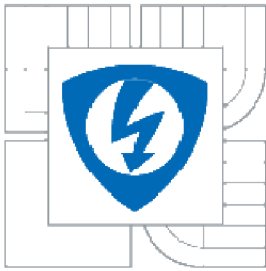




**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ  
ÚSTAV RADIOELEKTRONIKY**

**FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF RADIO ELECTRONICS**

# **IMPLEMENTACE METRIKY PRO HODNOCENÍ KVALITY VIDEOSEKVENČÍ DO DEKODÉRU H.264/AVC**

**IMPLEMENTING A VIDEO QUALITY METRIC IN THE H.264/AVC DECODER**

**DIPLOMOVÁ PRÁCE**  
MASTER'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**Bc. Michal Grúbel**

**VEDOUCÍ PRÁCE**  
SUPERVISOR

**Ing. Martin Slanina, Ph.D.**

**BRNO 2010**

# LICENČNÍ SMLOUVA

## POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

### 1. Pan/paní

Jméno a příjmení: Bc. Michal Grúbel  
Bytem: Moyzesova 17, Poprad, 058 01  
Narozen/a (datum a místo): 5. februára 1985 v Poprade

(dále jen „autor“)

a

### 2. Vysoké učení technické v Brně

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

## Čl. 1

### Specifikace školního díla

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

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

Název VŠKP: Implementace metriky pro hodnocení kvality videosekvencí do dekodéru H.264/AVC

Vedoucí/ školitel VŠKP: Ing. Martin Slanina Ph.D.

Ústav: Ústav radioelektroniky

Datum obhajoby VŠKP: \_\_\_\_\_

VŠKP odevzdal autor nabyvateli\*:

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

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

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

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

---

\* hodíci se zaškrtněte

## Článek 2

### Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
  - ihned po uzavření této smlouvy
  - 1 rok po uzavření této smlouvy
  - 3 roky po uzavření této smlouvy
  - 5 let po uzavření této smlouvy
  - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

## Článek 3

### Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 21. května 2010

.....  
Nabyvatel

.....  
Autor

## Abstrakt

V tejto diplomovej práci je predstavený a aplikovaný algoritmus pre hodnotenie kvality videosekvencií zakódovaných pomocou štandardu H.264. Ako merítka kvality obrazu je použitá objektívna metrika špičkový pomer signálu k šumu (PSNR). Zatiaľ čo výpočet PSNR zvyčajne vyžaduje referenčný signál, a ten porovnáva so skresleným, tento algoritmus dokáže vyčíslieť PSNR na základe kódovaných transformačných koeficientov. A teda nie je potrebný žiadny referenčný signál.

**Kľúčové slová:** H.264, skutočné PSNR, odhadnuté PSNR, kvalita obrazu, relatívna odchýlka, enkodér, dekodér, videosekvencia

## Abstract

In this diploma thesis an algorithm for the evaluation of picture quality of H.264-coded video sequences is introduced and applied. As a measure of picture quality objective metric the peak signal to noise ratio (PSNR) is used. While the computation of the PSNR usually requires a reference signal and compares it to the distorted video sequence, this algorithm is able to evaluate PSNR following the coded transform coefficients. Thus, no reference signal is needed.

**Keywords:** H.264, real PSNR, estimated PSNR, picture quality, relative error, encoder, decoder, video sequence



GRÚBEL, M. *Implementace metriky pro hodnocení kvality videosekvencí do dekodéru H.264/AVC*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 42 s. Vedoucí diplomové práce: Ing. Martin Slanina, Ph.D.

## PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma Implementace metriky pro hodnocení kvality videosekvencí do dekodéru H.264/AVC 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 .....

.....

(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 .....

.....

(podpis autora)

# Obsah

1	Úvod.....	- 5 -
2	H.264/MPEG-4 AVC.....	- 6 -
2.1	Úvod.....	- 6 -
2.1.1	Terminológia.....	- 6 -
2.2	Kodek H.264.....	- 6 -
2.3	Štruktúra H.264.....	- 8 -
2.3.1	Profily a Levely.....	- 8 -
2.3.2	Referenčné snímky.....	- 9 -
2.3.3	Prúžky.....	- 9 -
2.3.4	Makrobloky.....	- 10 -
2.3.5	Inter predikcia.....	- 10 -
2.3.6	Intra predikcia.....	- 10 -
2.3.7	Deblokovací filter.....	- 11 -
3	Princíp funkcie algoritmu.....	- 12 -
3.1	Vývoj za posledné roky.....	- 12 -
3.2	PSNR ako indikátor kvality.....	- 13 -
3.3	H.264 transformácia a kvantovanie.....	- 13 -
3.4	Výpočet kódovacej chyby v transformačnej oblasti.....	- 14 -
3.5	Odhad rozdelenia transformačných koeficientov.....	- 16 -
3.6	Vyhodnotenie algoritmu.....	- 17 -
3.7	Výpočet MSE.....	- 18 -
4	H.264/AVC referenčný softvér.....	- 19 -
4.1	Štruktúra.....	- 19 -
4.2	Referenčný enkodér.....	- 19 -
4.3	Referenčný dekodér.....	- 20 -
4.4	M-file „evaluate“.....	- 21 -
4.5	Overenie správnosti výpočtu.....	- 21 -
5	Optimalizácia.....	- 23 -
5.1	Analýza zdrojových kódov referenčného dekodéru.....	- 23 -
5.2	Sumár použitých funkcií.....	- 24 -
5.3	Záverečná úprava kódu.....	- 25 -
6	Použitie referenčného dekodéru.....	- 27 -
6.1	Konfiguračný súbor.....	- 27 -
6.2	Kalibrácia parametra $\alpha$ .....	- 27 -
6.3	Obsluha referenčného dekodéru.....	- 27 -
7	Ďalšie testy algoritmu.....	- 29 -
7.1	Charakteristika použitých videosekvencií.....	- 29 -
7.2	Vplyv kvality videosekvencie na presnosť odhadu PSNR.....	- 31 -
7.3	Vplyv obsahu videosekvencie na presnosť odhadu PSNR.....	- 33 -
7.4	Vplyv rozlíšenia videosekvencie na presnosť odhadu PSNR.....	- 37 -
	Záver.....	- 41 -
	Použitá literatúra.....	- 42 -

# 1 Úvod

V tejto diplomovej práci je predstavený štandard H.264/MPEG-4 AVC. Sú vysvetlené zákonitosti pre správnu funkciu enkodéru a dekodéru. Podrobne je objasnená štruktúra štandardu, od profilov a úrovní, cez typy snímkov, prúžkov, makroblokov až po typy predikcie a deblokovací filter.

Ďalej je predstavený algoritmus pre hodnotenie kvality obrazu videosekvencií kódovaných pomocou enkodéru H.264/AVC. Tento algoritmus nepotrebuje žiadny referenčný signál. Ako merítka kvality je použitý špičkový pomer signálu k šumu. Je tu vysvetlené, prečo je použitý práve PSNR ako indikátor kvality. Ďalej sa tu hovorí o transformácii a kvantovaní u štandardu H.264. Sú vysvetlené inovácie v štandarde H.264, a síce použitie aproximácie diskretnej kosínovej transformácie (DCT) namiesto klasickej DCT. Je tu ešte vysvetlený výpočet kódovacej chyby v transformačnej oblasti a tiež odhad rozdelenia transformačných koeficientov. Ako dôkaz kvality predstaveného algoritmu, sú tu uvedené výsledky testov porovnania skutočnej hodnoty PSNR a odhadnutej hodnoty PSNR pomocou daného algoritmu pre testovacie videosekvencie 720p50 a 1080i25 pre I, P aj B snímky.

Je tu predstavený H.264/AVC referenčný softvér, z ktorého boli získané dôležité hodnoty pre samotnú aplikáciu spomínaného algoritmu. Sú tu interpretované dosiahnuté výsledky odhadu hodnôt PSNR a ich porovnanie so skutočnými hodnotami. Bola tu vypočítaná korelácia skutočných a odhadnutých hodnôt PSNR.

Ďalej tu je vysvetlený postup pri optimalizácii referenčného dekodéru, aby jeho jedinou funkcionalitou bol odhad hodnôt PSNR, potom jeho použitie a správne nastavenie parametrov a nakoniec sú uvedené rozšírené testy algoritmu a ich výsledky, ktoré odhaľujú závislosti presnosti odhadu PSNR na rôznych vlastnostiach videosekvencie, ako napríklad: rozlíšenie, obsah, kvalita.

## 2 H.264/MPEG-4 AVC

### 2.1 Úvod

Moving Picture Experts Group a Video Coding Experts Group (MPEG and VCEG) vyvinuli štandard, ktorý sľubuje predčiť skoršie MPEG-4 a H.263 štandardy dosiahnutím lepšej kompresie video snímkov. Tento štandard sa nazýva Advanced Video Coding (AVC) a je publikovaný spoločne ako MPEG-4 Part 10 a H.264 [3].

#### 2.1.1 Terminológia

Kódovaný snímok pozostáva z množstva makroblokov, z ktorých každý obsahuje 16 x 16 lumanančných vzoriek a podvzorkované chrominančné vzorky (8 x 8 Cb a 8 x 8 Cr). V každom snímku sú makrobloky štruktúrované do prúžkov, kde prúžok je súbor makroblokov v určitom poradí. I prúžok môže obsahovať iba makrobloky typu I. P prúžok môže obsahovať P a I makrobloky a nakoniec B prúžok môže obsahovať makrobloky typu B a I (existujú aj ďalšie prúžky typu SI a SP) [3].

I makrobloky sú predikované použitím intra predikcie z dekódovaných vzoriek v aktuálnom prúžku. Predikcia je uskutočnená buď pre celý makroblok alebo pre každý 4 x 4 blok lumanančných vzoriek (a k nim prislúchajúce chrominančné vzorky) v makrobloku [3].

P makrobloky sú predikované použitím inter predikcie z referenčných snímkov. Inter kódované makrobloky môžu byť rozdelené do makroblokových partií (bloky o veľkosti 16 x 16, 16 x 8, 8 x 16 alebo 8 x 8 lumanančných vzoriek a im zodpovedajúce chrominančné vzorky). Ak je zvolená particia 8 x 8, každý 8 x 8 submakroblok môže byť ďalej rozdelený do submakroblokových partií o veľkosti 8 x 8, 8 x 4, 4 x 8 alebo 4 x 4 lumanančných vzoriek a im zodpovedajúce chrominančné vzorky [3].

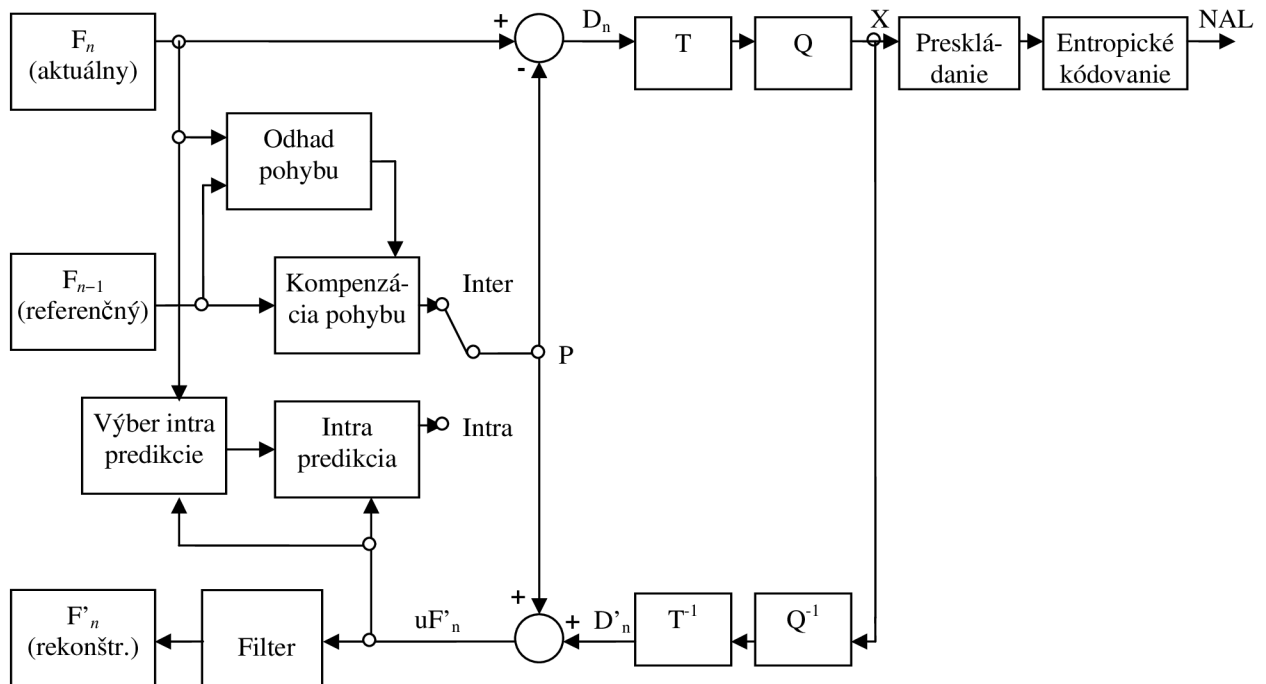
B makrobloky sú predikované použitím inter predikcie z referenčných snímkov. Každá makrobloková particia môže byť predikovaná z jedného alebo dvoch referenčných snímkov [3].

