dekoduje a jeho výstupem bude v našem případě soubor obsahující pouze jasový signál. To samozřejmě pro námi zvolený počet snímků. Syntaxe je v tomto případě následující:

```
ffmpeg -i <input.avi> -vframes <nFrames> -pix_fmt gray -y temp.yuv.
```

Takto provedený příkaz dekoduje „nFrames“ snímků ze souboru „input.avi“ a uloží jeho jasový signál do souboru s názvem „temp.yuv“. Parametr „-y“ pouze zabezpečí, že v případě, že v programové složce již soubor s tímto názvem existuje, bude automaticky přepsán.

Jakmile dosáhneme toho, že máme k dispozici jasový signál, můžeme jej velice jednoduše v prostředí Matlab načíst jako binární soubor. Poté už jen maticovými operacemi zbývá data přeuspořádat tak, abychom získali trojrozměrnou matici hodnot jasového signálu pro všechny snímky sekvence. Výhody z tohoto řešení plynoucí jsou jasné, odstranili jsme závislost naší aplikace na dostupných dekodovacích filtrech na daném počítači. Vedlejší výhodou tohoto řešení je značné urychlení samotného procesu načítání dat. Pokud jsme používali funkci `mreader`, bylo nutné načítat obrazovou informaci z videosouboru v cyklu délky rovnající se počtu snímků. V případě referenčního videa „xylophone.mpg“, které je součástí programu Matlab, trvalo získání jasového signálu přibližně 4 až 5 vteřin. Poznamenejme, že se jedná o sekvenci s rozlišením 320x240 pixelů a 141 snímky. V případě použití balíku FFmpeg se doba načtení dat včetně jejich dekodování, snížila na necelé 2 vteřiny. Tento rozdíl by se samozřejmě zvětšoval s rostoucím rozlišením a počtem snímků, jelikož v předešlém případě bylo nutné používat cyklů. Nevýhodou je, že aplikace si na disk ukládá dočasný soubor s jasovou složkou. V aplikaci je sice dále řešeno jeho následné smazání, jakmile už není potřeba, je však nutné mít oprávnění k zápisu do programové složky a aplikaci také nelze používat z médií, která jsou určena pouze pro čtení (např. CD). Dále lze za nevýhodu považovat i to, že uložený soubor může mít značnou velikost, např. pro referenční sekvenci přibližně 10 MB. Přesto však výhody převažují nad zápory tohoto řešení. Celý tento výše uvedený postup má na starosti námi napsaná funkce s názvem `getLuma.m`. Ukázkou kódu uvádíme níže: vstupem funkce je cesta k vybranému souboru, zvolený počet snímků a rozlišení obrazu. Výstupem je poté proměnná `Y` – trojrozměrná matice obsahující jasový obrazový signál pro jednotlivé snímky.

```

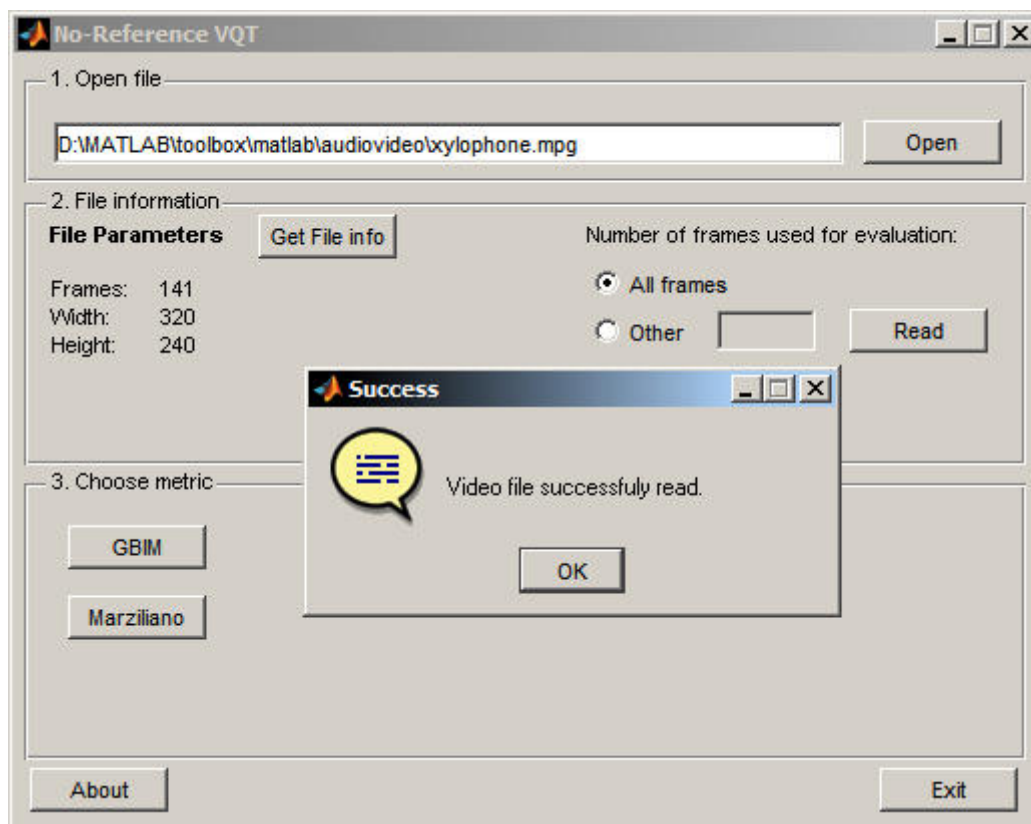
function Y=getLuma(file, nFrames, W, H)
ffstring=sprintf('ffmpeg -i "%s" -vframes %d -pix_fmt gray -y
temp.yuv', file, nFrames); % creating ffmpeg command

[a,b]=system(ffstring);
if a==0
    fid=fopen('temp.yuv');
    Ytemp=fread(fid, '*uint8');
    fclose(fid);
    Y=reshape(Ytemp,W,H, nFrames);
    Y=permute(Y, [2 1 3]);

else
    warndlg('Unexpected error ocured. Close the aplication and
try other video file.', 'Unexpected error')
Y=[];
end

```

Na ukázce kódu je názorně vidět, jak je celá procedura načítání sekvence naprogramována velice jednoduše. Nejprve je nutné na základě cesty k souboru a počtu snímků definovat příkaz pro FFmpeg. Pokud byl příkaz proveden bez chyby, je FFmpeg-em vytvořený soubor „temp.yuv“ načten do prostředí Matlab. Data jsou načtena jako vektor o délce ($W \cdot H \cdot nFrames$) s přesností 8 bitů. Jelikož jsou data vektor, je nutné je přeuspořádat. K tomu slouží funkce `reshape`, která z dat vytvoří 3D-matici. Jelikož však `reshape` postupuje po sloupcích, je nutné na závěr vyměnit hodnoty řádků a sloupců. Tím je procedura u konce a v okně hlavní aplikace se objeví hlášení o úspěšném načtení souboru, jak je uvedeno na obrázku 10.



Obr. 10 Hlášení o úspěšném načtení souboru

Nastane-li během zpracování příkazu FFmpeg chyba, je uživatel upozorněn a je mu doporučeno zapnout a vypnout aplikaci a pokusit se načíst jiný soubor. Toto je zde pouze pro nečekané situace, protože k podobné chybě by při správném užívání aplikace nemělo dojít.

3.4 Použité metriky a zobrazení výsledků

Pro prvotní provoz naší aplikace jsme se rozhodli použít dvě z metrik uvedené v kapitole 2.3. Jsou jimi metrika Generalized Block-Edge Impairment Metric (GBIM), [11] a No-Reference Perceptual Blur metric autorky Marziliano a dalších [9]. Jak již bylo zmíněno, jedná se o pixelově orientované metriky, jejich vstupem je tedy šedotónový obraz. Obě metriky lze poměrně jednoduše implementovat v prostředí typu Matlab. Jejich implementace byly však již několikrát provedeny, a proto by bylo neúčelné je implementovat znovu. Rozhodli jsme se tedy použít již hotové funkce pro obrazy, které byly jednoduše upraveny pro použití na videosekvencích.

3.4.1 Implementace GBIM

K provedení výpočtu metriky GBIM je použita implementace pro Matlab autora Mattea Naccari, [20]. Autor byl toho času členem Instituto de Telecomunicações v Lisabonu v Portugalsku. Licence jeho funkce s názvem `compute_GBIM.m` dovoluje její použití pro vědecké a nekomerční účely včetně její modifikace. Autor vytvořil funkci podle metriky [11], jejíž vstupem je šedotónový obraz a také informace o velikosti bloku. Metrika vychází z předpokladu, že blokové artefakty jsou nejvíce patrné na rozhraních sousedních bloků, a tedy na základě informace o velikosti bloků můžeme tyto hrany vyhledávat. Funkce uvažuje v souladu s [11] jako střední hodnotu šedi, na kterou je lidský zrak nejcitlivější, hodnotu 81. Programátor má však možnost si případně zvolit i hodnotu jinou.

Pro naše potřeby byla funkce mírně upravena. Byl přidán jeden vstupní parametr, počet snímků. Dále byl do funkce přidán cyklus, díky kterému počítáme hodnotu MGBIM pro každý jednotlivý snímek. Výstupním parametrem funkce je poté vektor hodnot MGBIM pro všechny snímky. V našem programu je funkce pojmenována jednoduše `GBIM.m`. Její volání je následující:

```
resultGBIM=GBIM(Y, 8, setFrames);
```

Zde „Y“ je 3D-matice jasového signálu a „setFrames“ značí počet snímků, které se mají zpracovávat. Hodnota „8“ udává, že předpokládáme velikost bloku 8x8. To je typická hodnota pro kódování MPEG-1 a MPEG-2. Vzhledem k tomu, že aplikace je navržena převážně pro tento typ kódování, je funkce vždy pevně volána s tímto parametrem. V případě potřeby by však nebyl problém umožnit tento parametr měnit. Jak již bylo řečeno, výstupem funkce je vektor. Jak jsou data následně zpracována, bude rozebráno dále v textu.

3.4.2 Implementace NR Perceptual Blur Metric – Marziliano

Druhou metrikou, která byla v programu použita, je metrika autorky Pina Marziliano a jejích kolegů. Tato metrika je orientována na zjištění rozmazání obrazu. I v tomto případě jsme použili již existující implementaci pro prostředí Matlab. Zde se jedná o implementaci použitou v programu VBQUEST, [21] autorů Lina Karam a Adithya Murthy. I tato může být dále použita, pokud součástí souboru bude originální licenční prohlášení, což je v našem případě dodrženo. Původní funkce dostala od autorů název `marziliano.m`, kterého se držíme i my. Vstupem funkce je opět šedotónový obraz a jeho rozměry, výstupem pak koeficient objektivního měření. Metrika funguje tak, že nejprve spočítá hranovou reprezentaci obrazu, zde je použit Cannyho detektor. Dále se v cyklech počítají šířky jednotlivých hran, z čehož je pak určen výsledný koeficient.

Pro naše potřeby nebyla funkce nijak modifikována. Výpočet metriky pro jednotlivé snímky je prováděn opakovaným voláním funkce v cyklu, jak je uvedeno níže:

```
[M N]=size(Y(:,:,1));  
for k=1: setFrames  
    resultMarz(k)=marziliano(Y(:,:,k), M, N);  
end
```

Zde je „Y“ opět matice jasového signálu a „setFrames“ pak počet snímků. Výstupem bude potom vektor „resultMarz“, jehož délka je rovna počtu snímků, a obsahuje koeficienty kvality pro jednotlivé snímky. Funkčnost metriky a její schopnost skutečně identifikovat viditelné rozmazání v obraze byly experimentálně ověřeny na čtyřech testovacích sekvencích, na které byl postupně aplikován rozmazávací filtr, viz obr. 11



a) Originál: 4,69



b) Blur filtr 1x: 6,56



c) Blur filtr 2x: 7,42



d) Blur filtr 3x: 8,14

Obr. 11 Ověření metriky Marziliano, testovací sekvence a hodnoty kvality

Byla použita již dříve zmíněná sekvence „xylophone.mpg“, která byla následně rozmazána pomocí filtru v aplikaci VirtualDub. Z obrázku a uvedených hodnot vidíme, že metrika správně určila postupně se zvyšující rozmazání obrazu..

3.4.3 Průběh výpočtu a zobrazení výsledků

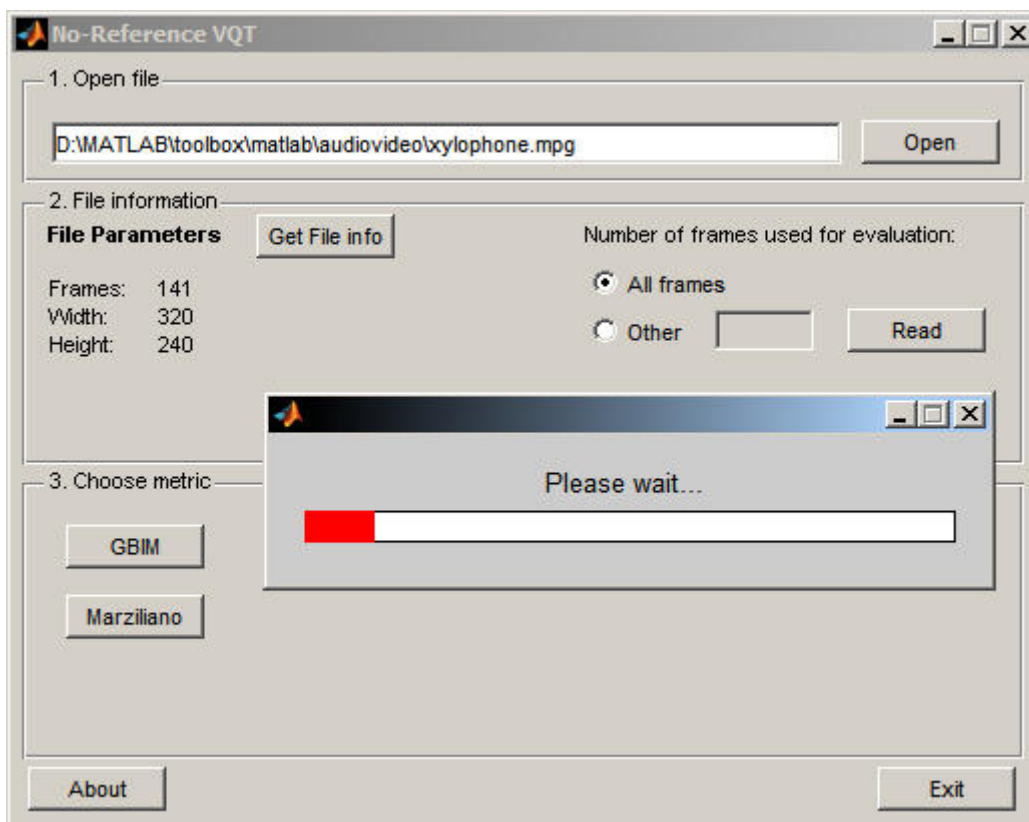
Výpočet metrik může být při použití souboru s velkým rozlišení a velkým počtem snímků značně časově náročný. Chtěli jsme proto uživatele informovat o průběhu aktuálního výpočtu. Toho jsme dosáhli jednoduchým vložením tzv. „progress bar-u“ do naší aplikace. Pro jeho zobrazení je použito funkce `waitbar`, a to následujícím způsobem:

```
h = waitbar(0, 'Please wait...'); %create progress bar
for k=1: setFrames

    ... computing metric ...

    waitbar(k / setFrames, h); %update progress bar
end
```

Funkce vytvoří okno s ukazatelem a v každém kroku výpočtu (tedy s každým dalším vypočítaným snímkem) se zvýší hodnota ukazatele o 1. Použití této funkce se sice velice jednoduché, její algoritmus však mírně prodlužuje výsledný čas potřebný k provedení výpočtu. Konkrétněji o tomto pohovoříme v oddíle, který se bude zabývat výkonem celé aplikace. Situaci můžeme vidět na obrázku 12.