### 2.2 Kodek H.264

Štandard H.264 explicitne nedefinuje KODEK (enKoder / DEKoder), ale definuje syntax enkódovaného bitového toku spolu s metódou dekódovania tohto bitového toku. V praxi, vyhovujúci enkodér aj dekodér by mali zahrnúť funkčné elementy ukázané na obrázkoch obr.1 a obr.2. S výnimkou deblokovacieho filtra, väčšina základných funkčných elementov (predikcia, transformácia, kvantovanie, entropické kódovanie) sú prítomné aj v predchádzajúcich štandardoch (MPEG-1, MPEG-2, MPEG-4, H.261, H.263), ale dôležité zmeny sú v detailoch každého funkčného bloku [3].

Enkodér (obr.1) zahŕňa dva datové toky, „priamy“ (z ľava do prava) a „rekonštrukčný“ (z prava do ľava). Datový tok v dekodéry (obr.2) ukazuje podobnosti medzi enkodérom a dekodérom. Ešte pred samotným skúmaním detailov štandardu H.264 budú popísané hlavné kroky pri enkódovaní a dekódovaní snímku videa. Následujúci popis je zjednodušený za účelom poskytnutia prehľadu v enkódovaní a dekódovaní. Termín „blok“ je použitý na

označenie makroblokovej particie alebo submakroblokovej particie alebo 16 x 16 alebo 4 x 4 blok luminančných vzoriek a im zodpovedajúcich chrominančných vzoriek [3].



Obr. 1 H.264 Enkodér.

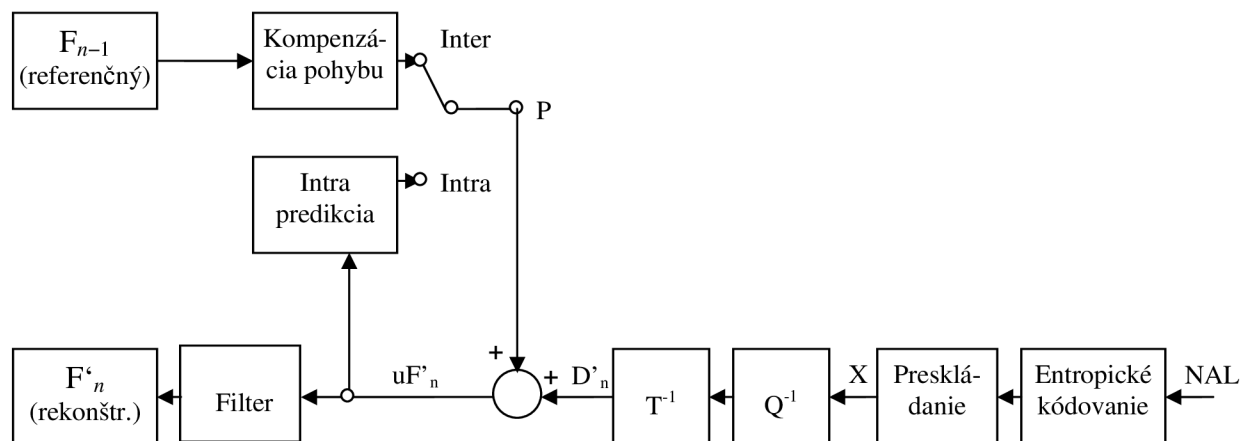
### Enkodér („priamy“ tok)

Vstupný snímok je spracovaný v jednotkách makrobloku. Každý makroblok je kódovaný v intra alebo inter móde a pre každý blok v makrobloku predikcia, na obrázku označená „P“ je vytvorená na základe zrekonštruovaných vzoriek snímku. V intra móde je predikcia vytvorená zo vzoriek v aktuálnom prúžku, ktoré boli predtým enkódované, dekódované a zrekonštruované. V inter móde je predikcia vytvorená predikciou kompenzácie pohybu z jedného alebo dvoch referenčných snímok. Na obrázkoch je referenčný snímok ukázaný ako predchádzajúci kódovaný snímok  $F'_{n-1}$ , ale referencia predikcie pre každú makroblokovú particiu (v inter móde) môže byť vybraná z ponuky minulých alebo budúcich snímok (v poradí ako sú zobrazené), ktoré už boli kódované, zrekonštruované a filtrované [3].

Predikcia je odpočítaná od aktuálneho bloku, aby vznikol diferenciálny blok  $D_n$ , ktorý je transformovaný (T) a kvantovaný (Q) a dáva X, súbor kvantovaných transformačných koeficientov, u ktorých je následne prehodné poradie a sú entropicky kódované. Entropicky kódované koeficienty, spolu s požadovanými bočnými informáciami potrebnými na dekódovanie každého bloku v makrobloku (mód predikcie, kvantizačný parameter, pohybový vektor atd.) vytvárajú komprimovaný bitový tok, ktorý je posunutý do sieťovej abstrakčnej vrstvy - NAL (Network Abstraction Layer) pre prenos alebo uloženie [3].

### Enkodér („rekonštrukčný“ tok)

Enkodér rovnako ako enkóduje a prenáša každý blok v makrobloku, tak ho aj dekóduje (rekonštruuje), aby poskytol referenciu pre ďalšiu predikciu. Koeficienty X sú inverzne kvantované ( $Q^{-1}$ ) a inverzne transformované ( $T^{-1}$ ), aby vytvorili diferenciálny blok  $D'_n$ . K tomuto diferenciálnemu bloku je pridaný predikčný blok a vzniká zrekonštruovaný blok  $uF'_n$  (dekódovaná verzia originálneho bloku;  $u$  indikuje, že je nefiltrovaný). Filter je aplikovaný, aby zredukoval blokové skreslenie a zrekonštruovaný referenčný snímok je vytvorený zo súboru blokov  $F_n$  [3].



Obr. 2 H.264 Dekodér.

## Dekodér

Dekodér prijíma komprimovaný bitový tok z NAL a entropicky dekoduje datové elementy tak, aby vznikol súbor kvantizačných koeficientov  $X$ . Tie sú ďalej škálované a inverzne transformované, aby dali  $D_n$  (identický ako  $D_n$  u enkodéru). Použitím hlavičkových informácií dekodovaných z bitového toku, dekodér vytvára predikčný blok, identický originálnemu predikčnému bloku vytvorenému v enkodéri. Tento blok je pridaný ku  $D_n$  a vzniká  $uF_n$ , ktorý je filtrovaný až napokon vznikajú samotné dekodované bloky  $F_n$  [3].

## 2.3 Štruktúra H.264

### 2.3.1 Profily a Levely

H.264 definuje 3 profily, každý z nich podporuje jednotlivé kódovacie funkcie a špecifikuje, čo je požadované od enkodéru aj dekodéru, aby vyhovovali danému profilu [3].

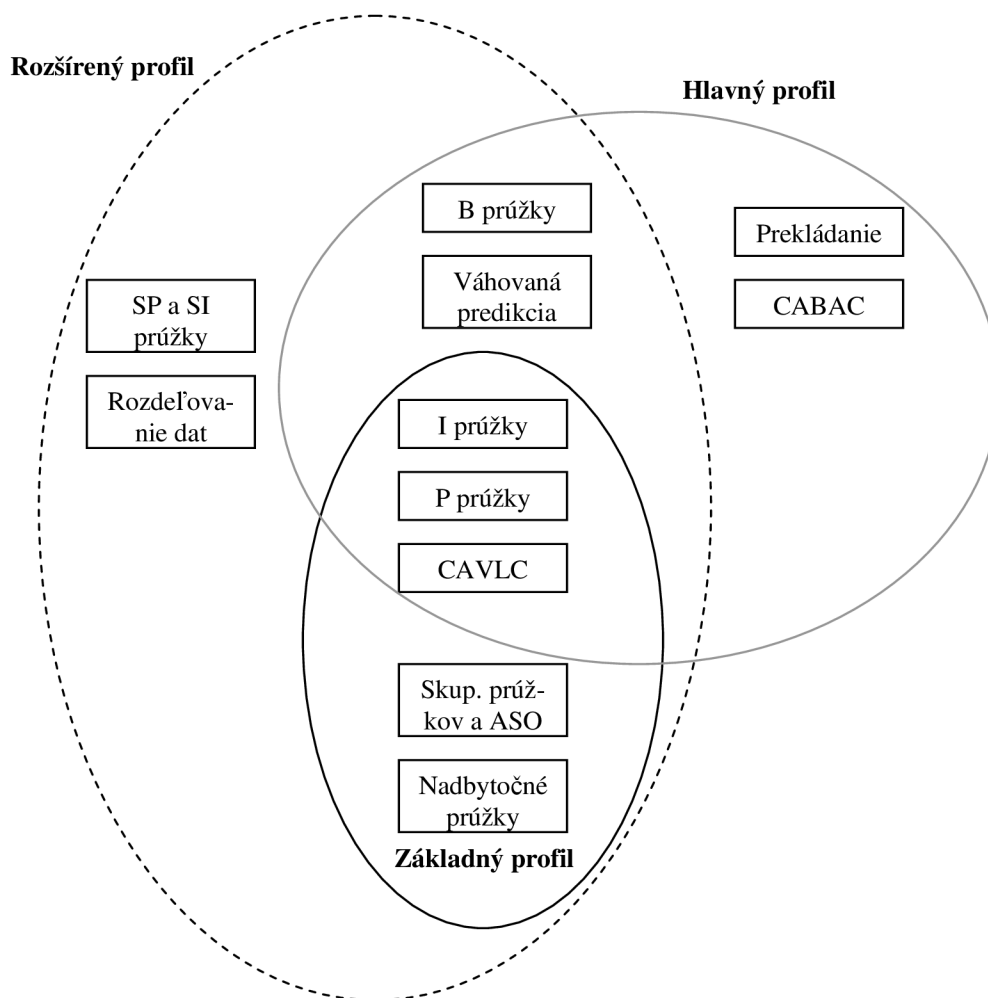
*Základný profil* podporuje intra a inter kódovanie (používajúce I-prúžky a P-prúžky) a entropické kódovanie s CAVLC (context-adaptive variable-length codes). Potenciálne aplikácie základného profilu zahŕňujú videotelefón, videokonferenciu a bezdrátové komunikácie [3].

*Hlavný profil* podporuje prekládané video, inter kódovanie používajúce B-prúžky, inter kódovanie používajúce váhovanú predikciu a entropické kódovanie používajúce CABAC (context-based arithmetic coding). Potenciálne aplikácie hlavného profilu zahŕňujú televízne vysielanie a ukládanie videa [3].

*Rozšírený profil* nepodporuje prekládané video alebo CABAC, ale pridáva módy na povolenie účinného prepínania medzi kódovanými bitovými tokmi (SP- a SI-prúžky) a vylepšenú odolnosť voči chybám. Rozšírený profil môže byť užitočný na aplikácie pre strímovanie videa [3].

Avšak, každý profil má dostatočnú flexibilitu na podporu veľkého rozsahu aplikácií, a preto by tieto príklady nemali byť pokladané za konečné [3].

Obr.3 ukazuje vzťah medzi týmito troma profilmi a kódovacími nástrojmi podporovanými štandardom. Je jasné, že základný profil je podmnožinou rozšíreného profilu, ale nie hlavného. Výkonové limity pre kodeky sú definované súborom úrovní [3].



Obr. 3 H.264 profily.

### 2.3.2 Referenčné snímky

H.264 enkodér môže použiť jeden alebo dva z množstva predchádzajúcich kódovaných snímok ako referenciu pre kompenzáciu pohybu každého inter kódovaného makrobloku alebo jeho particie. Toto povoľuje enkodéru hľadať najlepšiu zhodu pre aktuálnu makroblokovú particiu z širšieho výberu snímok ako len z predchádzajúceho snímku [3].

Enkodér aj dekodér si ukladajú jeden alebo dva zoznamy referenčných snímok, obsahujúce také snímky, ktoré už boli enkódované a dekódované. Inter kódované makrobloky a makroblokové particie v P prúžkoch sú predikované z obrázkov v jednom zozname, list 0. Inter kódované makrobloky a makroblokové particie v B prúžkoch môžu byť predikované z dvoch zoznamov, list 0 a list 1 [3].

### 2.3.3 Prúžky

Snímok je kódovaný ako jeden alebo viac prúžkov, z ktorých každý obsahuje celočíselný počet makroblokov od 1 po celkový počet makroblokov v snímku (1 prúžok v celom snímku). Počet makroblokov v prúžku musí byť konštantný vo vnútri celého snímku. Existuje 5 typov prúžkov (tab.1), z ktorých môže byť snímok poskladaný (rôzne). Napríklad,



u základného profilu môže snímok obsahovať mix I a P prúžkov a u hlavného a rozšíreného profilu môže snímok obsahovať mix I, P aj B prúžkov [3].