Obr. 12 Zobrazení ukazatele průběhu výpočtu

Výstupem obou metrik, jak již bylo několikrát zmíněno, je vektor koeficientů. Aplikace je navržena tak, že uživatelé ze všech těchto hodnot vybere pouze zajímavé hodnoty a ty zobrazí v prostoru k tomu určeném. Vybíranými hodnotami jsou maximum, minimum, průměrná hodnota a medián. Toto jsou však spíše statisticky zajímavé veličiny a nemusejí příliš vypovídat o výsledné kvalitě vnímané uživatelem. Proto aplikace počítá ještě jednu hodnotu nazvanou „Overall score“. Pro výpočet tohoto „Overall score“ je použita jedna z tzv. „temporal pooling“ metod, tedy metod, které se snaží vybrat z jednotlivých koeficientů kvality hodnotu, která bude mít dobrou výpovědní hodnotu a bude v korelaci se skutečným vjemem reálného pozorovatele.

Vlivem „temporal pooling“ metod na výsledné objektivní hodnocení obrazové kvality se zabývají autoři článku [22]. Autoři zde zkoumají vliv šestice „temporal pooling“ metod na výsledky získané různými metrikami. Určili poté nejlepší kombinace metoda-metrika, díky kterým lze získat nejlepší výsledky. Pro naše potřeby jsme zvolili metodu, která je za prvé jednoduchá na implementaci, a za druhá dosahovala dobrých výsledků napříč metrikami. Byla zvolena metoda Minkowského sumace.

Minkowského sumace je definována jednoduchým vztahem (10), [22]:

$$OM_{Mink} = \left[\frac{1}{T} \sum_{t=1}^T OM^p(t) \right]^{1/p}, \quad (10)$$

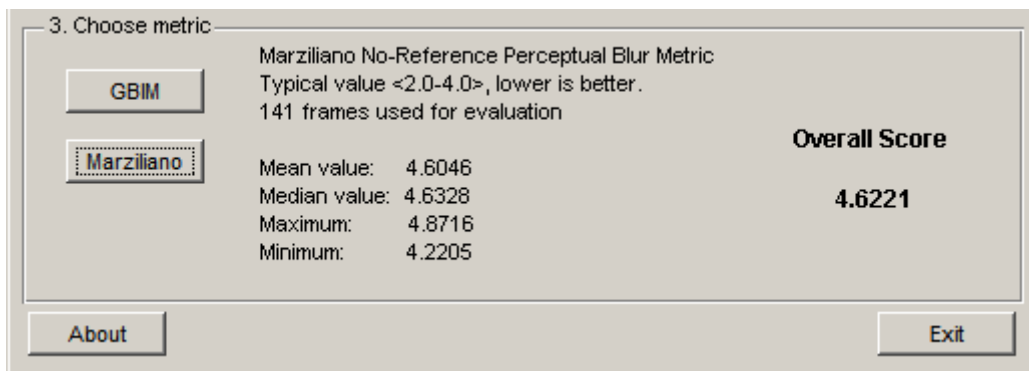
kde T je počet snímků, $OM(t)$ je hodnota objektivní metriky snímku t a p je parametr.

Autoři zkoumali vliv parametru na výsledek v rozmezí hodnot 1 až 12. Větší hodnota parametru p způsobuje, že výsledek je více ovlivněn nejvyšší hodnotou. Pro největší výpovědní hodnotu zde autoři doporučují hodnotu $p = 10$, které se budeme držet i my. Tato metoda nebere v úvahu tzv. recency effect, tedy jev, kdy při subjektivním hodnocení dává pozorovatel největší důraz na to, co viděl jako poslední. Tedy např. budeme-li mít dlouhou sekvenci, jejíž začátek je vysoce jakostní, ale závěr je zarušený, bylo by výsledné hodnocení negativní. Pro výpočet Minkowského sumace byla napsána funkce s názvem `minkowski.m`.

```
function score=minkowski(results)
T=length(results);
p=10;
score=(1/T*(sum(results.^p)))^(1/p);
```

Vidíme, že její kód je velmi jednoduchý, výsledky jsou však velmi dobré a takto vypočítané celkové skóre má dobrou výpovědní hodnotu.

Samotné zobrazení výsledků se provádí do prostoru vedle tlačítek pro spuštění metrik. Výpis výsledků v případě použití metriky Marziliano vidíme na následujícím obrázku.



Obr. 13 Zobrazení výsledků

Kromě samotných výsledků se do okna aplikace vypíše dále základní informace o použité metrice a informace o počtu snímků použitých pro hodnocení.

3.5 Optimalizace a výkon aplikace

Při návrhu byl kladem důraz i na to, aby nebyla aplikace zbytečně náročná na výpočetní výkon. Vzhledem k tomu, že musíme počítat vždy metriku pro každý snímek, nevyhneme se při programování použití cyklů. Kde to však bylo jen trochu možné, snažili jsme se kód optimalizovat. Níže uvedené časové údaje se budou vztahovat k testování na referenčním stroji s těmito parametry:

- Procesor AMD Athlon 64 X2 TK-55 (1800 MHz),
- RAM 3 GB, 667 MHz.

Příklad optimalizace si můžeme ukázat na rozdílném způsobu volání metrik GBIM a Marziliano. U metriky GBIM jsme cyklus vložili přímo jako část kódu samotné metriky. To z toho důvodu, že metrika ještě před tím počítá některé koeficienty, které se však již dále během výpočtu nemění (závisí na velikosti obrazu a nastavené střední hodnotě šedi). Proto je cyklus aplikován až po proběhnutí těchto procedur. Tím jsme mohli mírně zkrátit výpočetní dobu. V případě hodnocení Marziliano se výpočty, které by byly konstantní pro celou sekvenci, neprovádějí, proto ji voláme opakovaně.

Dalšího urychlení běhu aplikace jsme dosáhli již výše zmíněnou změnou principu načítání dat. Jádro jsme přenechali externí aplikaci a v prostředí Matlab jen načítáme binární data. Pro referenční sekvenci „xylophone.mpg“ jsme tím dosáhli urychlení z času přibližně 4 s na dobu menší než 2 s. A z toho samotné načtení dat ze souboru do Matlabu trvá pouze přibližně 0,16 sekund. Vzhledem k tomu, že v prvním případě doba potřebná pro zpracování souboru roste s rozlišením rychleji, než v případě s použitím FFmpeg, byl by v případě větších sekvencí tento rozdíl ještě větší.

Protichůdně k požadavku na optimalizaci kódu může působit zavedení zobrazení informace o průběhu výpočtu. Jelikož je v každém kroku nutné zvýšit hodnotu ukazatele, roste s počtem snímků i doba výpočtu. V případě stále téhož referenčního videa se doba zvýšila o přibližně 2 vteřiny. Tato doba je logicky pro obě metriky stejná, závisí totiž jen použitým počtu snímků. I přes tento fakt je zobrazení průběhu výpočtu v aplikaci použito pro lepší informování uživatele. Navíc mu i poskytuje možnost

výpočet přerušit, což bez něj nebylo možné.

Kromě referenčního videa „xylophone.mpg“ byly pro testování kompatibility použity i další sekvence. Testovali jsme další kodeky (DivX, MPEG-2, nekomprimované video), nebyly zaznamenány potíže. Jediným známým nedostatkem, který jsme pozorovali, byla nestabilita aplikace v případě, že některý ze snímků zobrazoval pouze jednobarevný obraz, typicky černá scéna před začátkem sekvence. V případě metriky GBIM byla výsledkem hodnota NaN, v případě metriky Marziliano došlo k zacyklení programu. Dále byl program testován na kompatibilitu v rámci různých verzí prostředí Matlab, konkrétně verze R2009a/b, R2010a/b, a R2011. Nebyly zaznamenány žádné potíže a tedy by neměl být problém s provozem programu při podmínkách odlišných našim.

4 ODHAD PSNR PRO H.264 KÓDOVANÉ SEKVENCE

Dalším cílem práce byla implementace metriky pracující pouze na základě informací získaných z bitového toku. Taková metrika je poté sice zaměřena jen na konkrétní kodek, její výhoda však tkví v tom, že pro její aplikaci není nutné kompletní dekódování videa a vystačíme si pouze s výběrem pro nás zajímavých hodnot z bitového toku. Výpočet takové metriky je navíc zpravidla rychlejší, než metriky pracující v oblasti jednotlivých pixelů. Pro naši práci jsme se zaměřili na metriku pro sekvence kódované pomocí kodeku H.264/AVC.

4.1 H.264/AVC

Kodek H.264/AVC, nazývaný též MPEG-4 part 10 (označení ISO MPEG), je kodek vytvořený na základě práce skupiny JVT (Joint Video Team). Jedná se o následovníka velmi populárního kodeku MPEG-2, který je využíván na nosičích DVD. V současné době má kodek H.264/AVC velmi široké uplatnění, a to buď na discích Blu-ray, v mobilních zařízeních nebo v internetovém vysílání a vysílání s vysokým rozlišením (např. standardy DVB druhých generací). V následujících řádcích budou ve zkratce popsány jeho vlastnosti, které jsou důležité pro náš záměr.

Tak jako v předešlém standardu MPEG-2, i H.264/AVC využívá možnosti interpredikce a také predikce a kompenzace pohybu pomocí pohybových vektorů. Vylepšením zde jsou však nové intrapredikce, kdy mohou být predikovány bloky jednoho snímku i na základě hodnot sousedních bloků v rámci téhož snímku. V případě kompenzace pohybu došlo ke zefektivnění tohoto procesu a ke zvýšení přesnosti na $\frac{1}{4}$ pixelu. Pro nás jsou však zajímavé jiné změny. Tou hlavní je změna transformačního kódování. Zatímco předešlé standardy využívaly diskrétní kosinové transformace (DCT), H.264/AVC má k dispozici celočíselnou transformaci založenou na DCT. Transformace se zde dále počítá na bloky velikosti 4x4 (rozdílový blok). Pro stejnosměrné jasové koeficienty intra kódovaného bloku se dále používá Hadamardova transformace 4x4. Další pro nás podstatnou novinkou je tzv. deblocking filtr. Ten slouží ke zmírnění viditelnosti blokové struktury v obraze (blokových artefaktů). Dalšími vylepšeními jsou pak např. zavedení nových typů proužků (slices) SI a SP a užití entropického kódování CAVLC, případně CABAC, které však pro nás již nejsou příliš důležité. Více informací lze nalézt např. ve [24], případně v originálním doporučení ITU-T H.264, [25].

4.2 Získání informací z bitového toku

Základem pro návrh metriky pracující s informacemi z bitového toku je postup, kterým tyto informace získáme. Pro naše potřeby byla použita modifikovaná verze dekodéru JEG JM H.264/AVC, [26]. Tento kodek je referenční implementací algoritmu H.264/AVC v jazyce C přímo od ITU, zde v upravené verzi s možností generování

výstupního XML souboru, který může obsahovat předem zvolené parametry zjištěné z bitového toku.

V našem případě užíváme pouze dekodér `ldecod.exe`, který je volán způsobem `ldecod [-s] -i input.264 -o out.yuv [-r ref.yuv] [-xmltrace trace.xml]`. Zde `input.264` představuje vstupní soubor, `out.yuv` výstupní dekódovaná obrazová data, soubor `ref.yuv` případný referenční soubor pro přímý výpočet PSNR a `trace.xml` konečně jméno výstupního XML souboru. Volitelný přepínač `-s` umožňuje tzv. silent mód, kdy dekodér vypisuje do konzolového okna jen minimum informací.

4.3 Úprava dekodéru JEG JM H.264/AVC

Pro naše potřeby bylo nutné v dekodéru provést některé dílčí úpravy pro zefektivnění výkonu programu. V základním nastavení dekodér kromě generování XML souboru provádí i dekódování obrazových dat do YUV souboru, který je pro nás nepotřebný a tento proces tak zbytečně prodlužuje dekódování. Celkem bylo ve zdrojovém kódu dekodéru provedeno více než 40 změn. Pro přehlednost jsou ve zdrojovém kódu označeny pomocí komentáře `//xzacho04` a jsou tak pro případné zájemce snadno dohledatelné. Změny se týkají těchto souborů:

- `image.c`,
- `ldecod.c`,
- `macroblock.c`,
- `output.c`,
- `tracehelper.c`.

V základní verzi umožňuje dekodér vypsat do XML souboru hodnoty kvantizačních parametrů pro každý makroblok, jednotlivé hodnoty jasových a chrominančních transformačních koeficientů, vektory pohybu a základní pozici makrobloku ve snímku a jeho pořadové číslo ve snímku. Tyto informace jsou získány pro každý snímek, spolu s informacemi o typu snímku (případně proužku).

V literatuře lze nalézt, že PSNR je možné odhadnout na základě hodnot kvantizačního parametru QP a počtu nulových jasových transformačních koeficientů, např. v [27] a [28]. Z tohoto důvodu jsme se rozhodli zaměřit pozornost právě na tyto údaje, čemuž jsme přizpůsobili následné úpravy dekodéru. Hlavní úpravou bylo vypnutí vypisování chrominančních koeficientů a vektorů pohybu do XML souboru. Dále jsme deaktivovali dekódování obrazových dat a jejich ukládání do YUV souboru. Závěrečné úpravy se zaměřily na úpravu výpisu jasových transformačních koeficientů. Místo výpisu všech koeficientů pro daný makroblok jsme přistoupili k výpisu pouze počtu nulových koeficientů, což velmi urychlilo běh aplikace a mělo za následek i menší výslednou velikost generovaného XML souboru. Upraveno bylo formátování vypisovaných dat a též byla deaktivována tvorba logovacích souborů dekodéru. Tímto jsme zrychlili běh aplikace přibližně 10krát a dosáhli 290krát menšího výstupního souboru. Jak jednotlivé optimalizační kroky ovlivnily rychlost aplikace a velikost XML souboru lze vidět v následující tabulce.

Tab. 1 Vliv úprav zdrojového kódu na rychlost běhu dekodéru ldecod

Úprava zdrojového kódu	Výpočetní čas [s]	Velikost XML souboru [MB]
Bez úprav	352	552
Pouze jasové transf. koeficienty	260	380
Bez vektorů pohybu	220	315
Pouze počet nulových koeficientů	78	83
Optimalizace formátu výpisu	57	20
Deaktivace dekódování obrazu*	40	20
Další optimalizace formátu	35	1,9

Z tabulky vidíme, že došlo jak k výrazné úspoře výpočetního času tak i ke snížení velikosti výstupního XML souboru. V případě deaktivace dekódování obrazu logicky nedošlo ke snížení velikosti XML souboru ale nedále již nedocházelo ke generování výstupního YUV souboru, který měl v našem případě velikost přibližně 137 MB. Dodejme, že byla použita táž PC sestava, která je uvedena v kapitole 3.5. Jako testovací soubor byla použita sekvence s rozlišením 768x432 pixelů, 217 snímků a tomu odpovídající dobou trvání 8,7 s. Pro srovnání je v příloze D uvedena část výstupního XML souboru při nulových a maximálních úpravách zdrojového kódu dekodéru.

4.4 Návrh algoritmu pro odhad PSNR

Data z výstupního XML souboru již můžeme použít pro návrh algoritmu pro odhad PSNR H.264 kódovaných sekvencí. Pro analýzu souvislostí mezi kvalitou a parametry extrahovanými z bitového toku jsme použili tři sekvence, jejichž skutečné PSNR nám bylo známo. Jednalo se o sekvence s nízkou, střední a vysokou kvalitou, abychom mohli zachytit široké spektrum případů, které mohou nastat. Šlo o sekvence s rozlišením 768x432 a 250, případně 500 snímků. Více o použitých sekvencích a jejich původu bude řečeno v páté kapitole.

Pro odhad PSNR pro videa kódovaná pomocí H.264 existuje více přístupů, již byl zmíněn přístup, který navrhnul A. Eden v [27]. Zde předpokládal, že transformační koeficienty mají Laplaceovo rozložení. Na základě počtu nulových koeficientů určoval pravděpodobnost, že daný koeficient bude kvantován na nulovou hodnotu. Tím byl schopen určit parametry Laplaceova rozložení, díky čemuž mohl dále určit hodnotu PSNR ve spektrální oblasti, která díky platnosti Parsevalova teoremu odpovídá PSNR v oblasti prostorové. Podobně postupovali i autoři v [28] s tím rozdílem, že předpokládali Cauchyho distribuci střídavých transformačních koeficientů. Oba postupy jsou schopny dosahovat velmi vysoké korelace nad 90%. My se však pokusíme navrhnout přístup jednodušší, s tím, že se spokojíme i s menší korelací se skutečnými hodnotami PSNR. Na základě analýzy hodnot skutečných PSNR, hodnot kvantizačního parametru a počtu nulových transformačních koeficientů jsme vysledovali některé souvislosti, které posloužili jako vodítko pro návrh algoritmu odhadujícího PSNR.