**Tab.1:** Typy prúžkov a ich popis

Typ prúžku	Popis	Profil(y)
<b>I (Intra)</b>	Obsahuje iba I makrobloky (každý blok alebo makroblok je predikovaný z predchádzajúcich kódovaných dat vo vnútri toho istého prúžku)	Všetky
<b>P (Predicted)</b>	Obsahuje P makrobloky (každý makroblok alebo makrobloková particia je predikovaná z jedného listu 0 referenčných obrázkov) a/alebo I makrobloky	Všetky
<b>B (Bi-predictive)</b>	Obsahuje B makrobloky (každý makroblok alebo makrobloková particia je predikovaná z listu 0 a/alebo listu 1 referenčných obrázkov) a/alebo I makrobloky	Rozšírený a hlavný
<b>SP (Switching P)</b>	Uľahčuje prepínanie medzi kódovanými tokmi; obsahuje P a/alebo I makrobloky	Rozšírený
<b>SI (Switching I)</b>	Uľahčuje prepínanie medzi kódovanými tokmi; obsahuje SI makrobloky (špeciálny typ intra kódovaných makroblokov)	Rozšírený

### 2.3.4 Makrobloky

Makroblok obsahuje kódované data o veľkosti 16 x 16 vzoriek (16 x 16 luminančných, 8 x 8 Cb a 8 x 8 Cr vzoriek). Makrobloky sú číslované (adresované) v rástrovom poradí vo vnútri snímku [3].

### 2.3.5 Inter predikcia

Inter predikcia vytvára predikčný model z jedného alebo viac predchádzajúcich kódovaných video snímokov s použitím blokovo založenej kompenzácie pohybu. Medzi dôležité odlišnosti oproti predchádzajúcim štandardom patrí podpora pre viacero veľkostí blokov (od 16 x 16 po 4 x 4) a jemné podvzorkovanie pohybových vektorov. Existuje niekoľko nástrojov pre inter predikciu v základnom profile, ktoré v hlavnom a rozšírenom profile dopĺňajú ešte B-prúžky a váhovaná predikcia [3].

### 2.3.6 Intra predikcia

V intra móde je predikčný blok P vytvorený na základe predchádzajúcich kódovaných a rekonštruovaných blokov a je odpočítaný od aktuálneho bloku pred samotným kódovaním. Pre luminančné vzorky, blok P je vytvorený pre každý 4 x 4 blok alebo 16 x 16 makroblok. Existuje spolu 9 možných predikčných módov pre každý 4 x 4 luminančný blok, 4 módy pre 16 x 16 luminančný blok a 4 módy pre chrominančné komponenty. Enkodér si väčšinou vyberie predikčný mód pre každý blok, čo minimalizuje rozdiely medzi P blokom a blokom, ktorý má byť kódovaný [3].

### 2.3.7 Deblokovací filter

Filter je aplikovaný na každý dekódovaný makroblok, aby zredukoval blokové skreslenie. Deblokovací filter nastupuje v enkodéri po inverznej transformácii (pred rekonštruovaním a uložením makrobloku pre budúcu predikciu) a v dekodéri (pred rekonštruovaním a zobrazením makrobloku). Filter vyhľadí hrany blokov, čím sa zlepši výzor dekódovaných snímok. Filtrovaný obraz je použitý pre predikciu kompenzáciou pohybu budúcich snímok a toto môže zlepšiť účinnosť kompresie, pretože filtrovaný obrázok je často vierohodnejšou reprodukciou originálneho snímku než štvorčekovaný nefiltrovaný obrázok. Enkodér je schopný podľa potreby znížiť „silu“ dekódovacieho filtru alebo ho úplne vyradiť. Defaultná činnosť filtra je nasledovná:

Filtrovanie je aplikované na vertikálne a horizontálne hrany blokov  $4 \times 4$  v makrobloku (s výnimkou hraničných hrán prúžkov), a to v nasledujúcom poradí:

- sú filtrované 4 vertikálne hranice luminančných komponentov
- sú filtrované 4 horizontálne hranice luminančných komponentov
- sú filtrované 2 vertikálne hranice každého chrominančného komponentu
- sú filtrované 2 horizontálne hranice každého chrominančného komponentu

„Sila“ filtra (veľkosť filtrovania) závisí na aktuálnom kvantizéry, kódovacích módoch susedných blokov a na gradiente vzoriek [3].

## 3 Princíp funkcie algoritmu

V tejto kapitole bude popísaný algoritmus od nemeckého autora Arnda Edena, ktorý dokáže vypočítať PSNR pre H.264 kódované videosekvencie bez použitia referencie.

### 3.1 Vývoj za posledné roky

Za posledné roky bolo vyvinutých niekoľko spôsobov ako zmerať kvalitu komprimovaných obrazov videosekvencií. Výsledok tohto vývoja je niekoľko odlišných algoritmov, z ktorých niektoré sú schopné stanoviť kvalitatívnu hodnotu s vysokou koreláciou porovnateľnú vnímanej kvalite. To môže byť dokázané porovnaním s výsledkami subjektívnych testov. Tieto algoritmi môžu byť rozdelené do troch skupín [1].

Prvá skupina porovnáva skreslený video signál s neskresleným referenčným signálom. Vnímaná odlišnosť týchto dvoch signálov je počítaná pomocou modelu nazvanom Systém ľudského videnia (HVS). Pomer signálu k šumu (PSNR - Peak Signal To Noise Ratio) je jeden z algoritmov, ktorý aplikuje veľmi jednoduchý model Systému ľudského videnia. Iné obľúbené algoritmi sú takzvaný Sarnoff model, ktorý počíta chybu signálu v jednotkách JND (just noticeable differences) a Analýza štrukturálnych odlišností. Algoritmi, ktoré používajú neskreslený signál ako referenčný signál sa priradzujú k plno-referenčným algoritmom. Avšak, sú aplikovateľné iba na v niekoľkých prípadoch, pretože vo väčšine aplikácií nie je dostupný neskreslený (nekomprimovaný) referenčný signál [1].

Za účelom zredukovania množstva dát potrebných pre výpočet, niektoré algoritmi vyberajú len určité zložky z referenčného a skresleného signálu. A teda, pre výpočet kvality obrazu sa berú do úvahy iba tieto zložky (vlastnosti) namiesto celej videosekvencie. Tieto takzvané redukované-referenčné algoritmi môžu byť použité na vyhodnotenie kvality obrazu na prijímacej strane prenosu. Avšak, ich použitie vyžaduje implementáciu častí algoritmu v zdroji signálu a prenos „bočných“ informácií v datovom toku. Hoci implementovať takýto systém je technicky realizovateľné, je žiadúce určiť kvalitu obrazu bez toho, aby sme museli vybrať informácie zo zdroja signálu [1].

Tretia skupina algoritmov určuje kvalitu obrazu iba zo skresleného signálu a z bočných informácií poskytnutých v datovom toku. Väčšina z týchto takzvaných bezreferenčných algoritmov je založená na detekcii dobre známych kódových artefaktov. Ako príklad takéhoto algoritmu môže byť algoritmus, kde sú blokové artefakty, uvedené transformačným kodérom, použité ako indikátor kvality. Tieto artefakty sa periodicky opakujú v obraze a teda sú detekovateľné, napríklad pomocou štatistickej analýzy hrán obrazu na okrajoch bloku. Väčšina všetkých bezreferenčných algoritmov je založená na detekcii blokových artefaktov. Blokové indikátory napríklad dobre pracujú pre sekvencie kódované MPEG-2. Ak sú videosekvencie kódované H.264, blokové artefakty sú redukované deblokovacím filtrom, ktorý je súčasťou štandardu. Z toho dôvodu detekcia blokových artefaktov nevedie k prijateľným výsledkom [1].

Iný spôsob je odhadnúť kódovacia chybu z kódovanej videosekvencie. Ako príklad môže byť uvedený algoritmus, kde chybová odchýlka je počítaná z transformačných koeficientov. Pre tento výpočet je odhadnuté rozdelenie transformačných koeficientov. Chybová odchýlka je založená na týchto rozdeleniach. Tento článok uvádza podobný spôsob, ale rozširuje ho do H.264 kódovaných videosekvencií. Okrem toho, je tu uvedená nová metóda odhadu štatistických vlastností transformačných koeficientov [1].

### 3.2 PSNR ako indikátor kvality

Špičkový pomer signálu k šumu (PSNR) je najčastejšie používaný indikátor pre hodnotenie kvality obrazu. Môže byť vypočítaný z jasových hodnôt neskresleného referenčného signálu  $p_{ref}(x, y)$  a skresleného signálu  $p(x, y)$ :

$$PSNR = 10 \cdot \log_{10} \left( \frac{(2^Q - 1)^2}{MSE} \right), \quad (3.1)$$

$$MSE = \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} (p(x, y) - p_{ref}(x, y))^2, \quad (3.2)$$

kde  $MSE$  (Mean Square Error) je stredná kvadratická chyba a  $Q$  je bitová hĺbka jasových hodnôt. PSNR počíta pomer signálu k šumu, kde chybová odchýlka kódovania je interpretovaná ako výkon šumu a špičková hodnota je predpokladaná pre výkon signálu. Typicky, iba jasová zložka sa používa pre hodnotenie kvality [1].

V porovnaní s algoritmi ako „Sarnoff Model“ alebo Analýza štrukturálnych podobností PSNR môže byť interpretované ako veľmi jednoduchý model Systému ľudského videnia (HVS). Predpokladá sa, že HVS je citlivý ku kvadratickým odchýlkam medzi referenčným a skresleným signálom. Tento predpoklad je hrubé zjednodušenie reálneho HVS. Napr. neberie to do úvahy efekty priestorového a dočasného maskovania, ktoré sa objavia na štrukturovaných častiach obrazu a rýchlych pohybov. Tieto efekty môžu zredukovať viditeľnosť artefaktov [1].

Napriek tomu bolo dokázané, že PSNR je dobré kritérium vnímanej kvality. Korelácie s výsledkami subjektívnych testov sú okolo 80 %. To je porovnateľné s realizovaním blokových indikátorov a dokonca s niektorými komplikovanými algoritmi. Preto PSNR môže byť použité ako indikátor pre kvalitu obrazu [1].

### 3.3 H.264 transformácia a kvantovanie

Jedna z inovácií v H.264 je druh transformácie. Namiesto klasickej diskkrétnej kosínovej transformácie (DCT), ktorá sa používa prakticky vo všetkých transformačných kodéroch, je použitá aproximačná DCT. Táto transformácia môže byť vypočítaná kompletne v celočíslenej aritmetike. Aby sme dosiahli túto výhodu, základné funkcie H.264 transformácie nie sú normalizované [1].

V originálnych profiloch H.264 používa bloky o veľkosti 4 x 4 pre transformáciu. S takzvaným rozšírením rozsahu vernosti bol predstavený vyšší profil, ktorý podporuje tiež bloky o veľkosti 8 x 8. Ak je táto vlastnosť využitá, veľkosť bloku sa môže meniť adaptívne pre každý makroblok [1].

Kvantovanie transformačných koeficientov je kontrolované pomocou kvantizačného parametru (QP), ktorý môže byť prispôbený pre každý makroblok. QP riadi veľkosť kvantizačného kroku exponenciálne. Veľkosť kroku QS je daná

$$QS = c \cdot 2^{\frac{QP}{6}}. \quad (3.3)$$

Z tohto dôvodu, môže byť QP interpretovaný ako veľkosť kvantizačného kroku v dB. Čo sa týka kvantovania normalizovaných koeficientov, konštanta  $c$  je rovnaká pre všetky

koeficienty. Vyšší profil povoluje prídavnú kvantizačnú maticu, ktorá môže byť zvolená ľubovoľne. Potom faktor  $c$  je odlišný pre každý koeficient. Avšak, principiálny výpočet veľkosti kvantizačného kroku ostáva nezmenený [1].

V nasledujúcom texte sa predpokladá, že veľkosť jedného transformačného bloku sa nemení pre celý obraz. Ak je použitá adaptívna veľkosť transformačného bloku, principiálna kalkulácia sa nemení. Avšak, výpočet chybových odchýlok musí byť uskutočnený dvakrát, kde každý raz sa berie do úvahy iba makroblok s jednou špecifickou veľkosťou bloku. Potom sú skombinované dve chybové odchýlky do jednej celkovej hodnoty, z ktorej je zistené PSNR [1].

### 3.4 Výpočet kódovacej chyby v transformačnej oblasti

Pre výpočet PSNR musí byť zistený výkon kódovacej chyby, ktorý je ekvivalentný ku strednej kvadratickej chybe (MSE). Ako transformácia pre transformačné kódovanie v H.264 je použitá ortogonálna transformácia, výkon môže byť vypočítaný v priestorovej oblasti práve tak ako v transformačnej oblasti. Táto vlastnosť je spôsobená Parsevalovým teorémom. Avšak, základné funkcie transformácie nie sú normalizované. A tak podiel každého koeficientu musí byť škálovaný korekčným faktorom. Výpočet výkonu pre jeden transformačný blok môže byť napísaný nasledovne:

$$P = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \frac{F^2(i, j)}{a_{ij}}, \quad (3.4)$$

kde  $F$  je jeden makroblok transformačných koeficientov pre jeden blok obrazu.  $N$  je veľkosť bloku, ktorá môže byť 4 alebo 8 v obrazoch kódovaných H.264. A  $a_{ij}$  je normalizačná konštanta pre koeficient na pozícii  $(i, j)$  vnútri bloku. Ak nie je známy konkrétny koeficient, ale pre každý zo 4 x 4 alebo 8 x 8 koeficientov je daná funkcia hustoty pravdepodobnosti (pdf), výkon môže byť vypočítaný ako suma momentov druhého stupňa každého pdf  $p_{ij}(\lambda)$

$$P = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \int_{-\infty}^{\infty} \left( \frac{\lambda}{a_{ij}} \right)^2 p_{i,j}(\lambda) d\lambda, \quad (3.5)$$

kde  $\lambda$  symbolizuje hodnotu koeficientu. V transformačnej oblasti sa kódovacia chyba vyskytne ako kvantizačná chyba. Obecne, kvantizačná chyba koeficientu od  $k$ -teho kvantizačného intervalu je

$$\lambda - q_k, \quad (3.6)$$

kde  $q_k$  je  $k$ -ta hodnota kvantizácie. Pre výpočet výkonu kvantizačnej chyby musí byť po častiach vyriešený integrál pre každý kvantizačný interval  $[a_k, b_k]$ .

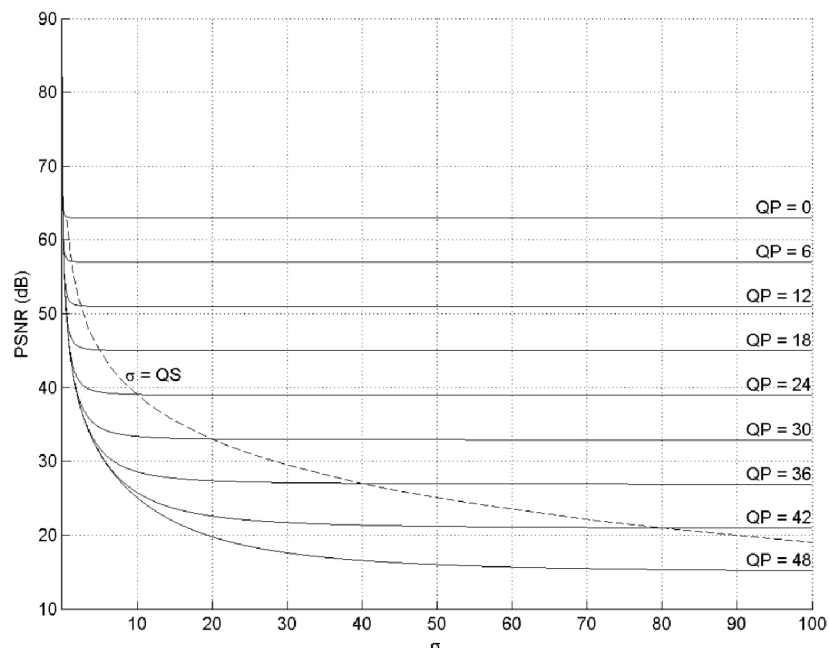
$$P = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \sum_{k=0}^{\infty} \int_{a_k}^{b_k} \left( \frac{\lambda - q_k}{a_{ij}} \right)^2 p_{i,j}(\lambda) d\lambda + \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \sum_{k=-\infty}^0 \int_{b_k}^{a_k} \left( \frac{\lambda - q_k}{a_{ij}} \right)^2 p_{i,j}(\lambda) d\lambda, \quad (3.7)$$

kde  $a_k$  a  $b_k$  sú parametre, ktoré môžu byť principiálne zmenené kodérom. Najbežnejšia možnosť je nastaviť

$$\begin{aligned}
 a_0 &= 0 & (3.8) \\
 a_k &= \text{sign}(k) \cdot (|k| - 1 + \alpha) \cdot QS & \text{pre } |k| = 1.. \infty \\
 b_k &= \text{sign}(k) \cdot (|k| + \alpha) \cdot QS & \text{pre } |k| = 0.. \infty \\
 q_k &= k \cdot QS & \text{pre } |k| = 0.. \infty .
 \end{aligned}$$

$QS$  je veľkosť kvantizačného kroku. Ak  $\alpha = 0,5$ , potom je kvantovanie vykonané jednoduchou zaokrúhľovacíou operáciou. Ak je pdf symetrické, potom suma v (2.7) môže byť vyhodnotená iba pre kladné hodnoty  $k$  a násobená dvomi. Ak je použité adaptívne kvantovanie,  $P$  musí byť vypočítané pre každý kvantizačný parameter a skombinované do jednej hodnoty a musí sa brať ohľad na frekventovanosť makroblokov s každým QP [1].

Obr. 4 ukazuje PSNR ako funkciu smerodajnej odchýlky  $\sigma$  transformačného koeficientu. Každá krivka predstavuje inú hodnotu QP. Možno si všimnúť, že PSNR saturuje pre vysoké hodnoty  $\sigma$ . Hodnoty saturácie závisia lineárne na QP. To je spôsobené logaritmickými vlastnosťami QP. Tieto saturačné hodnoty korešponujú s výkonom kvantizačného šumu pre rovnomerne rozložené koeficienty a stanovujú odhad najhoršieho prípadu PSNR. Realistické hodnoty  $\sigma$  sú blízke alebo menšie ako kvantizačný krok  $QS$ . Teda, najdôležitejšia časť obr. 4 je na ľavo od prerušovanej čiary. V tejto oblasti PSNR významne závisí na  $\sigma$ . Z tohto dôvodu môže byť odhad optimalizovaný v porovnaní s odhadom najhoršieho prípadu aplikovaním odhadu  $\sigma$  [1].



**Obr. 4** Závislosť PSNR na  $\sigma$  a QP pre jeden Lapliac. rozdelený koeficient s  $\alpha = 0,5$  [1]

Krivky dávajú tiež informáciu o presnosti odhadu. Na malých hodnotách  $\sigma$  a veľkých hodnotách QP sú sklony kriviek väčšie. A preto malé chyby v odhade  $\sigma$  vedú k porovnateľne veľkým chybám v odhade PSNR, zatiaľ čo pre veľké hodnoty  $\sigma$  a malé hodnoty QP je odhad veľmi dobrý. Okrem toho, podhodnotenie  $\sigma$  má horší efekt ako jeho nadhodnotenie [1].

### 3.5 Odhad rozdelenia transformačných koeficientov

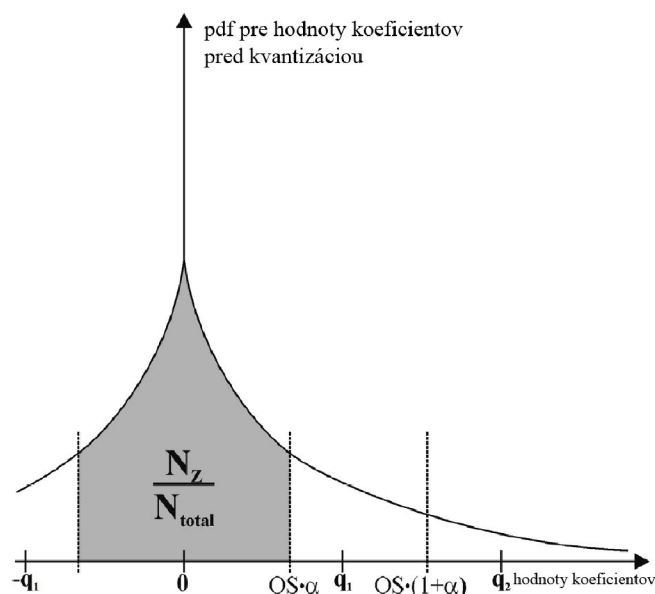
Predchádzajúca kapitola ukazuje ako môže byť PSNR vypočítané v transformačnej oblasti, ak je známe pdf transformačných koeficientov. Veľmi jednoduchý odhad PSNR môže byť zistený predpokladaním rovnomerného rozkladu koeficientov. V tomto prípade PSNR každého koeficientu by bolo nastavené na príslušnú saturačnú hodnotu v obr. 4. Pre sofistikovanejší odhad musia byť urobené nejaké predpoklady pdf koeficientov. Obľúbený predpoklad je, že koeficienty DCT sú Laplaciánsky rozdelené. Niekedy sa predpokladá tiež Gaussovo alebo Cauchyho rozdelenie. V nasledujúcom texte je predpokladané Laplaciánske rozdelenie. Toto rozdelenie sa javí ako dobrý odhad pre reálne rozdelenie H.264 transformačných koeficientov a takisto sa dobre spracováva matematicky. Jeho pdf je

$$p(\lambda) = \frac{1}{2\sigma} e^{-\frac{|\lambda|}{\sigma}}. \quad (3.9)$$

Priama metóda ako odhadnúť pdf môže byť odvodená z počtu transformačných koeficientov, ktoré sú kvantované k nule. Tieto môžu byť jednoducho určené z kódovaných koeficientov. Percentuálny podiel nulových hodnôt pre každý transformačný koeficient môže byť interpretovaný ako pravdepodobnosť, že jednotlivé koeficienty sú kvantované k nule. Rovnaká pravdepodobnosť môže byť odvodená z pdf integrovaním cez prvý kvantizačný interval. Pre Laplaciánske rozdelenie môže byť parameter  $\sigma$  vypočítaný analyticky:

$$\frac{N_z}{N_{total}} = \int_0^{\alpha \cdot QS} \frac{1}{\sigma} e^{-\frac{\lambda}{\sigma}} d\lambda \Rightarrow \sigma = -\frac{\alpha \cdot QS}{\ln\left(1 - \frac{N_z}{N_{total}}\right)}, \quad (3.10)$$

kde  $N_z$  je počet nulových transformačných koeficientov a  $N_{total}$  je celkový počet koeficientov. Odhad pdf môže byť vypočítaný pre každý koeficient. Obr. 5 ilustruje tento spôsob [1].



**Obr. 5** Závislosť počtu nulových koeficientov a pdf na hodnote koeficientu  $\lambda$  [1]

### 3.6 Vyhodnotenie algoritmu

Pre vyhodnotenie algoritmu bolo použitých 5 videosekvencií. Tieto videosekvencie boli kódované H.264 s použitím troch rôznych softvérových kodérov. Pre prvé testy bol použitý softvér, ktorý bol vyvinutý na Fraunhofer Heinrich Hertz Institut (HHI). Pre tento kodér bol algoritmus kalibrovaný zmenami parametru  $\alpha$ . Potom boli testy rozšírené do Nero Recode Software, ktorý implementuje verziu Ate me codec a nakoniec voľne dostupný x264. Pre tieto kodéry nebola vykonaná žiadna prídavná kalibrácia [1].

Pre každú kódovanú videosekvenciu bolo vypočítané PSNR s použitím nekomprimovaného signálu ako referenčného. A PSNR bolo tiež odhadnuté pomocou navrhovaného algoritmu. Po kalibrácii bol parameter  $\alpha$  nastavený na  $\alpha = 0,65$ ;  $\alpha = 0,92$  a  $\alpha = 1,08$  pre I-, P- a B-snímky. Bez kalibrácie by bol parameter  $\sigma$  systematicky podhodnotený. A teda PSNR je nadhodnotený. Táto kalibrácia kompenzuje odlišné javy. Jeden jav je, že pdf nie je presne Laplaciánske. Navyše, kvantovanie môže byť uskutočnené použitím „mrtveho pásma“. To znamená, že kvantizačná prahová úroveň leží bližšie k vyššiemu susednému kvantizačnému kroku ako k nižšiemu. V P a B snímkoch môže byť tiež vplyv vynechaných makroblokov. Tieto makroboky nenesú žiadnu informáciu. A preto, pre ne neexistuje žiadny vierohodný kvantizačný parameter (QP). Je možné, že QP, ktorý je skutočne použitý na kvantovanie týchto makroblokov je väčší ako ten, ktorý je použitý na susedné makrobloky. Avšak, pre odhad, najpraktickejší predpoklad je QP predchádzajúceho bloku v dekodovacom poradí, ktorý ale môže viesť ku nepresnostiam [1].