Byly zjištěny následující vztahy: Pro vyšší hodnoty kvantizačního parametru QP byla ve většině případů nižší hodnota PSNR, tedy nižší kvalita. To je skutečnost, kterou

jsme očekávali a je plně logická. Vyšší hodnota QP znamená „hrubší“ kvantizaci (větší kvantizační krok), tedy více koeficientů bude rovno nule. Tím může dojít k degradaci obrazové informace. Počet nulových koeficientů má obdobnou souvislost, zde to však již rozhodně nemusí být pravidlem. Bylo možné vyzkoušet, že i při vyšších výskytech nulových koeficientů v makrobloku byla skutečná hodnota PSNR vyšší (tedy kvalitnější obraz), než v opačném případě.

Hodnoty QP a počtu nulových koeficientů jsme získávali pro každý makroblok. Vzhledem k výše uvedeným testovacím sekvencím s rozlišením 768x432 bodů to znamenalo 1296 makrobloků v každém snímku. PSNR se však počítá pro celý snímek, chtěli jsme tedy zmenšit počet hodnot aniž bychom přišli o informaci v nich obsaženou. V případě kvantizačního parametru to problém nebyl. Ačkoliv standard umožňuje rozdílnou hodnotu QP pro každý makroblok ve snímku, v reálné situaci tomu tak často nebývá a nejčastěji je použita jen jedna hodnota pro celý snímek. To je pro nás prvním zjednodušením. U počtu nulových transformačních koeficientů je situace rozdílná. Bylo by možné například uvažovat průměrnou hodnotu. Zde však narazíme na problém. Ve většině makrobloků je počet nulových koeficientů vyšší než 250 (z celkových 256). Použití pro algoritmus pouze jednoduchý průměr by nepřineslo výsledek, protože průměr (vyjádřený v celočíselném tvaru) by byl pro většinu snímků totožný a pohyboval by se v rozmezí 253 až 256. Zjistili jsme však, že zajímavou informací nám poskytuje nejmenší počet nulových koeficientů v makrobloku. Takový makroblok poté nese větší informaci o obrazovém obsahu. Navíc tento počet je v korelaci s vnímanou kvalitou, protože můžeme pozorovat, že pokud snímek obsahuje blok s malým počtem nulových koeficientů, je skutečné PSNR snímku často vyšší, než v případě snímku, který má stejný QP a stejný průměrný počet nulových koeficientů. Proto budeme pro vstup našeho algoritmu užívat právě hodnoty QP a průměrného a nejmenšího počtu nulových transformačních koeficientů v makrobloku pro každý snímek.

4.4.1 Sestavení algoritmu

Naším cílem bylo navrhnout jednoduchý algoritmus (vzorec), jehož vstupem budou výše uvedené parametry a jeho výstupem bude odhad hodnoty PSNR. Na základě analýzy souvislostí parametrů z bitového toku a reálné kvality jsme se pokusili o zjištění lineární souvislosti mezi všemi hodnotami. Dále bylo řečeno, že větší hodnota QP ve většině případů odpovídá nižší kvalitě. Tuto závislost jsme pro výpočet chtěli převést na opačnou, tedy aby hodnota, která bude vycházet z kvantizačního parametru byla přímo úměrná hodnotě PSNR. Proto jsme definovali hodnotu, kterou bychom si mohli označit jako QP_d (QP derived, odvozené). Vztah mezi QP a QP_d jsme definovali jako

$$QP_d = \frac{1}{\ln(QP)} . \quad (11)$$

To má za následek dvě změny: mezi QP_d a PSNR je nyní přímá úměra. Zmenšil se také rozsah hodnot, které může QP_d na rozdíl od QP nabývat. Také skoková změna QP má za následek menší poměrnou změnu QP_d . Takto upravený kvantizační parametr spolu s informací o průměrném a minimálním počtu nulových transformačních koeficientů a samozřejmě reálné hodnoty PSNR posloužily jako vstup do regresní funkce programu Matlab. Na základě předchozích úvah byla zvolena lineární regrese. Regresní funkce byla poté volána následujícím způsobem:

```
coeff = regress(PSNR, [QP_d meanZ minZ]).
```

Zde *PSNR* je vektor reálných hodnot PSNR, *QP_d* představuje vektor modifikovaných kvantizačních parametrů a proměnné *meanZ* a *minZ* jsou vektory průměrných resp. minimálních počtů nulových koeficientů (*Z* – zero coefficients), to vše pro každý snímek. Výsledek je uložen do proměnné *coeff*, která reprezentuje jednotlivé koeficienty lineární regrese. Na základě tohoto můžeme určit vztah pro odhadnuté PSNR jako

$$PSNR_{estimated} = 285,8914QP_d - 0,2652meanZ + 0,0995 \min Z \quad (12)$$

Tato rovnice pak bude východiskem pro implementaci celé metriky v prostředí Matlab.

4.5 Ověření a ladění algoritmu

Schopnost odhadnout správně PSNR byla dále testována. Pro zjednodušení není výsledkem metriky hodnota kvality pro každý snímek, jako tomu bylo v předešlých pixelově-orientovaných metrikách, ale zobrazujeme rovnou celkové skóre za celý snímek. Zde se dostáváme k dalšímu zjištění. Algoritmus není schopen celkové PSNR určit správně v celém rozsahu kvantizačních parametrů. U vyšších hodnot QP (nad 37) docházelo často k nadhodnocení. Proto jsme přistoupili k dílčím úpravám.

V základním nastavení bylo výsledné námi odhadnuté PSNR počítáno jako průměr přes všechny snímky. Zjistili jsme však, že pro různé intervaly parametru QP odpovídá skutečné hodnotě jiná metoda určení výsledného skóre. Výsledkem experimentu bylo rozdělení QP do tří intervalů, v nichž se výpočet výsledného PSNR za celou sekvenci počítal s mírnými odlišnostmi. V případě průměrování můžeme vlastně říci, že pro výpočet celkového skóre použijeme vztah 12 s tím, že za jeho jednotlivé proměnné dosadíme odpovídající průměrné hodnoty za celý snímek. Bylo zjištěno, že pro různé intervaly QP je výhodnější použít pro každou proměnnou jiný přístup. Jakým způsobem bylo pro dané proměnné a intervaly parametru QP výsledné skóre počítáno, uvádí následující tabulka.

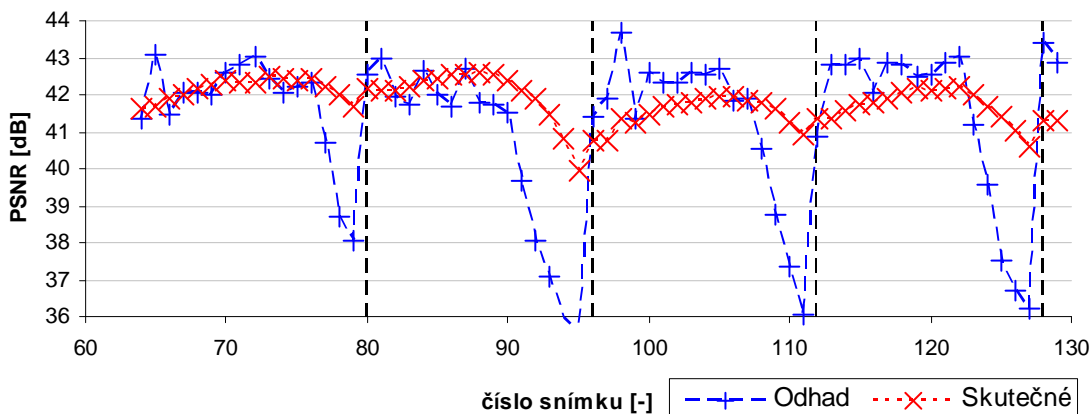
Tab. 2 Metoda výběru proměnných pro různé intervaly QP

	Metoda výběru		
Interval QP	QP	meanZ	minZ
do 30	průměr	průměr	modus
od 31 do 36	modus	průměr	modus
nad 36	průměr	průměr	průměr

Takto počítané výsledné PSNR pro celý snímek již vykazuje lepší výsledky.

Další zajímavou vlastností, kterou jsme zjistili, bylo, že spolehlivost odhadu pro snímek typu P klesá s tím, jak se daný P snímek ve skupině snímků přibližuje k následujícímu I snímku. Situaci můžeme vidět na obrázku 14.

Srovnání skutečného a odhadnutého PSNR

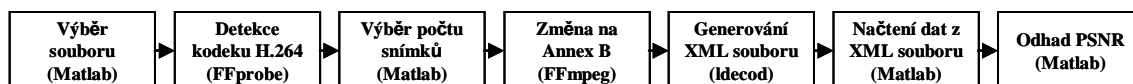


Obr. 14 Srovnání skutečného a odhadnutého PSNR, ověření algoritmu

Na obrázku je zobrazen průběh skutečného a odhadnutého PSNR pro sekvenci *station9*, více o ní v kapitole pojednávající o sestavení databáze videosekvencí. Vidíme zde výřez pro snímky číslo 64 – 130. Snímky číslo 64 a 130 jsou zároveň I snímky, další I snímky jsou v grafu znázorněny svislou černou čárkovanou čarou. Můžeme pozorovat, že po I snímku je odhad PSNR poměrně dobrý. Jak se však budeme pohybovat k dalšímu I snímku, spolehlivost rychle klesá a rozdíl se pohybuje v absolutních hodnotách až kolem 4 dB. Pro samotné I snímky je pak odhad zpravidla nepřesnější. Tuto skutečnost by bylo možné vyřešit například zavedením jistého váhování, které by bralo v potaz typ a pozici aktuálního snímku ve skupině snímků. To by však vedlo k narůstání složitosti našeho algoritmu, a proto se pro naše potřeby spokojíme s výše uvedenými principy, kde budeme počítat pouze celkovou hodnotu PSNR za celou sekvenci.

4.6 Implementace navržené metriky v prostředí Matlab

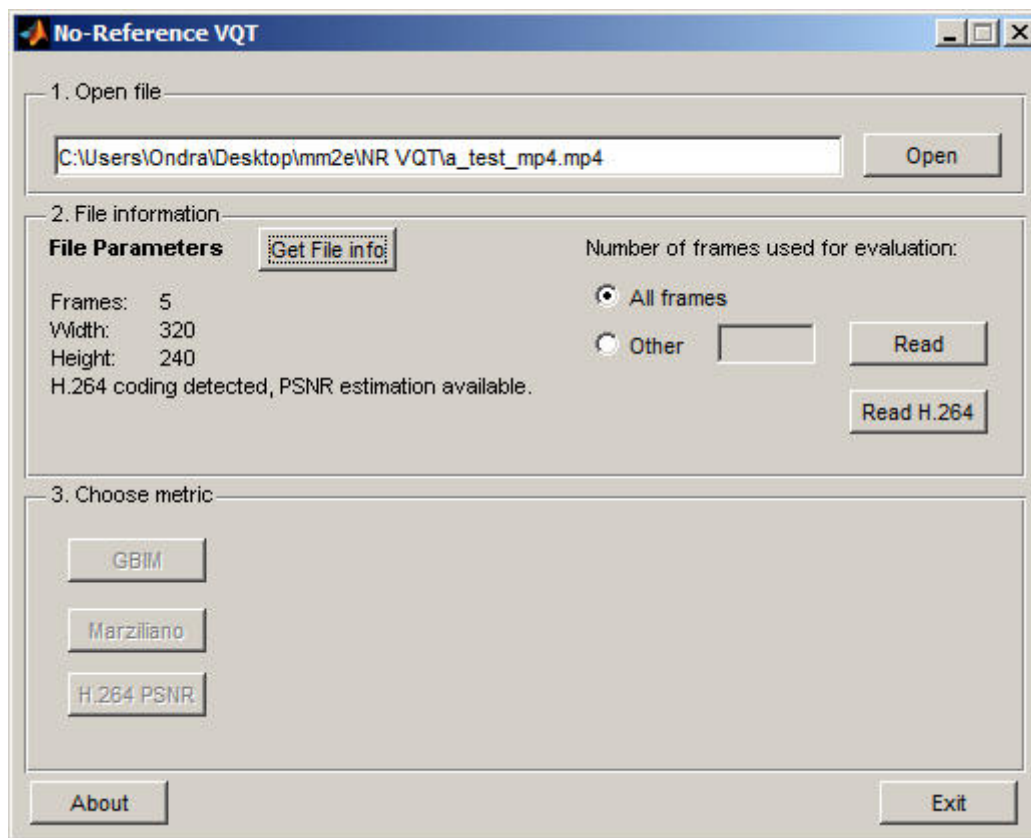
Již bylo vysvětleno, jak lze získat informace z bitového toku videa kódovaného kodekem H.264 a jak lze z těchto dat odhadnout celkové PSNR. Následující podkapitola osvětlí, jak byla celá tato procedura implementována do již dříve uvedené aplikace No-Reference Video Quality Tool (NR VQT). Základem je výše uvedený dekodér a využívat budeme opět balík FFmpeg. Schéma volání jednotlivých programů a funkcí je vidět na následujícím obrázku.



Obr. 15 Schéma použitých prostředků pro odhad PSNR

Povšimněme si, že i v případě metriky pro odhad PSNR na základě informací z bitového toku používáme balík FFmpeg. V tomto případě však nedochází k dekodování na obrazová data, ale ke změně formátu dat na tzv. byte-stream formát popsany v doplňku Annex B standardu H.264. To je z důvodu, že právě tento formát je

vyžadován jako vstup pro dekodér ldecod. Navíc užití FFmpeg nám umožní přidat uživateli volbu počtu snímků i v případě této metriky. Na základě těchto změn došlo i k dílčím úpravám v samotném kódu aplikace NR VQT, stejně jako v jejím grafickém prostředí. Upravené GUI můžeme vidět na obrázku 16. Obrázek je pro situaci, kdy byl jako vstupní soubor vybrán video soubor kódovaný pomocí H.264.



Obr. 16 Grafické rozhraní NR VQT, změny pro H.264 metriku

První změnou, které si můžeme všimnout, je hláška o tom, že vstupní soubor je formátu H.264 a tudíž je dostupná funkce odhadu PSNR. Dále došlo k přidání dvou nových ovládacích prvků, a to tlačítek „Read H.264“ a „H.264 PSNR“. Jak je v aplikaci zvykem, tato tlačítka se zpřístupní jen v případě, že byl skutečně detekován kodek H.264, případně pokud byl soubor i úspěšně dekodován (vygenerován soubor XML). Tlačítko „Read H.264“ volá nejprve FFmpeg, a to se syntaxí

```
ffmpeg -i <input.mp4> -vcodec copy -vframes <nFrames> -bsf:v
h264_mp4toannexb -an -y annexb.h264,
```

kde „input.mp4“ označuje vstupní soubor, „nFrames“ udává vybraný počet snímků a „annexb.h264“ je fixně nastavené jméno pro výstupní soubor z FFmpeg. Dále je volán dekodér ldecod a přímo z bitového toku jsou extrahovány hodnoty QP a počtu nulových transformačních koeficientů, které jsou následně uloženy v souboru XML. Syntaxe dekodéru ldecod je následující:

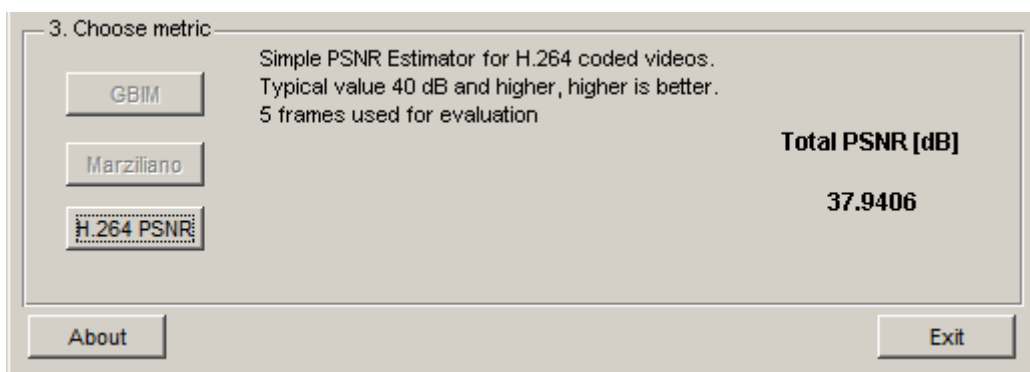
```
ldecod -i annexb.h264 -xmltrace trace.xml.
```

Soubor „annexb.h264“ je nám už známý a soubor „trace.xml“ je fixně nastavený název pro výstupní XML soubor dekodéru.

Nyní již můžeme data načíst v prostředí Matlab. K tomu využíváme zejména funkce `xml2struct.m` autora W. Falkany, [29], a námi vytvořenou funkci `xml2psnr.m`. Funkce `xml2struct.m` nám umožňuje jednoduché načtení dat z XML souboru do datové struktury programu Matlab. Díky tomu můžeme poté jednoduše přistupovat k jednotlivých proměnným QP a počtu nulových transformačních koeficientů. K úpravě dat a samotnému výpočtu PSNR slouží funkce `xml2psnr`. Volání funkce je následující:

```
PSNR_estimated=xml2psnr(xmlfile, W, H),
```

kde `xmlfile` je cesta ke XML souboru a `w` a `H` jsou šířka a výška obrazu. Ty zde potřebujeme, abychom mohli určit počet makrobloků v obraze a dopředu alokovat paměť pro jednotlivé parametry, což urychlí běh programu. V jádru funkce je volána výše zmíněná funkce pro načtení XML souboru jako datové struktury, dále dochází k extrahování dat ze struktury do jednotlivých vektorů parametrů a výpočet výsledného odhadnutého PSNR s ohledem na hodnotu QP (viz předchozí podkapitola). Na obrázku 17 můžeme vidět zobrazení výsledku v případě odhadu PSNR.



Obr. 17 Zobrazení výsledků metriky pro odhad PSNR H.264 kódovaných sekvencí

5 DATABÁZE VIDEOSEKVENČÍ

Aplikace byla dále použita k vedení objektivních testů kvality videosekvencí. Pro tyto potřeby byla sestavena databáze kódovaných videí. Vzhledem k tomu, že jsme naše výsledky chtěli srovnávat i s výsledky testů subjektivních, využili jsme k tvorbě naší databáze už existujících databází, které toto hodnocení obsahují. Byly vybrány databáze LIVE Video Quality Database, [30], [31], vytvořená na Universitě Texas a dále databáze vytvořená skupinou VQEG v rámci řešení projektu FRTV Phase I, [32], což byl projekt zaměřený na hodnocení FR a NR metrik pro měření kvality videosekvencí se standardním rozlišením.

5.1 Databáze VQEG

Databáze skupiny VQEG vznikla pro testování účinnosti 10 návrhů algoritmů pro metody hodnotící kvalitu videa. Obsahuje celkem 24 sekvencí ve standardním rozlišení, 12 s 625 řádky a dalších 12 s 525 řádky. Sekvence jsou od různých zdrojů, např. EBU, italské veřejnoprávní televize RAI apod. Každá sekvence je v databázi dostupná v 16 různých kvalitách (označovaných HRC - hypothetical reference circuit, hypotetický referenční obvod). To dává celkovou velikost databáze 384 sekvencí v různých kompresích. Pro naše potřeby bylo vybráno po třech sekvencích z každé skupiny zdrojových sekvencí a každá sekvence po 8 různých kvalitách HRC, celkem tedy 48. HRC všeobecně udává konkrétní typ komprese (algoritmus, bitová rychlost), my se zaměřili pouze na sekvence kódované algoritmem MPEG-2, kde lze při nedostatečných bitových rychlost poměrně dobře pozorovat blokové artefakty. Databáze však obsahuje i sekvence s kódováním H.263 [32].

Pro naše potřeby byla vybrána HRC uvedená v následující tabulce:

Tab. 3 Popis vybraných HRC databáze VQEG, [32]

HRC	Bitová rychlost [Mb/s]	Poznámka
2	12-19	Vzorkování 4:2:2
5	8 a 4,5	Použití dvou enkodérů za sebou
7	6	
8	4,5	Kompozitní PAL/NTSC
9	3	
10	4,5	
11	3	S chybami
12	4,5	S chybami

Vidíme, že námi vybraná HRC pokrývají poměrně široká nastavení. Pro kódování bylo použito kodeku MPEG-2 v hlavním profilu a hlavní úrovni. Dále jsme zmínili, že co se zdrojových sekvencí týče, vybrali jsme celkem 6 sekvencí. Sekvence mají dobu trvání

8 s a rozlišení 720x576, resp. 720x486 bodů. První snímky jednotlivých sekvencí jsou součástí přílohy. Jejich vlastnosti uvádí tabulka 4.

Tab. 4 Popis vybraných sekvencí VQEG, [32]

Číslo sekvence	Formát	Název	Charakter
2	625p	Barcelona	Syté barvy, maskovací efekt
3	625p	Harp	Syté barvy, přiblížení
7	625p	Fries	Pleťové barvy, film
16	525p	Betes pas betes	Syntetická scéna, pohyb, střih
18	525p	Autumn leaves	Příroda, přiblížení, vodopád
21	525p	Susie	Pleťové barvy

Již jsme zmínili, že součástí databáze jsou zároveň výsledky subjektivních testů. Autoři prováděli testy metodou DSCQS a pro každou sekvenci a každé HRC je poté k dispozici hodnota DMOS (differential mean opinion score, diferenční průměrné hodnocení). To nám poslouží ke srovnání výsledků s našimi objektivními testy. Vzhledem k tomu, že databáze neobsahuje sekvence kódované kodekem H.264, budeme na tyto sekvence aplikovat pouze metriky GBIM a Marziliano.

5.2 Databáze LIVE

Databáze LIVE Video Quality Database vznikla jako náhrada databáze od skupiny VQEG reflektující moderní požadavky na testované videosekvence. Nabízí celkem 10 zdrojových sekvencí, celkem 4 různé typy rušení (komprese MPEG-2 a H.264 a 2 typy rušení vlivem ztráty paketů při přenosu), každé rušení ve třech až čtyřech úrovních. To dává dohromady celkový rozsah 160 sekvencí (včetně reference). My jsme využili pouze rušení způsobené kompresí H.264 a použili jsme 8 sekvencí. Sekvence mají rozlišení 768x432 bodů, trvání 10 s a 25, příp. 50 snímků za vteřinu. Použité sekvence s krátkým popisem zaznamenává následující tabulka [30].

Tab. 5 Popis vybraných sekvencí databáze LIVE, [30]

Název sekvence	Zkratka	FPS	Popis
Pedestrian Area	pa	25	Statická kamera, lidé procházející pěší zónou
Park Run	pr	50	Pohyblivá kamera, postava běžící parkem podél řeky
River Bed	rb	25	Statická kamera, koryto řeky, pohybující se voda
Rush Hour	rh	25	Statická kamera, dopravní špička, pohyb
Sunflower	sf	25	Statická kamera, detail na slunečnici a včelu
Shields	sh	50	Pohyblivá kamera a přiblížení, drobné detaily
Station	st	25	Statická kamera, lidé pohybující se v kolejišti
Tractor	tr	25	Pohyblivá kamera, traktor jedoucí po poli

Jednotlivá nastavení kodéru H.264 pro každou sekvenci jsou označena přídatnými čísly 9 – 12. To je vlastně ekvivalentem HRC u předchozí databáze. Sekvence byly kódovány s datovými toky 200 kb/s až 5 Mb/s, s tím, že pro sekvence s příznakem 9 byl použit nejvyšší datový tok a s příznakem 12 nejnižší. Dodané soubory jsou přímo v byte-stream formátu podle Annex B, což je pro nás výhodou, vzhledem k tomu, že Annex B formát vyžaduje i námi používaný dekodér ldecod. První snímky z jednotlivých zdrojových sekvencí jsou opět součástí přílohy. Také LIVE databáze obsahuje informace o vnímané subjektivní kvalitě pomocí hodnoty DMOS, která nám i zde poslouží pro srovnání s výsledky našich objektivních testů. Autoři použili metodu Single Stimulus (tedy s jednou sekvencí) a pozorovatel kvalitu hodnotil v daném spojitěm rozsahu. Celkem se testů účastnilo 38 pozorovatelů, nedůvěryhodní pozorovatelé byli z testu vyřazeni [30].

Pro návrh algoritmu pro odhad PSNR byly použity sekvence „Station9“, „PedestrianArea10“ a „ParkRun12“, čímž jsme pokryli oblast kvalitních, středně kvalitních a zaručených sekvencí.

5.3 Výsledná databáze sekvencí

Výsledná databáze tedy čítá 6 zdrojových sekvencí kódovaných v 8 hladinách nastavení enkodéru MPEG-2 a 8 zdrojových sekvencí kódovaných ve 4 hladinách nastavení kodéru H.264. Celkem tedy máme k dispozici 80 videosekvencí ve standardním rozlišení. Všechny tyto sekvence podrobíme testům kvality pomocí metrik GBIM a Marziliano. Celkem 32 sekvencí H.264 podrobíme i metrice pro odhad PSNR na základě parametrů bitového toku.

Uvedené metriky mohou být použity bez problémů i pro HD sekvence. Nicméně rozsáhlejší testování HD sekvencí by bylo výpočetně náročné a proto HD databáze nebyla sestavena a spokojíme se se srovnáním účinnosti metrik pouze na základě výše uvedených SD sekvencí.

6 OBJEKTIVNÍ TESTY

Vytvořená databáze byla podrobena testům kvality. Vzhledem k velkému rozsahu databáze nebylo k hodnocení kvality použito grafického rozhraní programu, ale byly vytvořeny funkce dávkového zpracování. Tyto dávkové funkce jsou tři, jedna funkce, která dokáže přímo zpracovat YUV soubory od skupiny VQEG, dále funkce, která jako vstupní data potřebuje pouze jasovou složku z YUV souboru, a do třetice funkce pro odhad PSNR sekvencí kódovaných pomocí algoritmu H.264. Příklad volání dávkové funkce pro šedotónové sekvence a metriku GBIM by byl následující:

```
batch_vqeg_yuv('sekvence.txt', 768, 432, 'g')
```

Vstupními parametry těchto funkcí jsou cesta k textovému souboru, který obsahuje názvy zpracovávaných souborů a rozlišení sekvencí. Pro testy GBIM a Marziliano se dále parametrem předává, která metrika se má vypočítat. Po dokončení výpočtu pro danou sekvenci jsou výsledky za všechny její snímky uloženy do souboru CSV. Tyto výsledky byly opět přepočítány na celkové skóre pomocí Minkowskeho sumace.

6.1 Pixelově-orientované metriky, databáze LIVE

Jak již bylo několikrát řečeno, z databáze LIVE jsme pracovali pouze se sekvencemi kódovanými kodekem H.264. Vzhledem k tomu, že tento kodek obsahuje filtr pro potlačení blokové struktury, lze očekávat, že výsledky metriky GBIM nemusejí příliš vypovídat o skutečné kvalitě. Navíc, metrika je nastavena pro hledání blokových artefaktů na hranicích bloků o velikost 8x8, kdežto transformace u H.264 se provádí většinou na blocích o velikosti 4x4.

Výsledky byly nicméně překvapivé. Přestože metrika většinou určila nízkou úroveň blokových artefaktů, bylo možné rozlišit kvalitu mezi jednotlivými úrovněmi komprese a navíc trend změny hodnoty GBIM byl v souladu s trendem hodnoty DMOS subjektivního hodnocení.

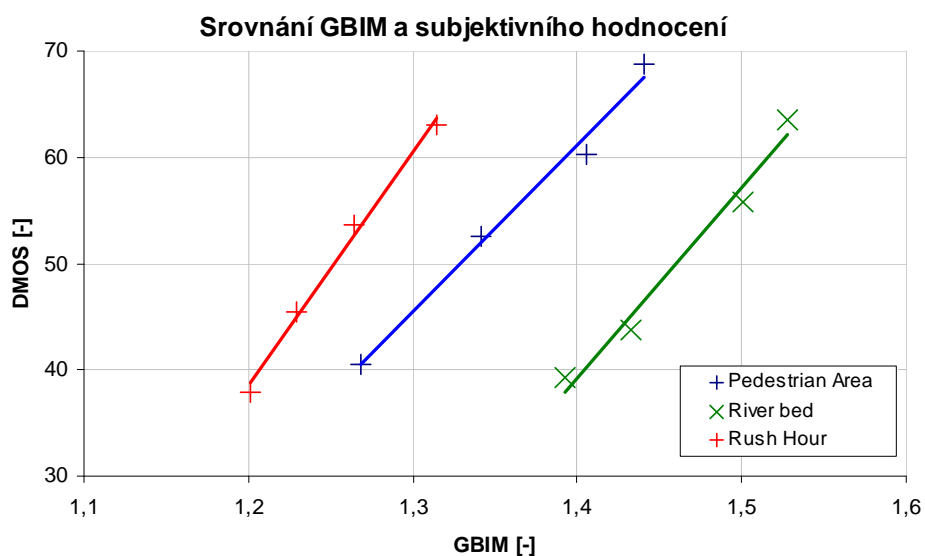
Metrika Marziliano pro rozmazání obrazu též poskytovala poměrně slušné výsledky. Výjimkou je sekvence „River Bed“, kde byl podle metriky opačný trend změny kvality, než byl určen na základě subjektivních testů.

K tomuto měření je nutno dodat ještě jednu poznámku. Hodnota kvality sekvencí se snímkovou rychlostí 50 fps po prvních testech neodpovídala. To bylo způsobeno tím, že sekvence obsahovaly na začátku dva šedé snímky a některé sekvence obsahovaly šedé snímky i na svém konci (např. posledních 10 snímků). Metrika GBIM totiž selhává pro obraz s jednolitou barevnou plochou a ani metrika Marziliano není schopna poskytnout relevantní výsledky. Tím by bylo celkové skóre nadhodnoceno a tak byly tyto šedé snímky při počítání celkového skóre vyřazeny.

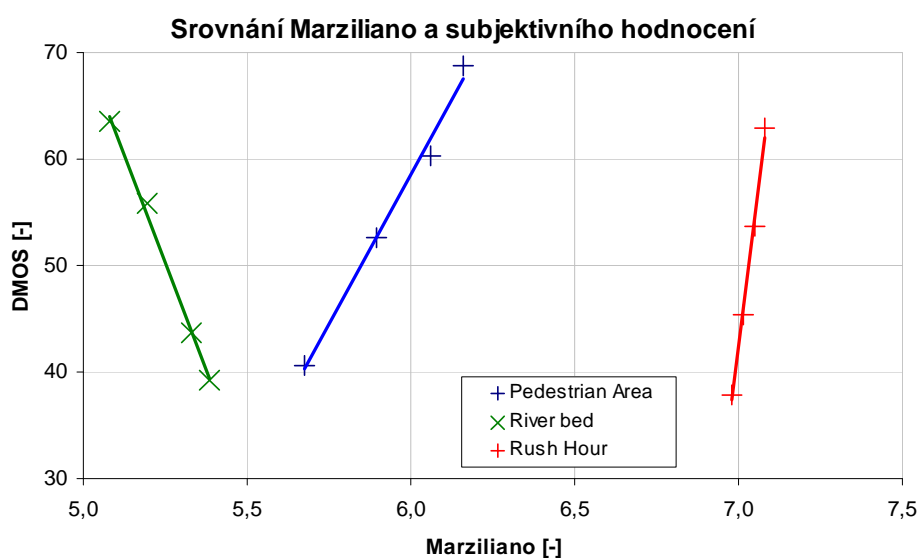
Na následujících grafech můžeme vidět srovnání vypočítané hodnoty GBIM a Marziliano ve srovnání s hodnotou DMOS. Vyšší hodnota DMOS, stejně jako vyšší hodnota koeficientů našich metrik značí méně kvalitní obrázek.