**Tab. 2:** Porovnanie skutočného PSNR a odhadnutého PSNR pre 720p50 videosekvencie

Kodér	Typ snímku	$\mu_{\varepsilon}$ [dB]	$\sigma_{\varepsilon}$ [dB]	Korelácia [%]
<b>HHI</b>	I	-0,01	0,82	99,4
	P	0,04	1,12	98,9
	B	0,17	1,57	97,6
	všetky	0,17	1,41	98,1
<b>Nero</b>	I	-0,77	0,76	99,6
	P	-1,62	1,23	98,5
	B	1,48	1,84	95,6
	všetky	0,09	2,06	94,1
<b>X264</b>	I	-0,24	0,88	99,7
	P	-0,69	1,13	99,0
	B	0,02	1,45	97,6
	všetky	-0,34	1,33	98,2

**Tab. 3:** Porovnanie skutočného PSNR a odhadnutého PSNR pre 1080i25 videosekvencie

Kodér	Typ snímku	$\mu_{\varepsilon}$ [dB]	$\sigma_{\varepsilon}$ [dB]	Korelácia [%]
<b>HHI</b>	I	-0,11	0,96	99,5
	P	0,04	1,15	98,7
	B	-0,71	1,30	98,4
	všetky	-0,43	1,30	98,5

Tab. 2 ukazuje výsledok porovnania medzi skutočným PSNR a odhadnutým PSNR pre 720p50 videosekvencie a tab. 3 výsledky pre 1080i25 videosekvencie. Korelácia medzi dvoma hodnotami leží nad 95 %, dokonca aj pre B snímky, kde je odhad porovnateľne horší. Pre I snímky je korelácia nad 99 %. Stredná chyba  $\mu_{\varepsilon}$  a štandardná odchýlka  $\sigma_{\varepsilon}$  chyby dovoľuje sofistikovanú analýzu výsledkov. Je možné si všimnúť, že pre I snímky môže byť



dosiahnutá vysoká presnosť. Štandardná odchýlka leží jednoznačne pod 1 dB. Pre P a B snímky je presnosť nižšia ale stále dobrá. Toto pravdepodobne spôsobuje fakt, že I snímky obsahujú väčšinu informácií v ich zobrazení predikčnej chyby, zatiaľ čo v P a B snímkoch už kompenzácia pohybu dáva dobrú predikciu obrazu. Toto vedie k menším hodnotám  $\sigma$  v P a B snímkoch, ktoré zvyšuje vplyv chyby odhadu zavinenú sklonom chybovej funkcie. Pre nízke hodnoty PSNR, ktoré odpovedajú hrubému kvantovaniu, je odhad menej presný. To môže byť vysvetlené rovnakým javom. Dodatočne, pre hrubšie kvantovanie je viac transformačných koeficientov kvantovaných k nule. Toto môže viesť tiež k nadhodnoteniu kódovacej chyby [1].

V nasledujúcej kapitole bude okrem iného popísaný môj vlastný výskum, v ktorom bude zisťovaná korelácia odhadnutých a skutočných hodnôt PSNR. Zistená korelácia bude porovnaná s výsledkami uvedenými v tejto kapitole.

### 3.7 Výpočet MSE

V tejto časti bude odvodený výraz pre výpočet strednej kvadratickej chyby pre jeden transformačný koeficient. Predpokladá sa Laplaciánske rozdelenie hodnoty koeficientu a kvantovanie splňuje (3.8). Potom stredná kvadratická chyba kvantovania je daná

$$MSE = \sum_{k=0}^{\infty} \int_{a_k}^{b_k} (x - q_k)^2 \cdot \frac{1}{\sigma} e^{-\frac{\lambda}{\sigma}} d\lambda = \sum_{k=0}^{\infty} \left[ -e - \frac{\lambda}{\sigma} \left( (\lambda - q_k)^2 + 2\sigma(\lambda - q_k) + 2\sigma^2 \right) \right]_{b_k}^{a_k}$$

a po zjednodušení

$$MSE = 2\sigma^2 - ((2\alpha - 1)QS^2 + 2\sigma QS) \cdot \frac{e^{\frac{\alpha QS}{\sigma}}}{1 - e^{\frac{QS}{\sigma}}}. \quad (3.12)$$

## 4 H.264/AVC referenčný softvér

### 4.1 Štruktúra

Autorom tohoto softvéru je organizácia ITU (International Telecommunications Union), ktorá sídli v Ženeve. Jej poslaním je udržiavať a prehľbovať medzinárodnú spoluprácu vo všetkých oblastiach telekomunikácií a rádiokomunikácií, podporovať rozvoj zodpovedajúcich technických prostriedkov, dbať o účelné rozdeľovanie kmitočtového spektra a koordináciu jeho využívania a poskytovať technickú pomoc v oblasti telekomunikácií a rádiokomunikácií [4].

Jedná sa o enkodér a dekodér štandardu H.264. Je optimalizovaný pre Visual Studio 6. Obsahuje workspace „tml.dsw“, ktorý zahŕňa 3 projekty:

- H.264/AVC referenčný enkodér
- H.264/AVC referenčný dekodér
- rtpdump – nástroj pre analýzu obsahu RTP paketov

### 4.2 Referenčný enkodér

Ako vstup pre referenčný enkodér bola použitá voľne dostupná videosekvencia „foreman.yuv“ (snímok z nej je na obr. 6). Z nej bolo kódovaním vytvorených 7 odlišných videosekvencií, ktoré pozostávajú z 50 snímokov (kvôli časovej náročnosti výpočtu). Pri ich tvorbe boli takmer všetky vstupné parametre enkodéru nastavené rovnako okrem kvantizačného parametru pre jednotlivé snímky, a to spôsobilo odlišnú kvalitu vytvorených videosekvencií. To aby mali vzniknuté videá odlišnú kvalitu bolo žiadané kvôli výpočtu korelácie skutočných a odhadnutých hodnôt PSNR.



Obr. 6 Snímok z videosekvencie „foreman.yuv“.

Referenčný enkodér má pre svoju funkciu definované vstupné parametre v konfiguračnom súbore „encoder.cfg“. V tab. 4 sú uvedené parametre, ktoré sú rovnaké pre všetky videosekvencie, v tab. 5 sú potom hodnoty kvantizačného parametru, ktoré sú pre každú videosekvenciu iné. Enkodér sa spúšťa z príkazového riadku príkazom „lencod“.

**Tab. 4:** Parametre rovnaké pre všetky videosekvencie

Vstup	foreman.yuv
Číslo začiatočného snímku	0
Počet enkódovaných snímkov	50
Počet snímkov za sekundu	30
Počet riadkov	288
Počet pixelov v riadku	352
YUV formát	4:2:0
Trace súbor	trace_enc.txt
Rekonštruovaný súbor	foreman_rec.yuv
Výstup	foreman.264
Profil	základný
Bitový tok	45,020 kb/s

**Tab. 5:** Kvantizačný parameter a približná kvalita videosekvencií

Názov	Kvantizačný parameter [-]	Približné PSNR [dB]
foreman1.264	28	37
foreman2.264	32	35
foreman3.264	35	33
foreman4.264	38	31
foreman5.264	42	29
foreman6.264	46	26
foreman7.264	48	23

### 4.3 Referenčný dekodér

Najdôležitejšou časťou softvéru pre túto prácu je práve dekodér. Po zorientovaní sa v zdrojovom kóde boli nájdené premenné potrebné pre aplikáciu algoritmu na vyčíslenie PSNR bez použitia referencie, ktorý bol podrobne predstavený v kapitole 3. Dôležité globálne premenné teda sú:

- $img->coeff[i0][j0][i][j]$  – do tejto premennej sa ukládajú transformačné koeficienty jasového signálu a chrominančného signálu
- $qp$  – tu sa ukládajú kvantizačné parametre pre jednotlivé snímky
- $BitDepthLuma$  – bitová hĺbka jasového signálu, defaultne má hodnotu 8 (povolený rozsah je až do 12)

Ako už bolo povedané, v premennej  $img->coeff[i0][j0][i][j]$  sa nachádzajú transformačné koeficienty jasového a chrominančného signálu. Mňa však zaujímajú len jasové vzorky, a tak pomocou jednoduchej podmienky, že rozmer  $j0$  danej premennej môže nadobúdať len hodnoty 0, 1, 2 alebo 3 boli odrezané koeficienty pre chrominančný signál a zostali len tie jasové. Implementáciou jednoduchého počítadla, v ktorom sa berú za radom koeficienty, ako idú a porovnávajú, či sú rovné nule, sa inkrementuje počítadlo *qual-*

$>total\_koef$  (celkový počet koeficientov) a počítadlo  $qual \rightarrow zero\_koef$  (počet nulových koeficientov). Keďže sú už známe všetky potrebné hodnoty, je možné podľa vzorca (3.10) určiť parameter  $\sigma$ , čo je štandardná odchýlka transformačných koeficientov. Z parametra  $\sigma$  je ďalej možné pomocou (3.12) vypočítať strednú kvadratickú chybu MSE. No a napokon z MSE dosadením do (3.1) stanovíme hodnotu špičkového pomeru signálu k šumu PSNR. Tieto výpočty boli doplnené do zdrojového kódu „image.c“ do funkcie „find\_snr“.

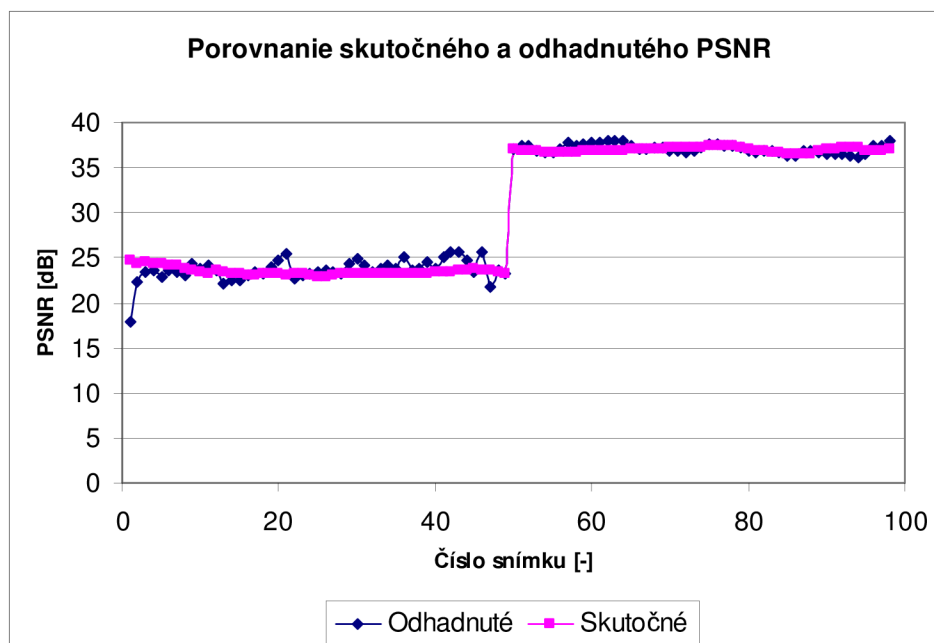
#### 4.4 M-file „evaluate“

M-file „evaluate“ je mnou vytvorený krátky zdrojový kód v programe Matlab, ktorý slúži hlavne na kalibráciu parametra  $\alpha$ . Postup je vysvetlený v kapitole 6.2.

Vstupnými datami pre tento m-file sú \*.mat súbory generované dekodérom pri dekódovaní videosekvencie. V ich premenných sa nachádzajú potrebné hodnoty: reálny celkový počet transformačných koeficientov v  $coef\_total$ , počet nulových koeficientov v  $coef\_zero$ , a napokon reálne hodnoty kvantizačných parametrov pre jednotlivé snímky nachádzajúce sa v premennej  $quant$ .

#### 4.5 Overenie správnosti výpočtu

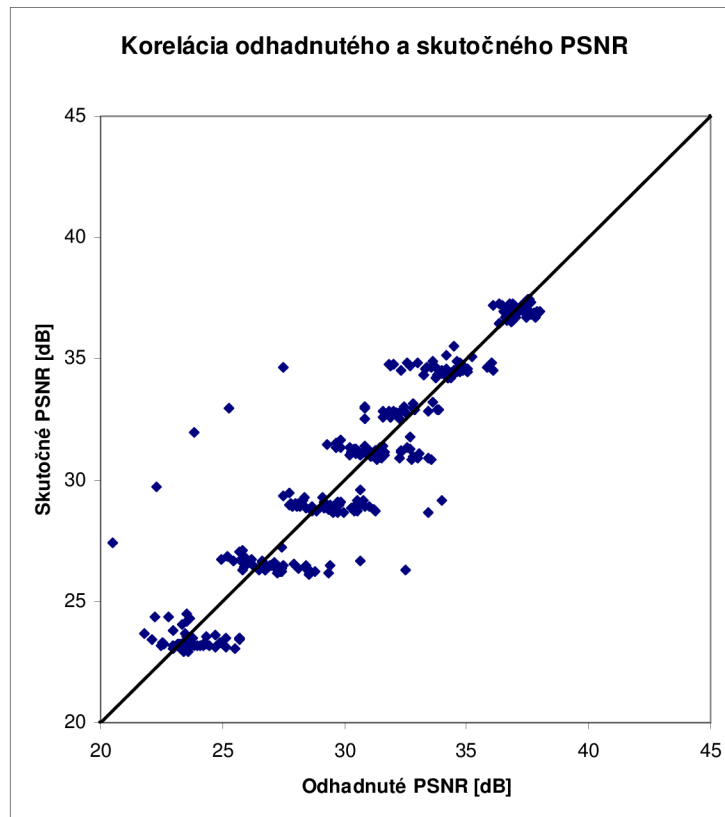
Pre všetkých 7 zakódovaných videosekvencií boli vyčíslené skutočné aj odhadnuté hodnoty PSNR. Skutočné hodnoty PSNR počítané algoritmom implementovaným v dekodéry nám vynikajúco poslúžia na overenie správnosti výpočtu. Obr. 7 ukazuje grafické porovnanie skutočných a odhadnutých hodnôt PSNR dvoch videosekvencií. Vyhodnotenie bolo prevedené tak, že prvých 50 snímkov je z videosekvencie foreman7.yuv a zvyšných 50 z foreman1.yuv.



Obr.7 Porovnanie skutočného a odhadnutého PSNR

Korelácia odhadnutých a skutočných hodnôt PSNR bola vypočítaná pre všetkých dostupných 350 snímkov (50 x 7) a číselne vychádza na 96,20 %. Jej výpočet je takisto

súčasťou súboru „evaluate.m“. Na obr. 8 je bodový graf zodpovedajúci danej korelácii. Je na ňom tiež vyznačená priamka  $y = x$ , ku ktorej by sa body mali ideálne čo možno najviac približovať.



**Obr. 8** Korelácia odhadnutého a skutočného PSNR

Korelácia hodnôt bola počítaná podľa vzťahu

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y}, \quad (4.1)$$

kde  $s_x$  a  $s_y$  sú štandardné odchýlky a tie sa počítajú podľa

$$s_{x,y} = \sqrt{\frac{(x_1 - \mu)^2 + (x_2 - \mu)^2 + \dots + (x_N - \mu)^2}{N}}. \quad (4.2)$$

# 5 Optimalizácia

## 5.1 Analýza zdrojových kódov referenčného dekodéru

Na úvod je potrebné analyzovať všetky zdrojové kódy referenčného dekodéru, aby bolo jasné, ktorý sa o čo stará, aké ma vstupy a aké z neho vychádzajú výstupy.

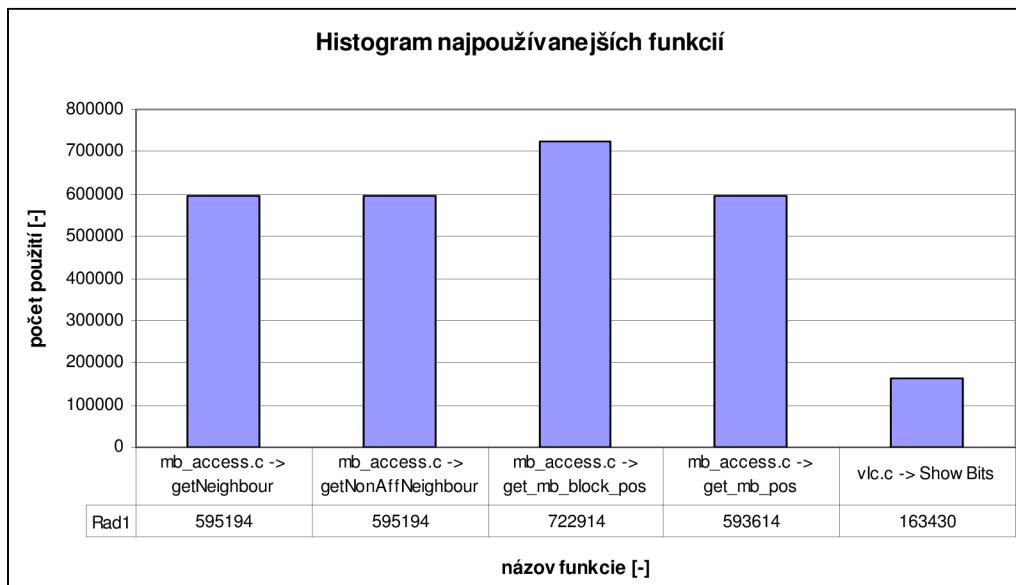
- `annexb.c` – vracia veľkosť jednotky NALU (Network Abstraction Layer Unit), to znamená, počet bitov medzi štartovacími symbolmi
- `biaridecod.c` – zabezpečuje binárne aritmetické operácie dekodéru
- `block.c` – operácie na úrovni jednotlivých blokov v makrobloku ako napríklad inverzná transformácia jasového aj oboch chrominančných signálov, predikcia (horizontálna, vertikálna), výpočet kvantizačného parametra
- `cabac.c` – CABAC entropické dekódovanie
- `context_ini.c` – inicializácia obsahu CABAC
- `erc_api.c` – rozhranie pre modul na ukrytie chyby
- `erc_do_i.c` – algoritmus na ukrytie chyby pre I-snímky
- `erc_do_p.c` – algoritmus na ukrytie chyby pre P-snímky
- `errorconcealment.c` – algoritmus na ukrytie chyby; ak je detekovaný chybný element, je označený ako chybný; ak si dekodér vyžiada element, najprv si zistí, či je chybný alebo nie, ak áno, nahradí ho opraveným elementom
- `filehandle.c` – operácie spojené s takzvaným *trace file*
- `fmo.c` – zodpovedný za flexibilné radenie makroblokov, generovanie konkrétnej mapy, ktorá bola pri tomto procese použitá (typy 0 až 6)
- `header.c` – spracovávanie hlavičiek jednotlivých prúžkov
- `image.c` – dekódovanie jednotlivých snímkov, výpočet PSNR pre každý snímok na základe porovnania skreslenej videosekvencie s referenčnou (a to pre jasový signál aj oba chrominančné), výpočet priemerných hodnôt PSNR pre každý signál, inicializácia poľa pre referenčné snímky, na konci dekódovania obrázku ukryť chyby a uložiť obrázky do bufferu pre dekódované snímky; dekódovanie jednotlivých prúžkov, sledovanie prípadnej straty snímku
- `ldecod.c` – výpis takzvaných help správ, výpis hlavičky a úvodných informácií o dekodéri, meranie času dekódovania jedného snímku a celkového času dekódovania, čítanie vstupov z konfiguračného súboru, výpis výstupných hodnôt
- `leaky_bucket.c` – spracováva parametre pre algoritmus leaky bucket
- `loopFilter.c` – deblokovací filter na zredukovanie blokových artefaktov vyhladením hrán makroblokov
- `macroblock.c` – inicializácia a dekódovanie jednotlivých makroblokov, mód makrobloku, nastavenie vektora predikcie pohybu, získanie informácie o predikcii pohybu z NAL, predikcia zo susedných blokov pre jasový a oba chrominančné signály, inverzné kvantovanie, spracovávanie transformačných koeficientov a operácie s nimi
- `mb_access.c` – spracovávanie susedných makroblokov pre účely predikcie
- `mbuffer.c` – buffer pre ukládanie snímkov pre účely referencie, generovanie listu 0 a 1
- `memalloc.c` – alokácia pamätí pre rôzne 2D, 3D a 4D polia
- `nal.c` – konvertuje EBSP (Encapsulated Byte Sequence Packets) do RBSP (Raw Byte Sequence Packets), a potom do SODB (String Of Data Bits)
- `nal_part.c` – NAL pre časť súboru

- `nalu.c` – podporné funkcie pre dekodér NALU
- `nalucommon.c` – podporné funkcie pre bežné NALU
- `output.c` – spracováva výstupné obrázky
- `parset.c` – nastavovanie rôznych parametrov videosekvencie a obrázkov
- `parsetcommon.c` – generovanie a ovládanie parametrov videosekvencie a obrázkov
- `quality.c` – vyhodnotenie kvality skreslenej videosekvencie a export premenných do matlabu
- `rtp.c` – spracovávanie RTP paketov
- `sei.c` – interpretácia SEI (Supplemental Enhancement Information) správ
- `transform8x8.c` – funkcie pre transformáciu blokov 8 x 8
- `vlc.c` – podporné funkcie pre kódovanie s premennou dĺžkou slova

## 5.2 Sumár použitých funkcií

Na všetky zdrojové kódy referenčného dekodéru bol aplikovaný parser, ktorý do každej funkcie vložil jeden riadok kódu navyše s príkazom *printf*, ktorého parametrami sú: názov danej funkcie a názov zdrojového súboru, v ktorom sa funkcia nachádza (napríklad: `printf("image.c -> decode_one_frame\n");`). Týmto spôsobom bol získaný presný prehľad o tom, čo a v akom poradí je vykonávané pri spustení referenčného dekodéru a dekódovaní komprimovanej videosekvencie a to úplne od začiatku až do konca. Keďže výstupné data sú rozsahovo veľmi veľké, bola pre tento účel vytvorená len veľmi krátka videosekvencia, ktorá sa skladá zo 4 snímok. Svoj účel to ale splnilo a bol získaný podrobný výpis všetkých použitých funkcií. Tento výpis bol ďalej exportovaný do textového súboru *vypis.txt*.

Ďalším užitočným nástrojom na presné pochopenie funkcie dekodéru je zosumarizovanie použitých funkcií. Toto bolo vykonané a do súboru *sumar.txt* bol teda uložený sumár, ktorý hovorí o tom, koľkokrát bola použitá ktorá funkcia. Keďže je tých funkcií príliš veľa (celkom 169), bol vykreslený histogram len piatich najpoužívanejších (obr. 9). Z obrázka je pekne vidno, že najčastejšie dekodér vykonáva operácie s jednotlivými makroblokmi a ich susedmi za účelom predikcie. Tieto procesy sú pre správnu funkciu dekodéru nevyhnutné a preto nemôžu byť odstránené. Ostatné funkcie majú veľmi výrazne nižší počet vykonaní. Najbližšia najvyššia hodnota počtu vykonaní funkcie je „len“ 38016, ktorá patrí funkcii pre vykonanie inverznej transformácie jednotlivých blokov, nachádza sa v zdrojovom kóde `block.c`.



Obr. 9 Histogram najpoužívanějších funkcií

### 5.3 Závěrečná úprava kódu

Po predchádzajúcich zisteniach mohla byť prevedená záverečná úprava kódu s cieľom optimalizovať zdrojový kód referenčného dekodéru tak, aby mal len jediná funkciu, a to výpočet hodnôt PSNR pre jednotlivé snímky bez nutnosti použitia referencie. Bol ponechaný aj užitočný údaj o čase, aby bolo vidno, ako dlho proces trvá, a teda koľko času optimalizácia ušetrila.

Niektoré zdrojové kódy neboli potrebné, a preto boli odstránené. Napríklad nie je potrebný deblokovací filter, ktorého úlohou je vyhladiť hrany makroblokových štruktúr, aby bol minimalizovaný efekt blokových artefaktov v dekódovanej videosekvencii. To je síce veľmi užitočná vec, ale keďže dekódovaná videosekvencia nebude zobrazovaná, je zbytočné, aby túto funkciu môj dekodér mal. Rovnako je zbytočné ukrývanie chýb, ak videosekvencia nemá byť zobrazovaná.

Odstránené zdrojové kódy teda sú:

- erc\_do\_i.c
- erc\_do\_p.c
- errorconcealment.c
- leaky\_bucket.c
- loopFilter.c
- output.c
- rtp.c
- sei.c

Naopak iné zdrojové kódy sú zas nevyhnutné pre správnu funkciu dekodéru. Napríklad kódy, ktoré zabezpečujú spracovávanie blokov v jednotlivých makrobloch, procesy inverznej transformácie, entropického dekódovania, predikcie pohybu, alokácie pamätí pre potrebu rôznych premenných atď.



Takže bez zmeny museli ostat' zdrojové kódy:

- annexb.c
- biaridecod.c
- cabac.c
- context\_ini.c
- erc\_api.c
- filehandle.c
- mb\_access.c
- memalloc.c
- nal.c
- nal\_part.c
- nalu.c
- nalucommon.c
- transform8x8.c
- vlc.c

No a v ostatných zdrojových kódach boli uskutočnené buď rozsiahlejšie alebo menšie úpravy ako napríklad zmazanie niektorých funkcií alebo zmazanie časti funkcie (napr. zmazanie časti funkcie, ktorá sa stará o chrominančné signály – pre túto prácu nepotrebné).

Výsledok celého tohoto procesu je, že bolo ušetrených cca 19,51 % výpočetného výkonu. K tomuto číslu sa došlo tak, že bolo uskutočnených 5 meraní dekódovania rovnakej videosekvencie a to jak s pôvodným dekodérom, tak s optimalizovaným. Bol zapísaný celkový čas trvania procesu oboch dekodérov a vypočítaný ušetrený čas a z toho potom aj percentuálny podiel ušetreného výpočetného výkonu (tab. 6).

**Tab. 6:** Trvanie dekódovacieho procesu a ušetrený výpočetný výkon

Poradové číslo merania	Čas trvania pôvodného dekodéru [s]	Čas trvania optimalizovaného dekodéru [s]	Ušetrený čas [s]	Ušetrený výpočetný výkon [%]
1.	34,974	43,000	8,026	18,67
2.	34,158	43,123	8,965	20,79
3.	34,797	42,955	8,158	18,99
4.	34,638	43,060	8,422	19,56
5.	34,640	43,062	8,422	19,56
<b>Priemer:</b>				19,51

## 6 Použitie referenčného dekodéru

### 6.1 Konfiguračný súbor

Referenčný dekodér má vstupné data definované v konfiguračnom súbore s názvom „dec\_muj.cfg“, ktorý sa nachádza v adresári „bin“. Najdôležitejšie vstupné parametre sú (v zátvorke budú napísané konkrétne nastavenia pre videosekvenciu „foreman.264“):

- názov vstupnej komprimovanej videosekvencie pre dekodér (foreman.264)
- názov výstupnej dekódovanej videosekvencie (foreman\_dec.yuv)
- názov referenčnej videosekvencie pre výpočet PSNR (foreman\_rec.yuv)
- YUV formát (4:2:0)
- parameter  $\alpha$  pre jednotlivé typy snímkov I, P, B, SI, SP (0,43 a 0,77)
- NAL mód (Annex B)
- offset pre výpočet PSNR (0)
- POC škála (2)
- názov konfiguračného súboru pre „Leaky Bucket“ (leakybucketparam.cfg)
- použitie ukrytia chyby (nepoužitý)

### 6.2 Kalibrácia parametra $\alpha$

Aby referenčný dekodér fungoval správne, je potrebné nastaviť v konfiguračnom súbore korektné hodnoty parametra  $\alpha$  pre všetky typy snímkov, z ktorých sa videosekvencia skláda. Kalibrácia spočíva v tom, že sa vezmu skutočné hodnoty PSNR a hodnoty  $\alpha$  sa menia tak, aby medzi skutočnými a odhadnutými hodnotami PSNR boli najmenšie možné rozdiely. Keď je známe, aké hodnoty  $\alpha$  prislúchajú danej videosekvencii, sú potom nastavené v konfiguračnom súbore. Od tej doby dekodér dáva na výstupe správne hodnoty PSNR pre jednotlivé snímky.