Tab. 6 Hodnoty kvality, databáze LIVE, sekvence (pa), (rb) a (rh)

Pedestrian Area			River Bed			Rush Hour		
DMOS [-]	GBIM [-]	Marz. [-]	DMOS [-]	GBIM [-]	Marz. [-]	DMOS [-]	GBIM [-]	Marz. [-]
40,56	1,27	5,67	39,20	1,39	5,39	37,87	1,20	6,98
52,61	1,34	5,89	43,68	1,43	5,33	45,44	1,23	7,01
60,25	1,41	6,06	55,86	1,50	5,19	53,63	1,26	7,05
68,72	1,44	6,16	63,58	1,53	5,08	62,99	1,31	7,08



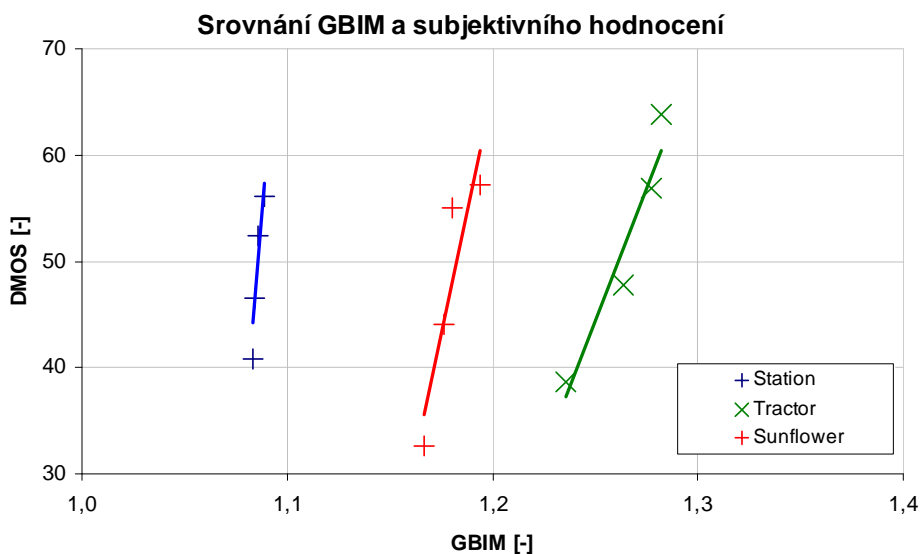
Obr. 18 Srovnání GBIM a DMOS, databáze LIVE, sekvence (pa), (rb) a (rh)



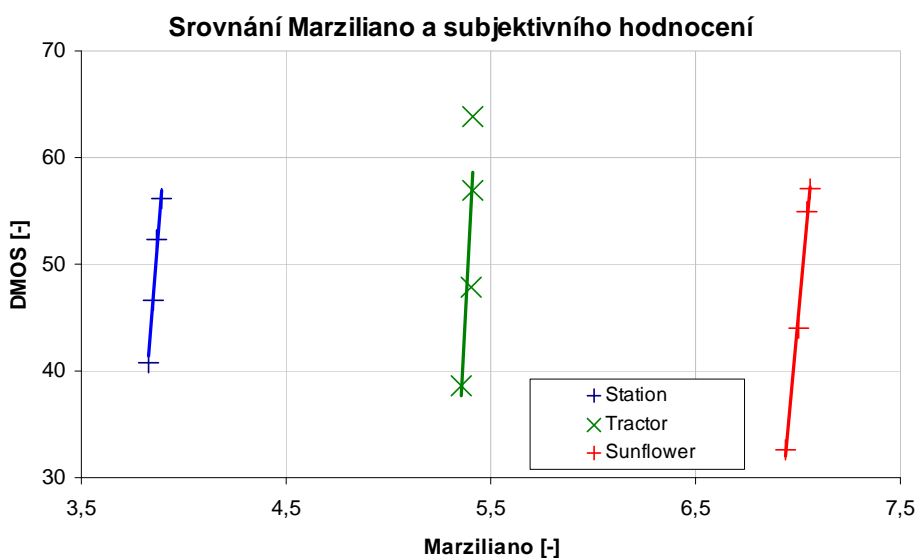
Obr. 19 Srovnání Marziliano a DMOS, databáze LIVE, sekvence (pa), (rb) a (rh)

Tab. 7 Hodnoty kvality, databáze LIVE, sekvence (sf), (st) a (tr)

Sunflower			Station			Tractor		
DMOS [-]	GBIM [-]	Marz. [-]	DMOS [-]	GBIM [-]	Marz. [-]	DMOS [-]	GBIM [-]	Marz. [-]
32,60	1,17	6,94	40,77	1,17	3,83	38,67	1,17	5,36
44,02	1,18	7,00	46,56	1,18	3,85	47,77	1,18	5,40
54,94	1,18	7,04	52,33	1,18	3,87	56,91	1,18	5,41
57,15	1,19	7,06	56,08	1,19	3,89	63,80	1,19	5,41



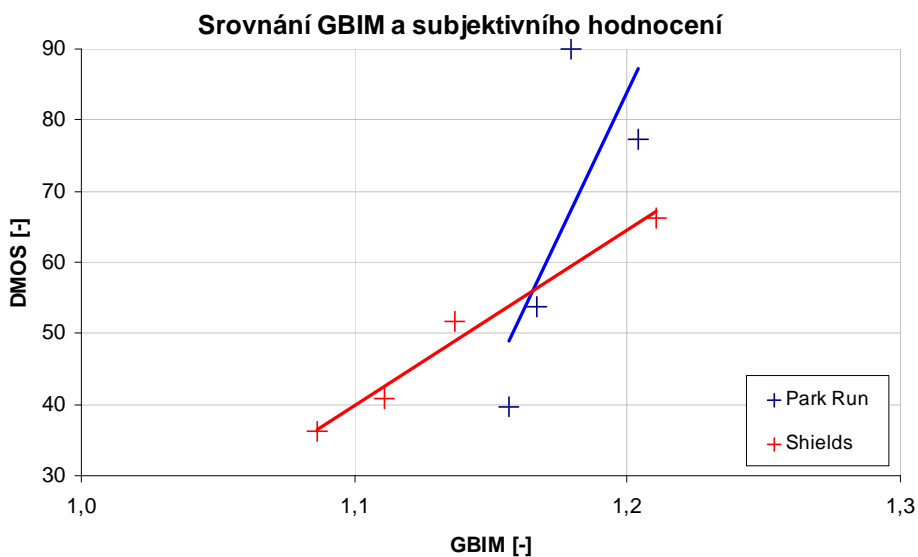
Obr. 20 Srovnání GBIM a DMOS, databáze LIVE, sekvence (sf), (st) a (tr)



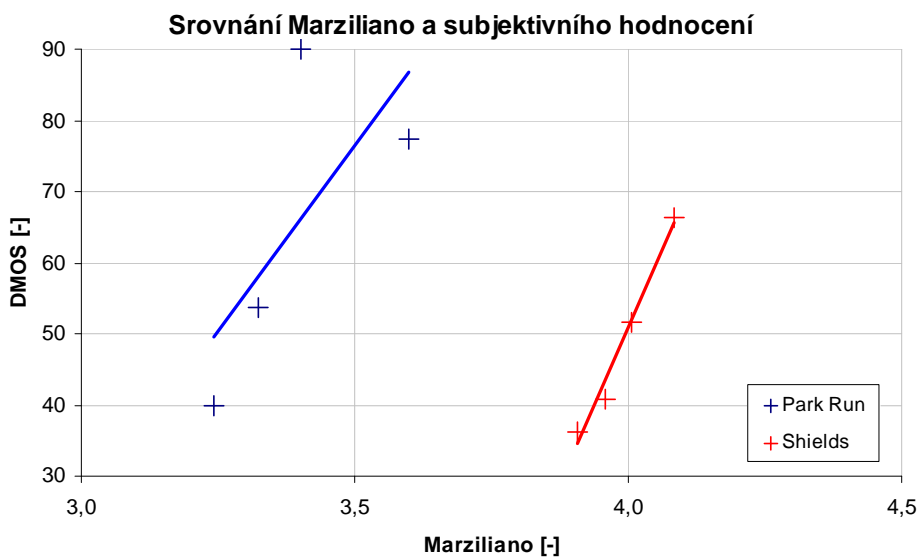
Obr. 21 Srovnání Marziliano a DMOS, databáze LIVE, sekvence (sf), (st) a (tr)

Tab. 8 Hodnoty kvality, databáze LIVE, sekvence (pr) a (sh)

Park Run			Shields		
DMOS [-]	GBIM [-]	Marz. [-]	DMOS [-]	GBIM [-]	Marz. [-]
39,81	1,16	3,24	36,24	1,09	3,91
53,76	1,17	3,32	40,80	1,11	3,96
89,89	1,18	3,40	51,62	1,14	4,00
77,25	1,20	3,60	66,32	1,21	4,08



Obr. 22 Srovnání GBIM a DMOS, databáze LIVE, sekvence (pr) a (sh)



Obr. 23 Srovnání Marziliano a DMOS, databáze LIVE, sekvence (pr) a (sh)

Na předešlých grafech jsme se snažili ukázat, že námi zjištěné výsledky objektivního hodnocení kvality odpovídají testům subjektivním. Vidíme, že sice nelze přesně definovat, např. jaká hodnota GBIM bude vždy odpovídat konkrétní subjektivní kvalitě, nicméně trend změny kvality odpovídá u obou typů hodnocení. Dále vidíme, že při jistém zjednodušení bychom mohli souvislost mezi objektivním a subjektivním hodnocení považovat za lineární, nicméně vždy jen v rámci jedné sekvence.

Vidíme také, že pro sekvenci „River Bed“ trend v případě metriky Marziliano neodpovídá. Jednalo se o sekvenci s poměrně velkými detaily a rychlými změnami (pohyb vln na vodě). Je možné, že při vyšších bitových rychlostech byl použit filtr pro potlačení blokové struktury více, než při nižších, což mohlo způsobit následné rozmazání obrazu.

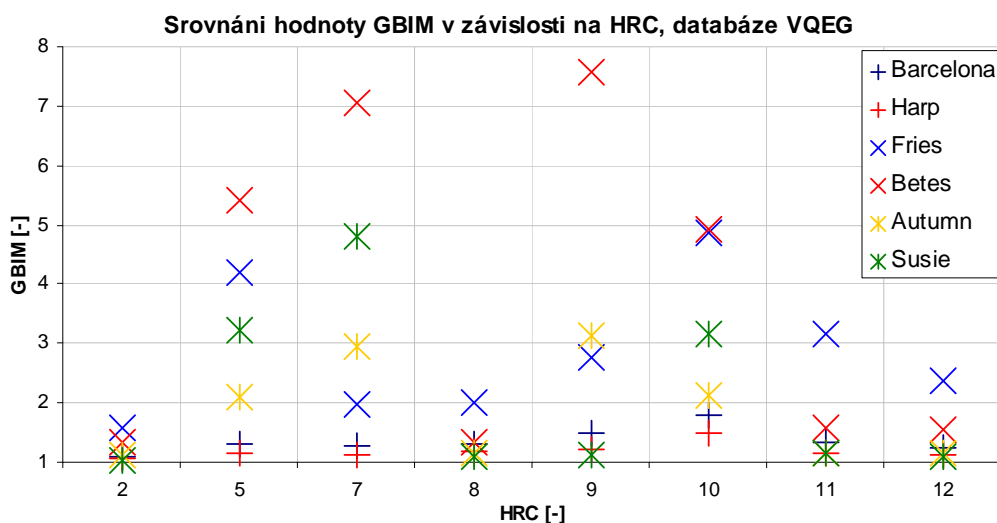
6.2 Pixelově-orientované metriky, databáze VQEG

Pro hodnocení výsledků sekvencí z databáze VQEG jsme použili odlišný přístup. Jednotlivá HRC totiž nemají nutně postupně se zhoršující kvalitu, jak tomu bylo v případě databáze LIVE, a proto by bylo obtížné definovat trend změny kvality. Níže uvedené grafy zobrazují závislost jednotlivých hodnocení kvality (GBIM, Marziliano a DMOS) na použitých HRC. Budeme tak srovnávat, jestli kvalita pro daná HRC odpovídá napříč metrikami. Pro některé sekvence jsou naše výsledky zcela odlišné oproti subjektivním testům. To může být např. způsobeno některými problémy samotné databáze, o čemž budeme ještě hovořit, případně tím, že daná sekvence je v jednotlivých snímcích sice skutečně nekvalitní, ale při subjektivních testech nemá pozorovatel možnost toto rušení zrakem postřehnout.

Graf na obr. 24 zobrazuje závislost metriky GBIM na jednotlivých HRC. Povšimněme si, že hodnota GBIM se pohybuje i kolem hodnoty 8. Dříve (a také v programu NR VT) jsme zmínili, že typická hodnota GBIM se pohybuje v rozsahu 1-2. To je však hodnota typická pro ne příliš zarušené sekvence. Pro sekvence se zřetelnými blokovými artefakty tato hodnota rychle roste nad hodnotu 2 a výše.

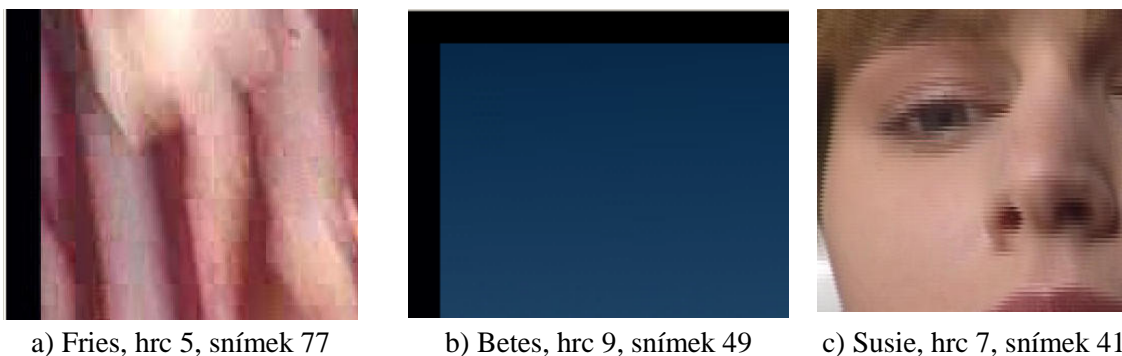
Tab. 9 Hodnoty metriky GBIM v závislosti na HRC, databáze VQEG

HRC	Barcelona	Harp	Fries	Betes	Autumn	Susie
2	1,09	1,06	1,57	1,35	1,13	1,04
5	1,30	1,16	4,20	5,40	2,10	3,22
7	1,26	1,11	1,97	7,04	2,96	4,80
8	1,31	1,18	2,02	1,34	1,14	1,08
9	1,48	1,21	2,77	7,57	3,12	1,11
10	1,80	1,50	4,86	4,93	2,13	3,16
11	1,33	1,15	3,17	1,59	1,15	1,16
12	1,23	1,12	2,36	1,55	1,15	1,08



Obr. 24 Srovnání hodnoty GBIM v závislosti na HRC, databáze VQEG

Z grafu vidíme, že v případě HRC 2 dosahuje metrika GBIM pro všechny sekvence hodnoty maximálně 1,6. To značí, že všechny sekvence jsou kvalitní bez viditelné blokové struktury. To je i v souladu s tím, že HRC 2 značí sekvence kódované s bitovým tokem 12-19 Mb/s. Nejlepších výsledků přes všechna HRC dosahují sekvence „Barcelona“ a „Harp“. Vidíme také, že prakticky pro všechna HRC dosahuje nejhorší kvality sekvence „Betes“. To není ani tak způsobeno tím, že by skutečně obsahovala tak výraznou blokovou strukturu, ale nešvarem celé databáze VQEG. Podle toho, jak byly konkrétní sekvence kódované, obsahují sekvence černé okraje po stranách. V případě sekvence „Betes“ je to okraj přímo o šířce 8 pixelů. Ze své podstaty v tomto případě metrika GBIM selhává, protože okraj snímku chápe jako velkou změnu jasu na rozhraní a tedy blokový artefakt. Je-li toto přítomné v celé sekvenci, jsou tím velmi výrazně ovlivněny výsledky. Dalším nešvarem databáze je prokládané řádkování. Ačkoliv jsme se snažili vybírat sekvence s co nejméně viditelným řádkovým prokladem, můžeme toto pozorovat, zejména u sekvence „Susie“. V případě sekvence „Fries“ je vysoká hodnota GBIM napříč všemi HRC opodstatněná, jelikož sekvence skutečně obsahuje výrazně viditelnou blokovou strukturu. Detaily těchto chyb můžeme vidět na obrázku.

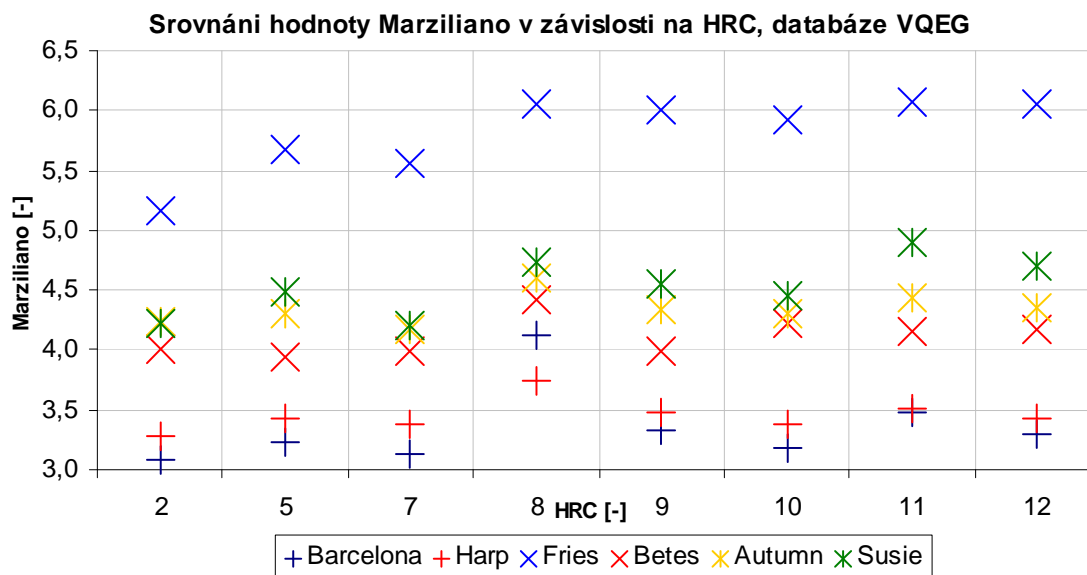


Obr. 25 Detaily některých sekvencí, databáze VQEG

Tab. 10 Hodnoty metriky Marziliano v závislosti na HRC, databáze VQEG

HRC	Barcelona	Harp	Fries	Betes	Autumn	Susie
2	3,08	3,28	5,17	4,00	4,23	4,23
5	3,23	3,42	5,67	3,95	4,31	4,49
7	3,12	3,37	5,55	3,99	4,17	4,20
8	4,12	3,74	6,06	4,42	4,61	4,73
9	3,33	3,48	6,00	4,00	4,35	4,55
10	3,18	3,38	5,92	4,22	4,30	4,45
11	3,47	3,51	6,07	4,15	4,43	4,91
12	3,30	3,43	6,06	4,17	4,36	4,70

V případě metriky Marziliano je situace obdobná, s tím rozdílem, že zde nelze jasně určit HRC s nejvyšší kvalitou. Opět se jako nejkvalitnější jeví sekvence „Barcelona“ a „Harp“. U sekvence „Fries“ se projevil její další, na první pohled nepatrný, nešvar, a to její značná neostrost. Při prohlížení sekvence snímek po snímku vidíme velmi neostře kontury a rozmazaný pohyb. Hodnoty jsou graficky zpracovány na obrázku 26.



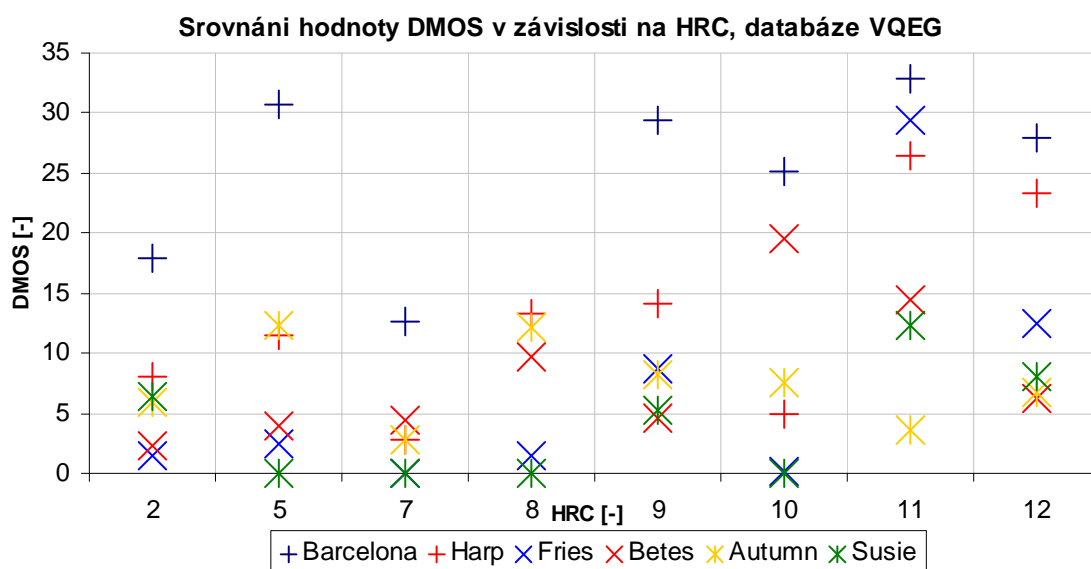
Obr. 26 Srovnání hodnoty Marziliano v závislosti na HRC, databáze VQEG

Na základě grafu však můžeme konstatovat, že vlivem rozdílného stupně komprese se hodnota získaná metrikou Marziliano mění jen mírně a je převážně daná obsahem samotné sekvence.

Poslední graf s tabulkou zobrazují vliv HRC na výsledné subjektivní hodnocení. Zde ještě dodejme, že hodnota DMOS pro sekvence databáze VQEG je získána jiným způsobem, než v případě DMOS u sekvencí databáze LIVE a pohybují se v jiném rozsahu.

Tab. 11 Hodnoty DMOS v závislosti na HRC, databáze VQEG

HRC	Barcelona	Harp	Fries	Betes	Autumn	Susie
2	17,86	8,10	1,40	2,34	5,96	6,41
5	30,79	11,57	2,43	3,90	12,36	-0,66
7	12,70	2,87	-1,79	4,39	2,85	-2,98
8	38,26	13,31	1,50	9,69	12,15	-2,09
9	29,43	14,15	8,66	4,63	8,27	5,30
10	25,17	4,87	0,09	19,47	7,64	-1,06
11	32,78	26,46	29,44	14,40	3,54	12,38
12	27,90	23,30	12,43	6,19	6,75	8,06



Obr. 27 Srovnání DMOS v závislosti na HRC, databáze VQEG

Na základě grafu na obr. 27 můžeme konstatovat, že v některých případech se výsledky objektivních a subjektivních testů významně liší. Zde však musíme brát v potaz již výše uvedené nedostatky databáze VQEG, které měly vliv na námi použité metriky. Na základě subjektivních testů se jeví jako nejkvalitnější sekvence „Susie“. V případě, že by v sekvenci nebyl patrný řádkový proklad, by bylo možné dospět k podobnému výsledku i objektivními metrikami. Zajímavou situací je, že podle subjektivních testů by měly být sekvence s HRC 7, což byly sekvence s bitovým tokem 6 Mb/s, kvalitnější, než sekvence s HRC 2 (12-19 Mb/s). Zde je však možné, že hrála roli i prvotní kvalita ještě nekomprimovaných sekvencí.

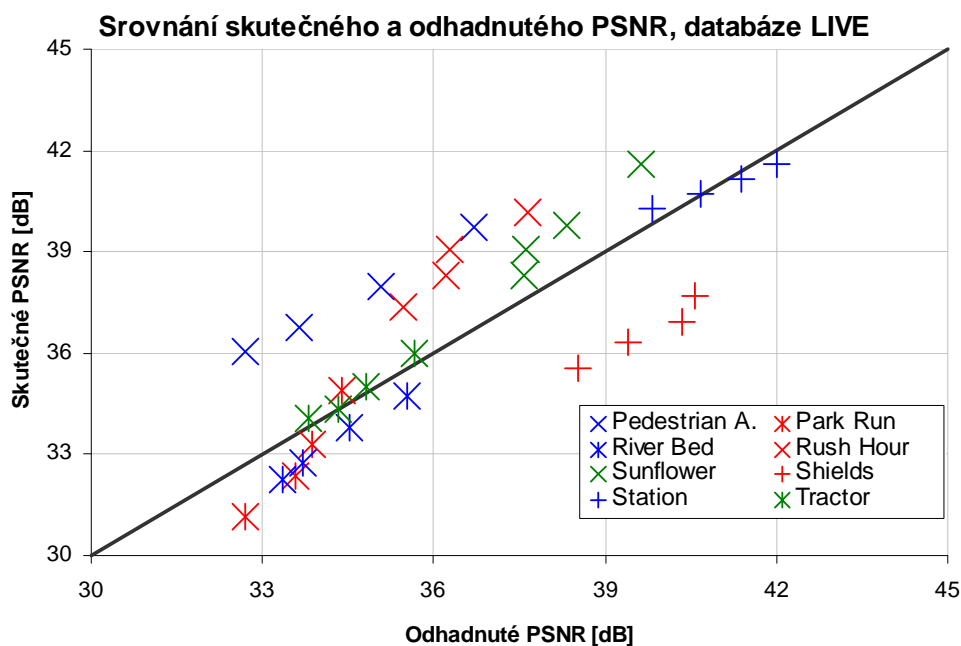
6.3 Odhad PSNR H.264 kódovaných sekvencí

Pro odhad PSNR sekvencí z databáze LIVE byla použita výše zmíněná námi navržená metrika. Pro zjištění skutečného PSNR byl napsán jednoduchý skript pro Matlab, který vykonává výpočet v souladu se vztahy uvedenými v kapitole 1.2.1. V následující tabulce jsou uvedeny hodnoty PSNR pro použité sekvence. Z celkových 32 sekvencí byly 3 sekvence použity pro učení algoritmu. Hodnoty PSNR pro tyto 3 sekvence jsou zvýrazněny v tabulce.

Tab. 12 Skutečná a odhadnutá PSNR, databáze LIVE

	Skutečné PSNR [dB]				Odhadnuté PSNR [dB]			
	9	10	11	12	9	10	11	12
Pedestrian A.	39,71	<u>37,95</u>	36,77	36,03	36,70	<u>35,06</u>	33,63	32,71
Park Run	34,88	33,31	32,34	<u>31,13</u>	34,41	33,86	33,57	<u>32,70</u>
River Bed	34,74	33,81	32,75	32,26	35,53	34,51	33,72	33,35
Rush Hour	40,15	39,04	38,31	37,36	37,65	36,27	36,23	35,45
Sunflower	41,59	39,75	39,09	38,32	39,63	38,34	37,61	37,57
Shields	37,70	36,95	36,30	35,54	40,57	40,35	39,42	38,52
Station	<u>41,57</u>	41,18	40,73	40,25	<u>42,01</u>	41,40	40,67	39,84
Tractor	36,01	34,98	34,35	34,05	35,67	34,81	34,33	33,82

Můžeme pozorovat, že odhadnuté hodnoty se ve většině případů příliš neliší od hodnot skutečných. Přehledněji můžeme situaci vidět na následujícím grafu..



Obr. 28 Srovnání skutečného a odhadnutého PSNR, databáze LIVE

V grafu jsou zobrazeny hodnoty PSNR pro všechny sekvence a dále přímka $x = y$, ke které by se hodnoty měly blížit. Vidíme, že i přes mnohá zjednodušení, kterých jsme se dopustili, tomu tak skutečně je. Výjimkou s velkou odchylkou je pouze sekvence „Pedestrian Area“, kde jsou hodnoty nadhodnocené. To může být např. tím, že se jedná o sekvenci s poměrně mnoha detaily a pohybem, na kterou kodér musel použít nižší hodnotu QP, přesto je však kvalita kvůli míře pohybu a detailům nedostatečná.

Následující tabulka uvádí relativní chybu odhadu PSNR. Vidíme, že chyba se pohybuje v maximálním rozmezí $\pm 9\%$. Nejvyšší odchylky byly u sekvencí „Pedestrian Area“ a „Shields“.

Tab. 13 Relativní chyba odhadu PSNR

	Relativní chyba odhadu [%]			
	9	10	11	12
Pedestrian A.	7,58	<u>7,61</u>	8,55	9,22
Park Run	1,36	-1,66	-3,79	<u>-5,03</u>
River Bed	-2,30	-2,09	-2,96	-3,39
Rush Hour	6,24	7,10	5,42	5,10
Sunflower	4,71	3,54	3,78	1,96
Shields	-7,63	-9,19	-8,59	-8,40
Station	<u>-1,04</u>	-0,53	0,16	1,03
Tractor	0,94	0,49	0,05	0,67

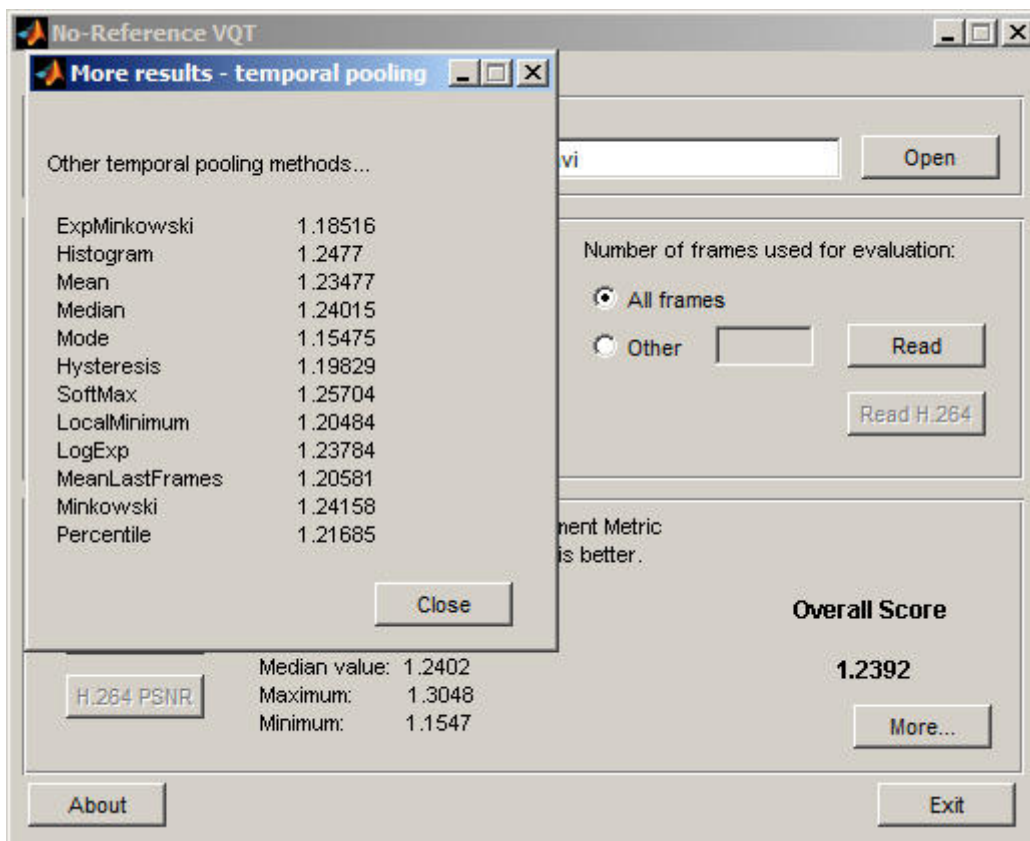
Relativní chybu odhadu jsme zde počítali podle vztahu

$$R = \frac{PSNR_{real} - PSNR_{est}}{PSNR_{real}} \cdot 100. \quad (13)$$

Zde $PSNR_{real}$ označuje skutečnou hodnotu PSNR a $PSNR_{est}$ pak hodnotu odhadnutou pomocí našeho algoritmu.