### 6.3 Obsluha referenčného dekodéru

Dekodér sa spúšťa z príkazového riadku príkazom „ldecod“, pričom je ešte potrebné definovať konfiguračný súbor „dec\_muj.cfg“. Po potvrdení spúšťačieho príkazu, začne dekodér postupne spracovávať jednotlivé snímky a ako výstup vracia odhadnuté hodnoty PSNR a čas, za ktorý bol každý snímok spracovaný. Na obr. 10 je ukážka dekódovania niekoľkých začiatkových snímkov spolu s hlavičkou, kde je vysvetlené, ktorá hodnota čo znamená. Po ukončení procesu dekódovania je známy celkový čas jeho trvania (obr. 11).

```

C:\WINDOWS\system32\cmd.exe - ldecod dec_muj.cfg
Microsoft Windows XP [Verze 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

D:\Skola\MM2E\JM2_3\bin>ldecod dec_muj.cfg
----- JM 11.0 (PRExt) -----
Decoder config file           : dec_muj.cfg
-----
Input H.264 bitstream        : foreman.264
Output decoded YUV           : foreman_dec.yuv
Output status file           : log.dec
Input reference file          : foreman.yuv
-----
POC must = frame# or field# for SNRs to be correct
-----
Frame      POC    Pic#   QP    SnrY    Y:U:V   Time[ms]
-----
0000(I)    0     0     28   26.6736  4:2:0   578
0001(P)    2     1     28   39.5894  4:2:0   312
0002(P)    4     2     28   38.7817  4:2:0   344
0003(P)    6     3     28   38.6368  4:2:0   344
0004(P)    8     4     28   38.6235  4:2:0   344
0005(P)   10     5     28   38.7797  4:2:0   344
0006(P)   12     6     28   38.9560  4:2:0   328
0007(P)   14     7     28   38.9978  4:2:0   343
0008(P)   16     8     28   38.8766  4:2:0   344
0009(P)   18     9     28   38.6028  4:2:0   344
0010(P)   20    10     28   38.8473  4:2:0   329
0011(P)   22    11     28   39.0023  4:2:0   329

```

Obr. 10 Začiatok dekódovacieho procesu

```

C:\WINDOWS\system32\cmd.exe
0073(P)    146    9     28   38.7636  4:2:0   359
0074(P)    148   10     28   39.0473  4:2:0   406
0075(P)    150   11     28   39.4261  4:2:0   375
0076(P)    152   12     28   39.3938  4:2:0   344
0077(P)    154   13     28   39.3025  4:2:0   375
0078(P)    156   14     28   39.2124  4:2:0   359
0079(P)    158   15     28   39.1002  4:2:0   375
0080(P)    160    0     28   38.6501  4:2:0   375
0081(P)    162    1     28   38.5309  4:2:0   515
0082(P)    164    2     28   38.6028  4:2:0   375
0083(P)    166    3     28   38.6770  4:2:0   375
0084(P)    168    4     28   38.5038  4:2:0   359
0085(P)    170    5     28   38.1500  4:2:0   375
0086(P)    172    6     28   38.1578  4:2:0   422
0087(P)    174    7     28   38.6616  4:2:0   359
0088(P)    176    8     28   38.7062  4:2:0   360
0089(P)    178    9     28   38.4330  4:2:0   390
0090(P)    180   10     28   38.3659  4:2:0   344
0091(P)    182   11     28   38.3898  4:2:0   375
0092(P)    184   12     28   38.2694  4:2:0   391
0093(P)    186   13     28   38.1671  4:2:0   563
0094(P)    188   14     28   37.9238  4:2:0   391
0095(P)    190   15     28   38.4139  4:2:0   375
0096(P)    192    0     28   39.2595  4:2:0   344
0097(P)    194    1     28   39.2051  4:2:0   359
0098(P)    196    2     28   39.7860  4:2:0   328
0099(P)    198    3     28   -1.#IND  4:2:0   359
-----
Total decoding time : 37.869 sec
-----
Exit JM 11 (PRExt) decoder, ver 11.0
D:\Skola\MM2E\JM2_3\bin>

```

Obr. 11 Koniec dekódovacieho procesu

## 7 Další testy algoritmu

### 7.1 Charakteristika použitých videosekvencí

Pre ďalšie testy boli použité ďalšie 3 videosekvencie, ktoré majú rôzne vlastnosti v časovej a frekvenčnej oblasti. Boli na nich uskutočnené rôzne merania, ktoré zisťujú, na čom závisí presnosť odhadu PSNR a tiež rýchlosť dekódovacieho procesu, resp. ušetrený výpočetný výkon. Videosekvencie sú voľne šíriteľné a teda boli stiahnuté z internetu. Sú v nekomprimovanom formáte \*.yuv. Potom boli zakódované pomocou referenčného enkodéru s rôzne nastavenými parametrami podľa potreby daného testu a následne znova dekódované a vyhodnotené. Sú to tieto 3:



**Obr. 12** Snímok z videosekvencie „flower\_cif“.

#### „flower\_cif“

- stredne dynamická – pohyb kamery z ľava doprava
- v spodnej časti vysoká spektrálna aktivita, množstvo detailov
- v hornej časti nízka spektrálna aj časová aktivita
- veľký jas
- sýte farby
- dĺžka 50 snímkov
- ukážka na obr. 12



Obr. 13 Snímek z videosekvence „bridge\_close\_cif“.

#### „bridge\_close\_cif“

- velmi statická scéna – malý pohyb lidí na mostě a malý pohyb vody
- vysoká spektrální aktivita, množství detailů
- nízká časová aktivita
- velký jas
- středně syté barvy
- délka 50 snímků
- ukážka na obr. 13



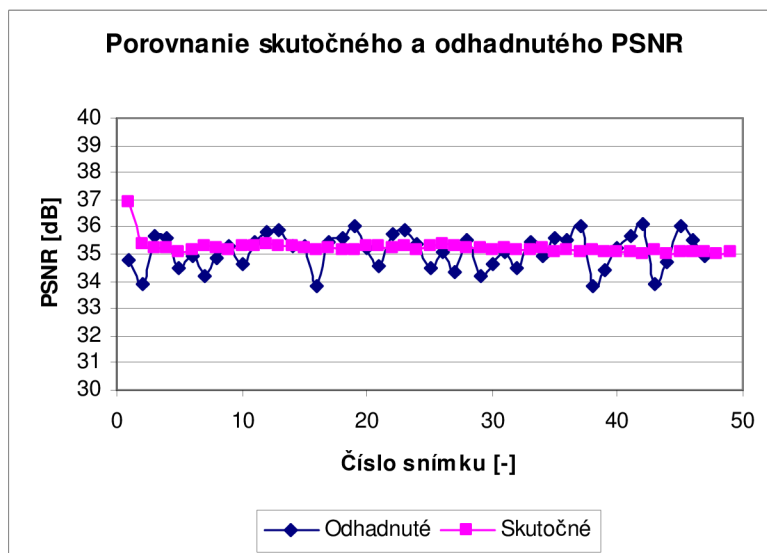
Obr. 14 Snímek z videosekvence „coastguard\_cif“.

#### „coastguard\_cif“

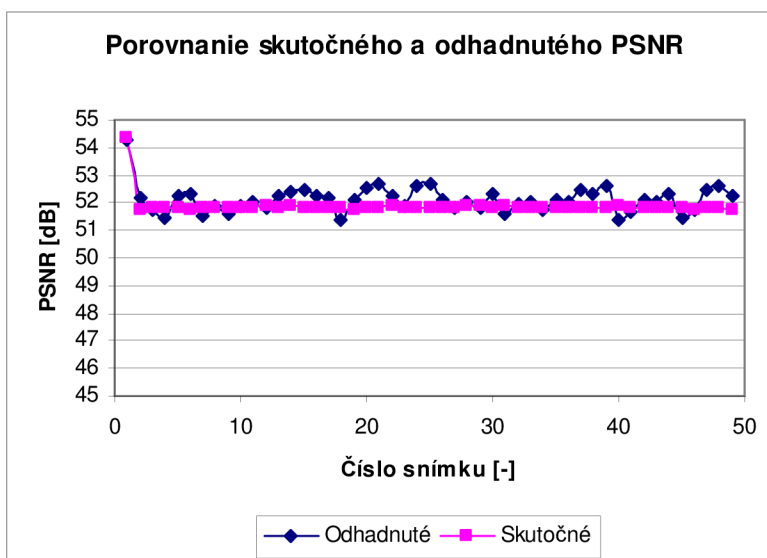
- středně dynamická scéna – pohyb oboch lodí, pohyb kamery
- vysoká spektrální aktivita
- slabší jas
- středně syté barvy
- délka 50 snímků
- ukážka na obr. 14

## 7.2 Vplyv kvality videosekvencie na presnosť odhadu PSNR

Na toto meranie bola použitá videosekvencia „flower\_cif“ a to v dvoch kvalitatívnych verziách. Pre prvú bol nastavený kvantizačný parameter na hodnotu 28, a z toho vyplýva výsledná priemerná hodnota skutočného PSNR 35,22 dB. Pre druhú bol kvantizačný parameter na hodnote 8 a priemerné skutočné PSNR je 51,87 dB. Na obr. 15 a 16 je vidieť rozdiely medzi skutočnými a odhadnutými hodnotami PSNR. Na obr. 17 a 18 je relatívna odchýlka zodpovedajúca zisteným hodnotám.

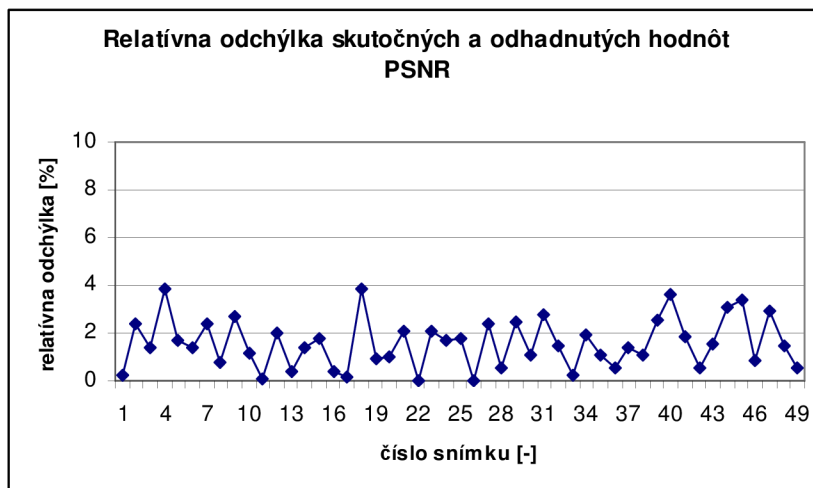


Obr. 15 Porovnanie skutočného a odhadnutého PSNR pre menej kvalitnú videosekvenciu

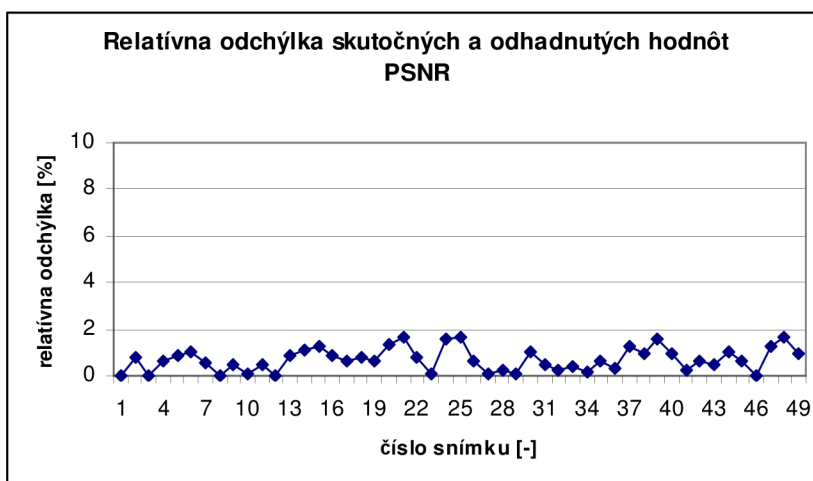


Obr. 16 Porovnanie skutočného a odhadnutého PSNR pre kvalitnejšiu videosekvenciu

Je tu pekne vidno, že pre kvalitnejšie videosekvencie je aj odhad hodnôt PSNR presnejší.