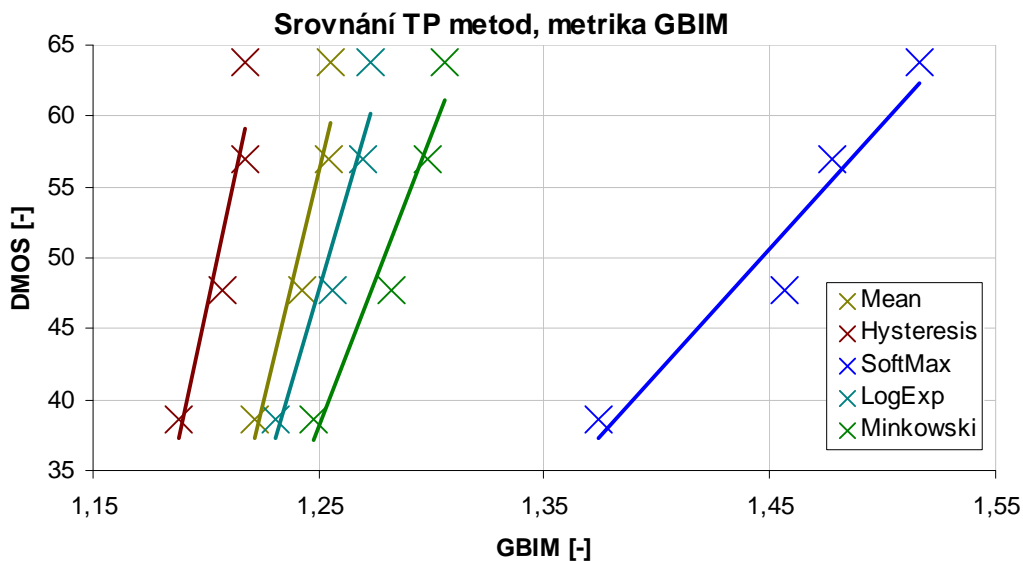
6.4 Srovnání přístupů temporal pooling

Výsledné skóre pro pixelové metriky počítáme pomocí Minkowskeho sumace, která je jednou z tzv. temporal pooling (TP) metod. Do našeho programu jsme implementovali pro uživatele možnost nechat si výsledné skóre spočítat i pomocí dalších TP přístupů. Použili jsme některé přístupy uvedené v literatuře [33] a dále přidali statistický medián a modus. Více o konkrétních přístupech lze nalézt tamtéž. Zobrazení výsledků v aplikaci můžeme vidět na obr. 29.



Obr. 29 Zobrazení dalších výsledků, temporal pooling

Dále jsme se pokusili určit, který přístup by mohl být nejlepší, tedy nejlépe vystihovat výsledný vliv sekvence na reálného pozorovatele. Postupovali jsme tak, že jsme pro jednu sekvenci a každou metriku vynesli závislost koeficientu metriky na hodnotě DMOS pro různé TP přístupy. Vzhledem k tomu, že mnohé přístupy dávaly velmi podobné výsledky, byly v grafu pro přehlednost ponechány jen některé z nich. Výsledky jsou dále ovlivněny tím, že v našem případě se hodnoty metrik pro jednotlivé snímky ve většině případů lišily jen velmi málo. Na následujícím grafu vidíme vliv použité TP metody na výsledném skóre pro metriku GBIM. Data odpovídají sekvenci „Tractor“.



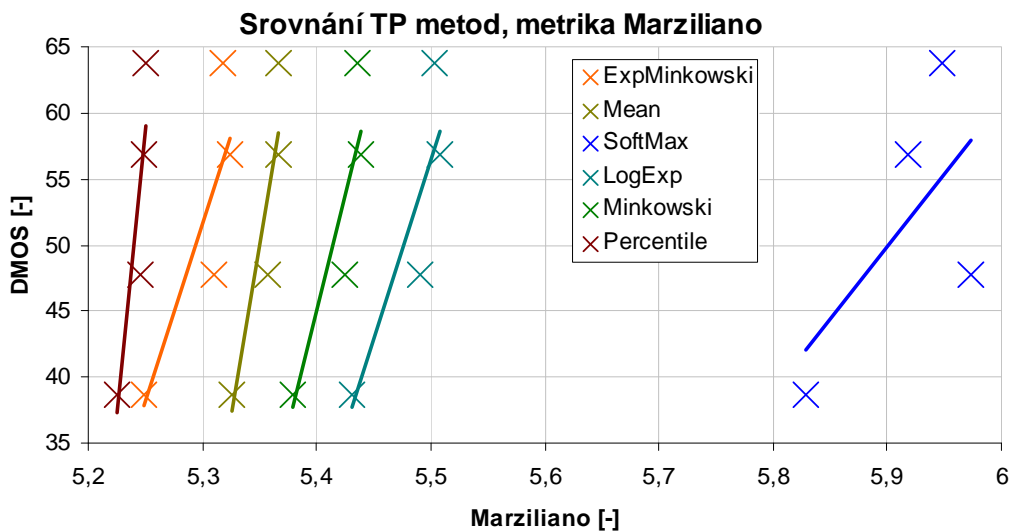
Obr. 30 Srovnání TP přístupů, metrika GBIM

V grafu vidíme 5 z celkových 12 metod. Metody LocalMinimum, Percentile, MeanLastFrames a Histogram byly vynechány z důvodu velké blízkosti výsledků s metodou Mean (tedy klasickým průměrem). Dále průběh metody Mode (modus) byl téměř totožný s průběhem přístupu Hysteresis. K datům jsme zobrazili lineární regresní přímku. Předpokladem je, že aby metoda co nejlépe odpovídala skutečnému vlivu na pozorovatele, měly by se hodnoty co nejvíce blížit právě regresní přímce. Vidíme, že ze zobrazených přístupů k tomu mají nejbližší klasický průměr a námi používaný Minkowski, případně ještě metod LogExp. Metoda SoftMax má příliš velkou strmost, a proto její užití není příliš vhodné.

Totéž srovnání jsme provedli i pro metriku Marziliano, použita byla tatáž sekvence. V tomto případě bylo relevantních více TP přístupů. Vynechána byla hystereze, jejíž trend byl opačný, než bychom očekávali, podobně i metoda modus. Graf můžeme vidět na obrázku 31. Z grafu vidíme, že nejvíce se lineární regresní přímce blíží opět výše jmenované přístupy Minkowski, průměr a také LogExp. Přístup Percentile také v případě metriky Marziliano vykazuje téměř lineární průběh. Přístup SoftMax má i zde příliš velkou strmost a nelze jej proto doporučit.

Je nutné podotknout, že v tomto případě se jedná pouze o velmi jednoduché srovnání přístupů temporal pooling. Nelze také jednoznačně určit, který přístup bude vždy nejlepší. Je potřeba vybírat v souladu s tím, jakou používáme metriku, podle toho i případně upravit nastavení jednotlivých TP metod, [22].

Všechny použité TP přístupy, kromě průměru, mediánu a modu, jsou parametrické funkce. Pro naše potřeby jsme použili hodnoty použité v [22], příp. [33]. Pro zohlednění recency efektu bylo ve většině případů použito přibližně 40% posledních snímků (vždy podle aktuálního počtu snímků). Případně byly parametry nastaveny v souladu s tím, že pro námi použité metriky roste jejich hodnota se snižující se kvalitou obrazu.



Obr. 31 Srovnání TP přístupů, metrika Marziliano

Výsledky, ke kterým jsem dospěli tímto jednoduchým srovnáním jsou i v souladu s výsledky uvedenými v [33]. Námi jmenované přístupy Minkowski, průměr a Percentile byly mezi prvními pěti nejlépe hodnocenými přístupy. Navíc na prvních dvou místech autoři umístili metody, které jsme my zde nepoužívali. Lze dále říci, že i jednoduchý průměr bude ve většině případů poskytovat relevantní výsledky, [33].

6.5 Shrnutí

V případě sekvencí databáze LIVE jsme dosáhli očekávaných výsledků. Hodnoty zjištěné pomocí pixelově-orientovaných přístupů skutečně byly v souladu s daty ze subjektivních měření. Kromě případu sekvence „River Bed“ a metriky Marziliano vždy nižší objektivní kvalita značila i nižší kvalitu subjektivní. Bylo by i možné vysledovat lineární závislost mezi oběma hodnoceními. Nicméně tento trend by byl aplikovatelný vždy jen na konkrétní sekvenci. Nejsme schopni přesně definovat, jaká hodnota námi použitých objektivních metrik bude vždy odpovídat skutečnému vlivu na reálného pozorovatele.

Také algoritmus pro odhad PSNR sekvencí kódovaných kodekem H.264/AVC splnil naše očekávání. Metoda byla testována pro 32 sekvencí. Relativní chyba odhadu PSNR za celou sekvenci se pohybovala v maximálním rozmezí $\pm 9\%$. Což je vzhledem ke zjednodušení, která jsme zavedli, poměrně dobrý výsledek. Horší je však odhad pro jednotlivé snímky. Zde jsme vyzorovali souvislost mezi chybou odhadu a pozicí konkrétního snímku ve skupině snímků. Čím blíže byl snímek typu P k následujícímu I snímku, tím více se chyba zvětšovala. Pro I snímky pak byl odhad zpravidla velmi přesný. Tento jev by bylo možné postihnout dalšími změnami algoritmu.

Testy databáze VQEG nesplnily naše očekávání. Zde se plně projevila ne příliš velká vhodnost této databáze pro náš účel. Sekvence obsahovaly řádkový proklad, s čímž si použité metriky nedokázaly bez problému poradit. Námi získaná data byla dále zkreslena tím, že sekvence často obsahovaly černé okraje, často i o velikosti

násobků velikosti bloku. To vedlo k nepřesnostem zejména u metriky GBIM.

Závěrem jsme se pokusili zhodnotit vliv použité temporal pooling metody na výsledné skóre za celou sekvenci. Provedli jsme srovnání několika přístupů. Výsledky prokázaly, že z hlediska výsledků a i náročnosti implementace je poměrně vhodný přístup založený na Minkowskeho sumaci. Také prostý průměr měl poměrně dobrý výkon.

ZÁVĚR

V práci byly popsány základní principy hodnocení kvality videa. Krátce byly uvedeny možnosti subjektivního hodnocení kvality, šířeji jsme se poté zabývali metodami objektivními. Ty kvalitu hodnotí pouze na základě matematických výpočtů a jsou tedy snáze aplikovatelné než testy subjektivní.

Hlavní zájem práce byly objektivní metriky bez použití referenčního snímku, označované jako no-reference přístupy. V druhé kapitole bylo nastíněno, jak takovéto metriky pracují. Bylo řečeno, že je třeba metriku navrhovat pro konkrétní typ rušení, který nás zajímá. Dále byly uvedeny rozdíly mezi metrikami pixelově-orientovanými a metrikami pracujícími na základě informací z bitového toku. Seznámili jsme se se třemi zástupci no-reference metrik a pokusili se zhodnotit náročnost jejich implementace.

V prostředí Matlab byla vytvořena aplikace pro bezreferenční hodnocení kvality videosekvencí. Byly implementovány dvě pixelově-orientované metriky, konkrétně metrika GBIM pro hodnocení blokových artefaktů a dále metrika Marziliano, která je schopna identifikovat neostrost obrazu.

Dále byla navržena a implementována metrika pro odhad PSNR pracující pouze na základě informací z bitového toku sekvencí kódovaných kodekem H.264. Metrika je velmi jednoduchá a rychlá, přesto poskytuje poměrně dobré výsledky.

Navržená aplikace byla použita k vedení objektivních testů kvality. K tomu byla použita databáze videosekvencí kódovaných jak ve standardu MPEG-2, kde jsme očekávali výraznou blokovou strukturu v obraze, tak i sekvence komprimované pomocí standardu H.264. Výsledky objektivních testů byly srovnány s výsledky subjektivních testů. V případě sekvencí kódovaných H.264 výsledky poměrně dobře odpovídají. U databáze sekvencí kódovaných MPEG-2 dochází k rozdílům, ty jsou však způsobeny ve většině případů přímo samotnými sekvencemi. Výsledky odhadu PSNR se lišily od skutečných hodnot maximálně o $\pm 9\%$, což můžeme pokládat vzhledem k množství zjednodušení za dobrý výsledek. Závěrem jsme se pokusili srovnat některé z temporal pooling přístupů. Jako efektivní se projevil námi již dříve používaný přístup Minkowskeho sumace.

LITERATURA

- [1] CORRIVEAU, Philip. Video Quality Testing. In: *Digital video image quality and perceptual coding*. Boca Raton: Taylor, 2006. 640 s. ISBN 0-8247-2777-0.
- [2] Rec. ITU-R BT.500-13. *Methodology for the subjective assessment of the quality of television pictures*. Geneva: ITU, 2012.
- [3] SLANINA, Martin. *Hodnocení kvality v analogové a digitální videotechnice*. Brno, 2011. Přednáška č. 12 předmětu MVDK. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky. 27 s.
- [4] WINKLER, Stefan. *Digital video quality*. London: Wiley, 2005. 175 s. ISBN 0-470-02404-6.
- [5] ZACH, Ondřej. *Měření kvality obrazu při příjmu analogového a digitálního televizního vysílání*. Brno, 2011. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky. 57 s.
- [6] WINKLER, Stefan. Perceptual Video Quality Metrics – A review. In: *Digital video image quality and perceptual coding*. Boca Raton: Taylor, 2006. 640 s ISBN 0-8247-2777-0.
- [7] SLANINA, Martin. *Methods and tools for image and video quality assesment*. Brno, 2008. Disertační práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky. 83 s.
- [8] FISCHER, Walter. *Digital video and audio broadcasting technology*. Berlin: Springer – Verlag, 2008. 580 s. ISBN 973-3-540-76357-4.
- [9] MARZILIANO, Pina et al. A no-reference perceptual blur metric. In: *Proceeding ICIP*, 3:57-60. Rochester , NY, Sep.22-25, 2002.
- [10] CAVIEDES, Jorge E. a Franco OBERTI. No-reference quality metric for degraded and enhanced video. In: *Digital video image quality and perceptual coding*. Boca Raton: Taylor, 2006. 640 s ISBN 0-8247-2777-0.
- [11] WU, H.R. a M. YUEN. A generalized block-edge impairment metric for video coding. In: *IEEE Signal processing Letters*, 4(11):317-320. Nov. 1977.
- [12] UZAIR, Muhammad a Dalia FAYEK. An efficient no-reference blockiness metric for intra-coded video frames. In *14th International Symposium on Wireless Personal Multimedia Communications*. Brest (France), 2011. p. 1-5.
- [13] LIU, H., WANG, J., REDI, J., LE CALLET, P., HEYNDERICKX, I. An efficient no-reference metric for perceived blur. In *3rd European Workshop on Visual Information Processing (EUVIP)*. Paris (France), 2011. p. 174 - 179.
- [14] JAN, Jiří. *Digital Signal Filtering, Analysis and Restoration*, IEE London (UK) 2000, ISBN 0-85296-760-8.
- [15] VQEG. *Validation of reduced-reference and no-reference objective models for standard definition television, phase I*. [online]. 2009. [2012-04-24]. Dostupný z: ftp://vqeg.its.bldrdoc.gov/Documents/Projects/rrnr-tv/RRNRTV_Final_Report_v1.8.pdf.
- [16] WANG, C., JIANG, X., MENG, F., WANG, Y. Quality Assessment for MPEG-2 Video Streams Using a Neural Network Model. In: *IEEE 13th International Conference on*

- Communication Technology (ICCT)*. Jinan, 2011 p. 868-872.
- [17] ICHIGAYA, A., KUROZUMI, M. et al A Method of Estimating Coding PSNR Using Quantized DCT Coefficients. In: *IEEE Transactions on Circuits and Systems for Video Technology*, vol 16, no. 2. Feb. 2006. ISSN: 1051-8215.
- [18] SCHWARZ, K. *FFmpeg W32 Static build version 2012-11-05 git-c995644* [online]. 2012. [cit. 2012-11-20]. Dostupný z : <http://ffmpeg.zeranoe.com/builds>.
- [19] Mathworks. *GUI design environment*. Součást balíku Matlab R2010a. 2010.
- [20] NACCARI, Matteo. *No-reference Generalized Block Impairment Metric (GBIM)* [online]. Instituto de Telecomunicações, Lisbon, Portugal. 2011. [cit. 2012-12-01]. Dostupný z: <https://sites.google.com/site/mateonaccari/software>.
- [21] MURTHY, V., KARAM, L. *VBQUEST- Visual Blur Quality Evaluation Software*. [online]. Image, Video and Usability Lab, Arizona State University. 2010. [cit. 2012-12-02]. Dostupný z: <http://ivulab.asu.edu/software/VBQUEST>.
- [22] RIMAC-DRLJE, S., VRANJES, M., ZAGAR, D. Influence of temporal pooling method on the objective video quality evaluation. In: *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, 2009. BMSB '09, 2009, pp. 1 –5.
- [23] Rec. ITU-R BT. 710-4. *Subjective assessment methods for image quality in high-definition television*. Geneva: ITU, 1998.
- [24] SLANINA, Martin. *Moderní komprimační standardy.MPEG-4 Part 2,MPEG-4 Part 10.* Brno, 2013. Přednáška č. 8 předmětu MVDK. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky. 41 s.
- [25] Rec. ITU-T H.264. *Advanced Video Coding for Generic Audiovisual Services*. Geneva: International Telecommunication Union, 2003.
- [26] VQEG. *Modified JM H.264/AVC codec: JM Reference Software v16.1 with XML tracefile generation* [online]. 2011. [cit. 2013-03-30]. Dostupný z: <http://vqegstl.ugent.be/?q=node/14>.
- [27] EDEN, Arnd. No-Reference Estimation of the Coding PSNR for H.264-Coded Sequences. In: *IEEE Transactions on Consumer Electronics*, vol. 53, no.2, pp. 667-674. 2007.
- [28] SHIM, S.-Y., MOON, J.-H., HAN, J.-K. PSNR estimation scheme using coefficient distribution of frequency domain in H.264 decoder. In: *Electronics Letters*, vol. 44, no.2, pp. 108-109. 2008.
- [29] FALKENA, Wouter. *xml2struct: Matlab function* [online]. TU Delft. 2010. [cit. 2013-04-10]. Dostupný z: <http://www.mathworks.com/matlabcentral/fileexchange/28518-xml2struct>.
- [30] SESHADRINATHAN, K., SOUNDARARAJAN, R. et al. Study of Subjective and Objective Quality Assessment of Video. In: *IEEE Transactions on Image Processing*, vol.19, no.6, pp.1427-1441. June 2010.
- [31] SESHADRINATHAN, K., SOUNDARARAJAN, R. et al. A Subjective Study to Evaluate Video Quality Assessment Algorithms. In: *SPIE Proceedings Human Vision and Electronic Imaging*. Jan. 2010.
- [32] VQEG. *FRTV Phase I: Video Quality Experts Group Test Sequences* [online]. 2000. [cit 2013-04-15]. Dostupný z: ftp://vqeg.its.bldrdoc.gov/SDTV/VQEG_PhaseI/.
- [33] SEUFERT, M., SLANINA, M., EGGERL, S., KOTTKAMP, M. "To pool or not to

pool”: A comparison of temporal pooling methods for http adaptive video streaming. In: *Quality of Multimedia Experience (QoMEX), 2013 Fifth International Workshop on*. 2013. [v tisku].

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AVC	Advanced Video Coding
CSF	Contrast Sensitivity Function
DCT	Discrete Cosine Transform
DSCQE	Double Stimulus Continuous Quality Scale
DSIS	Double Stimulus Impairment Scale
DVB	Digital Video Broadcasting
EBU	European Broadcasting Union
FR	Full-reference
GBIM	Generalized block-edge impairment Metric
HDTV	High Definition Television
HRC	Hypothetical Reference Circuit
HVS	Human Visual System
ITU	International Telecommunication Union
LIVE	Laboratory for Image and Visual Engineering
MPEG	Motion Picture Expert Group
MSE	Mean squared error
MSQE	Mean squared quantization error
NR	No-reference
NR VQT	No Reference Video Quality Tool
PSNR	Peak Signal to Noise Ratio
QP	Quantization Parameter
RR	Reduced-reference
SDTV	Standard Definition Television
SSCQE	Single Stimulus Continuous Quality Evaluation
SVD	Singular Value Decomposition
TP	Temporal Pooling
VQEG	Video Quality Experts Group

SEZNAM PŘÍLOH

A	CD	56
B	Seznam programových souborů	57
C	Uživatelská příručka	58
D	Původní a upravený xml soubor	60
E	Vybrané sekvence databáze VQEG	61
F	Vybrané sekvence databáze LIVE	62

A CD

- **Soubory aplikace** (nrvqt\...)
- **Programy ffmpeg a ffprobe** (nrvqt\ffmpeg\...)
- **Licence knihoven ffmpeg** (nrvqt\ffmpeg\Licenses\...)
- **Referenční video „xylophone.mpg“** (nrvqt\...)
- **Sekvence databáze LIVE** (LIVE\...)
- **Zdrojové kódy ldecod** (ldecod\...)

B SEZNAM PROGRAMOVÝCH SOUBORŮ

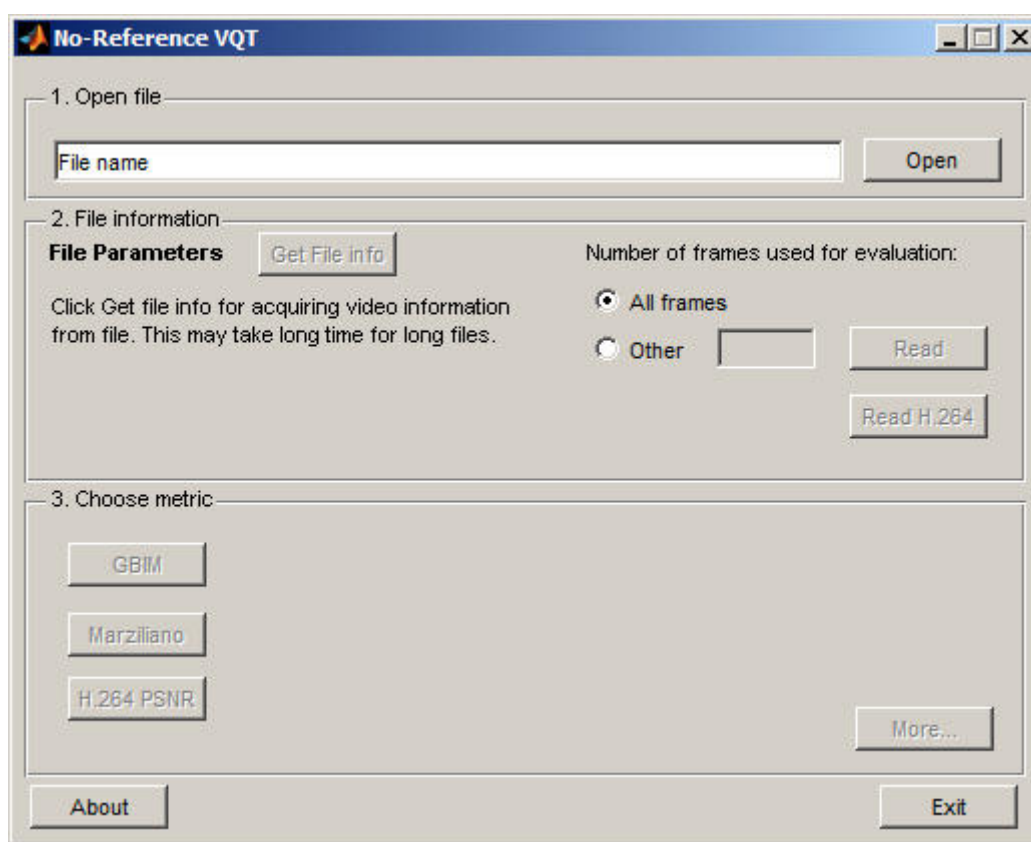
Soubor	Popis
about.fig	Grafické rozhraní zobrazení informací o programu
about.m	Funkce grafického rozhraní informací o programu
GBIM.m	Funkce metriky GBIM
getLuma.m	Funkce pro získání jasového signálu
ldecod.exe	Modifikovaný H.264 dekodér s generováním XML souboru
marziliano.m	Funkce metriky Marziliano
minkowski.m	Funkce pro výpočet Minkowskeho sumace
more_temp.fig	Grafické rozhraní zobrazení dalších TP metod
more_temp.m	Funkce zobrazení dalších TP metod
nrvqt.fig	Grafické rozhraní hlavní aplikace
nrvqt.m	Funkce hlavní aplikace
resfind.m	Funkce pro vyhledání informací o souboru
temppool.m	Funkce pro výpočet skóre pomocí dalších TP metod
prirucka.pdf	Uživatelská příručka programu NR VQT
xml2psnr.m	Funkce pro odhad PSNR H.264 kódovaných sekvencí
xml2struct.m	Funkce pro import dat z XML souboru do Matlabu

C UŽIVATELSKÁ PŘÍRUČKA

No-Reference Video Quality Tool (NR VQT) je aplikace pro prostředí Matlab, která slouží k hodnocení kvality videa bezreferenčními metrikami.

Upozornění: Aplikace musí být spouštěna z úložiště s možností zápisu. Aplikace si vytváří v rámci své složky dočasné soubory, se kterými dále pracuje. Po ukončení aplikace jsou dočasná data smazána.

Na obrázku 1 můžeme vidět uživatelské rozhraní při spuštění aplikace.



Obrázek 1 Uživatelské rozhraní NR VQT

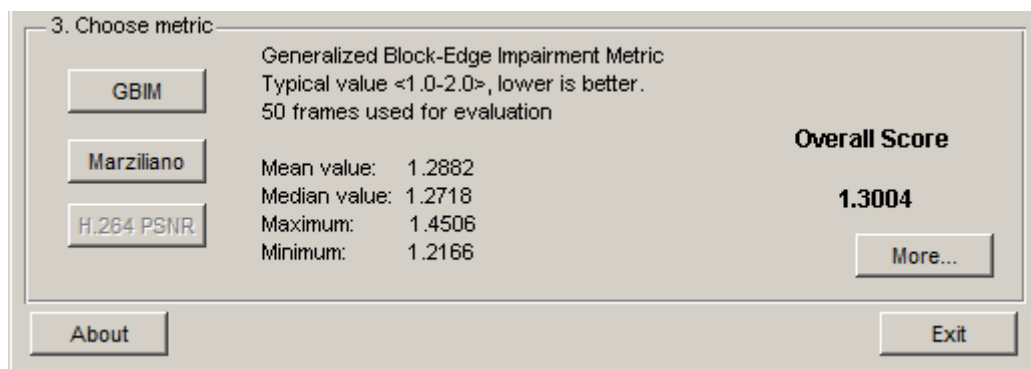
Pro načtení souboru slouží tlačítko OPEN. Uživatel je vyzván k výběru souboru, jehož kvalitu chce zjistit. Implicitně je možný výběr souborů s příponou *.AVI a *.MPG a dále videa kódovaná pomocí kodeku H.264/AVC s příponami *.MP4, *.264 a *.h264. Po výběru souboru se zpřístupní tlačítko GET FILE INFO. Toto tlačítko slouží ke zjištění informací o souboru. Je-li soubor regulérním video souborem, do prostoru pod tlačítkem se vypíše parametry videosekvence, jako počet snímků, jeho šířka a výška. V případě, že soubor není platným video souborem, je vypsána hláška. Dále je signalizována případná detekce kodeku H.264/AVC.

Je-li vybraný soubor platným video souborem, uživateli se zpřístupní tlačítko READ, případně READ H.264. V této chvíli také může zvolit počet snímků, které chce

načíst a následně použít pro hodnocení kvality. Platný rozsah počtu snímků je 1 až počet snímků uvedený v části „File Parameters“. Zadá-li uživatel neplatnou hodnotu, aplikace po stisku tlačítka ohlásí chybu. V opačném případě proběhne dekodování souboru, aplikace uloží dočasný soubor s názvem „temp.yuv“ do programové složky a následně z něj načte data do prostředí Matlab. V případě použití aplikace pro odhad PSNR dojde k převodu datového toku na formát podle Annex B kodeku H.264 a dojde k extrakci parametrů z bitového toku do souboru „trace.xml“. O úspěšném ukončení procesu je uživatel informován hláškou. Velikost dočasného souboru závisí na rozlišení videa a počtu snímků.

Upozornění: Výběr nevhodného souboru může způsobit, že výpočet bude trvat neúměrně dlouho. Doporučujeme proto vybírat kratší sekvence, případně ručně zvolit menší počet snímků.

Nyní se uživateli zpřístupní tlačítka jednotlivých metrik GBIM a MARZILIANO, případně i H.264 PSNR. Po stisknutí tlačítka začne výpočet metriky, uživatel je o průběhu informován pomocí ukazatele. Po skončení výpočtu se v prostoru vedle tlačítek zobrazí výsledky, příklad viz obrázek 2.



Obrázek 2 Příklad zobrazení výsledků

Zde vidíme zobrazení výsledků výpočtu metriky GBIM. První řádek odpovídá názvu použité metriky, druhý pak informuje o typických hodnotách výsledku a třetí řádek je počet skutečně použitých snímků pro výpočet. Na následujících řádcích jsou poté vypsány průměr, medián, maximální a minimální hodnota ze souboru hodnot pro všechny snímky. Na základě všech hodnot je pak vypočítáno celkové skóre (Overall Score). Uživatel má možnost výpočtu celkového skóre pomocí dalších metod, k tomu slouží tlačítko MORE.... Pokud uživatel otevře nový soubor, jsou tyto výsledky vymazány.

Ukončení aplikace je možné stiskem tlačítka EXIT, případně pomocí křížku v pravém horním rohu. Při ukončení aplikace smaže dočasný soubor „temp.yuv“, případně další data, která aplikace během své činnosti vytvořila.

Požadavky aplikace:

- OS Windows (XP nebo novější), Matlab (testováno pro v2009, 2010 a 2011)
- 42 MB volného místa na disku pro aplikaci s podpůrnými soubory + další prostor pro dočasný soubor.
- Doporučený procesor 1,5 GHz a více.

D PŮVODNÍ A UPRAVENÝ XML SOUBOR

a) Původní dekodér bez úprav, část XML souboru:


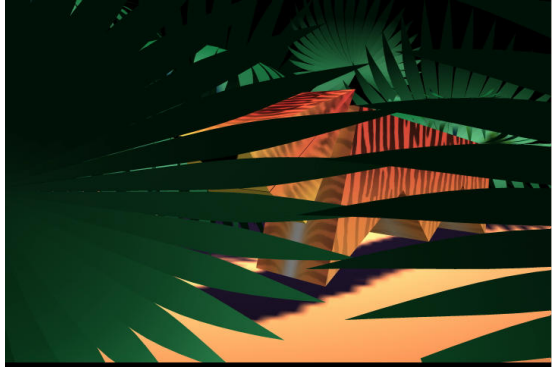

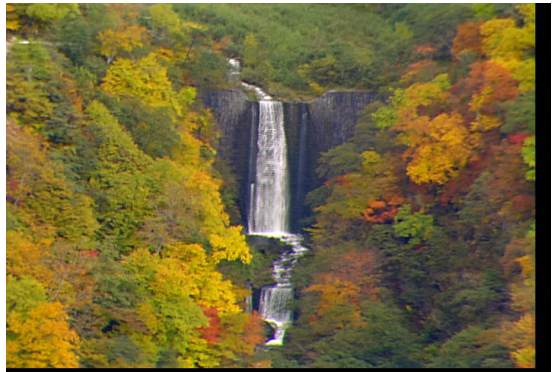


```
...
<MacroBlock num="0">
  <Position>
    <X>0</X>
    <Y>0</Y>
  </Position>
  <QP_Y>35</QP_Y>
  <Type>0</Type>
  <TypeString>I_4x4</TypeString>
  <PredModeString>BLOCK_TYPE_I</PredModeString>
  <SkipFlag>0</SkipFlag>
  <Coeffs>
    <Plane type="0">
      <Row>-5184,-736,0,0,576,1472,0,0,0,0,0,0,0,0,0,0,0,0,0,0</Row>
      <Row>0,0,0,0,-1472,-928,0,0,0,0,0,0,0,0,0,0,0,0,0,0</Row>
      <Row>0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0</Row>
      <Row>0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0</Row>
    </Plane>
  </Coeffs>
</MacroBlock>
...
```

b) Finální verze úprav dekodéru, část XML souboru:

```
...
<Picture Type="I">35,235;35,245;35,242;35,251;35,249;35,233; ...
...
```

Zatímco v případě a) je zobrazena pouze část jasových hodnot pro jeden makroblok (další koeficienty včetně chrominančních nejsou již z prostorových důvodů zobrazeny), v případě b) vidíme hodnoty QP a počtu nulových jasových transformačních koeficientů pro 6 makrobloků. Z toho lze pozorovat, že optimalizace je značná a přesto mají data pro naše potřeby stále stejnou informační hodnotu. Pro úplnost jen dodáváme, že se jedná o data pro stejný vstupní videosoubor jako v kapitole 4.3.

E VYBRANÉ SEKVENCE DATABÁZE VQEG

	
Sekvence 2: Barcelona	Sekvence 16: Betes pas betes
	
Sekvence 7: Harp	Sekvence 18: Autumn leaves
	
Sekvence 7: Fries	Sekvence 21: Susie

F VYBRANÉ SEKVENCE DATABÁZE LIVE

A busy urban pedestrian area with people walking and a cyclist in the foreground.	A park run event with a stone wall in the foreground and trees in the background.
A close-up view of a river bed with water flowing over rocks.	A busy street during rush hour with many cars in traffic.
A close-up of a sunflower head with a small insect on it.	A man pointing at a display of various heraldic shields.
A railway station with multiple tracks and overhead power lines.	A green tractor with a red wheel and a blue implement in a field.