**Obr. 17** Retalívna odchýlka skutočných a odhadnutých hodnôt PSNR zodpovedajúca menej kvalitnej videosekvencii



**Obr. 18** Retalívna odchýlka skutočných a odhadnutých hodnôt PSNR zodpovedajúca kvalitnejšej videosekvencii

Relatívna odchýlka bola počítaná podľa:

$$\delta = \left| \frac{b-a}{b} \right| \cdot 100 [\%], \quad (7.1)$$

kde  $b$  je hodnota skutočného PSNR a  $a$  je hodnota odhadnutého PSNR.

Retalívna odchýlka skutočných a odhadnutých hodnôt PSNR (obr. 17 a 18) je teda nižšia pre kvalitnejšiu videosekvenciu. Dosahuje tu maximálnych hodnôt niečo pod 2 %, kdežto u menej kvalitnej videosekvencii je to až dvojnásobok (4 %), čo je ale ešte stále dobré číslo.

Bol tu takisto zmeraný čas dekodovania oboch videosekvencií a to jak v pôvodnom, tak v optimalizovanom dekodéry (tab. 6 a 7).



**Tab. 6:** Trvanie dekodovacieho procesu a ušetrený výpočetný výkon pre menej kvalitnú videosekvenciu

Poradové číslo merania	Čas trvania pôvodného dekodéru [s]	Čas trvania optimalizovaného dekodéru [s]	Ušetrený čas [s]	Ušetrený výpočetný výkon [%]
1.	19,594	15,511	4,083	25,59
2.	19,544	15,387	4,157	21,27
3.	19,419	15,434	3,985	20,52
4.	19,488	15,389	4,099	21,03
5.	19,497	15,219	4,278	21,94
<b>Priemer:</b>				22,07

**Tab. 7:** Trvanie dekodovacieho procesu a ušetrený výpočetný výkon pre kvalitnejšiu videosekvenciu

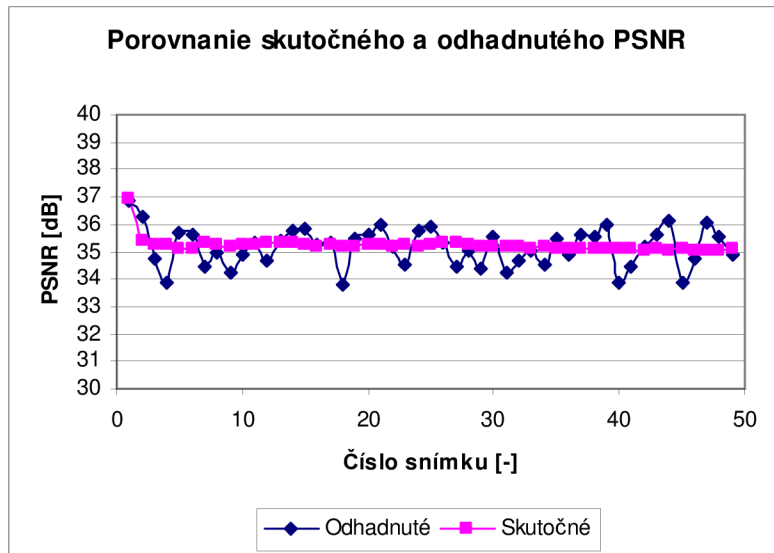
Poradové číslo merania	Čas trvania pôvodného dekodéru [s]	Čas trvania optimalizovaného dekodéru [s]	Ušetrený čas [s]	Ušetrený výpočetný výkon [%]
1.	76,842	71,031	5,811	7,56
2.	76,782	71,472	5,310	6,92
3.	76,065	71,421	4,644	6,11
4.	76,096	71,767	4,329	5,69
5.	75,457	71,112	4,345	5,76
<b>Priemer:</b>				6,41

U menej kvalitnej videosekvencii (priemerné PSNR 35,22 dB) bolo optimalizáciou dekodéru ušetrených cca 22,07 % výpočetného výkonu. U tej veľmi kvalitnej (priemerné PSNR 51,87 dB) to vyšlo len 6,41 %.

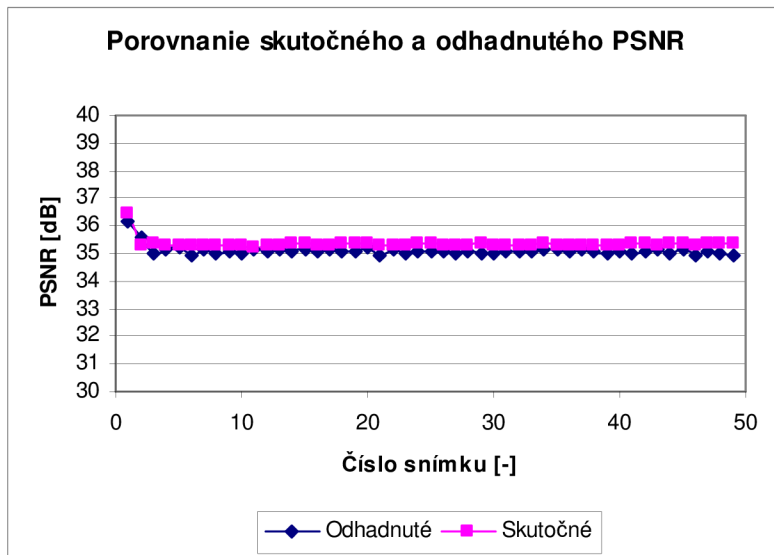
### 7.3 Vplyv obsahu videosekvencie na presnosť odhadu PSNR

Na toto meranie boli použité všetky 3 videosekvencie. Pre všetky bol nastavený rovnaký kvantizačný parameter 28 a aj priemerná hodnota PSNR vyšla veľmi podobne, u „flower\_cif“ 35,22 dB, u „bridge\_close\_cif“ 35,34 dB a u „coastguard\_cif“ to bolo 34,37 dB. To znamená, že sú to zhruba rovnako kvalitné videosekvencie s odlišným obsahom scény. Na obr. 19, 20 a 21. sú rozdiely medzi hodnotami PSNR a na obr. 22, 23 a 24 relatívne odchýlky.

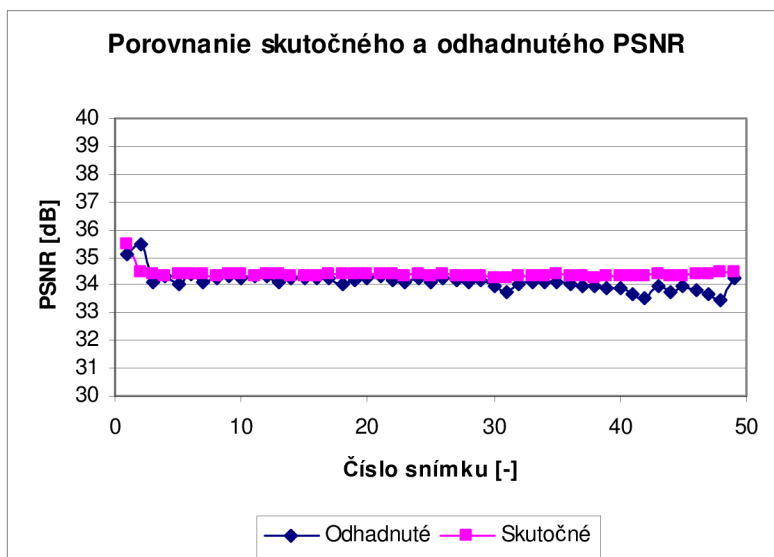




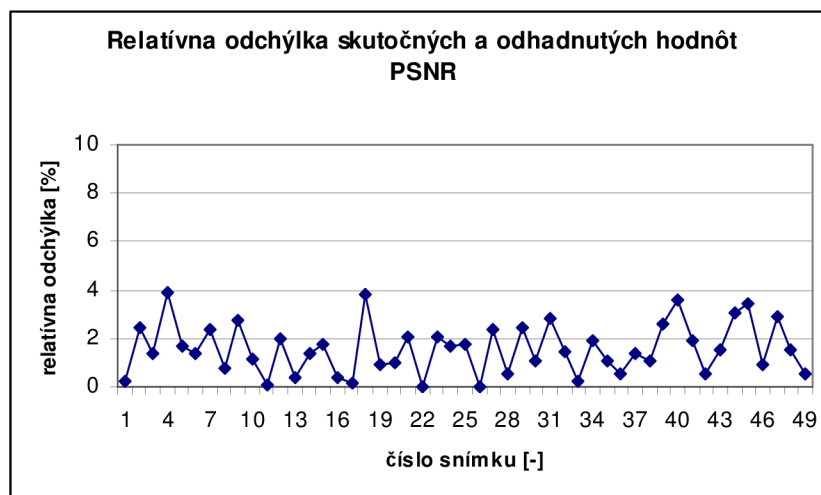
Obr. 19 Porovnanie skutočného a odhadnutého PSNR pre videosekvenciu „flower\_cif“



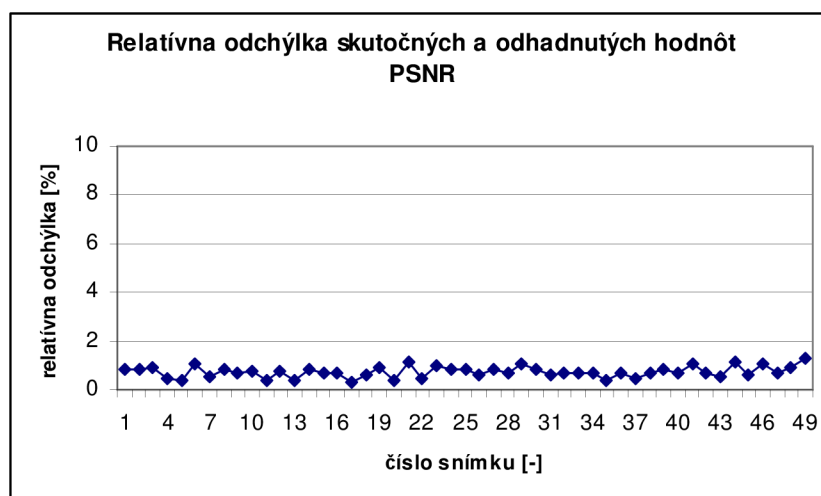
Obr. 20 Porovnanie skutočného a odhadnutého PSNR pre videosekvenciu „bridge\_close\_cif“



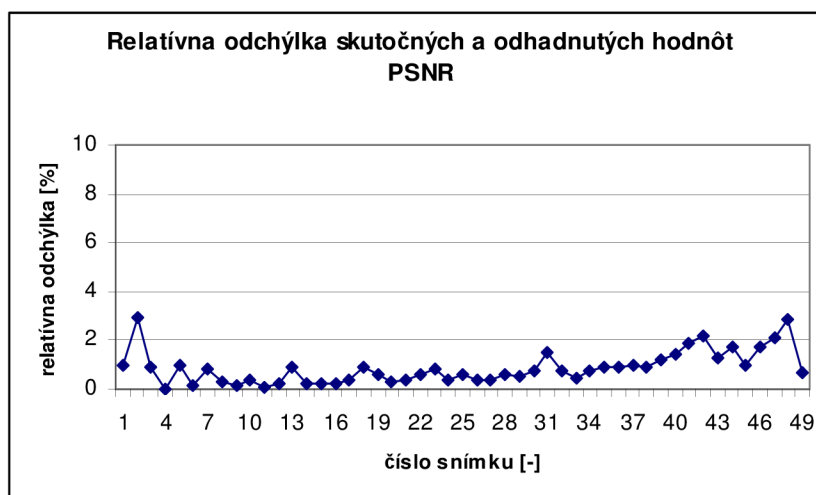
**Obr. 21** Porovnanie skutočného a odhadnutého PSNR pre videosekvenciu „coastguard\_cif“



**Obr. 22** Relatívna odchýlka skutočných a odhadnutých hodnôt PSNR pre videosekvenciu „flower\_cif“



**Obr. 23** Relatívna odchýlka skutočných a odhadnutých hodnôt PSNR pre videosekvenciu „bridge\_close\_cif“



**Obr. 24** Relatívna odchýlka skutočných a odhadnutých hodnôt PSNR pre videosekvenciu „coastguard\_cif“

V tomto meraní sa pekne ukázalo, že odhad PSNR je jednoznačne presnejší pre videosekvencie so statickejšou scénou. Relatívna odchýlka tu dosahovala maximálnych hodnôt okolo 1,22 %, čo je vynikajúce. Pri väčšej dynamike scény odhad už tak presný nebol, ale stále sa dá považovať za veľmi dobrý (relatívna odchýlka cca do 4 %).

Namerané časy dekodovacieho procesu a ušetrený výpočetný výkon sú v tab. 8, 9 a 10.

**Tab. 8:** Trvanie dekodovacieho procesu a ušetrený výpočetný výkon pre videosekvenciu „flower\_cif“

Poradové číslo merania	Čas trvania pôvodného dekodéru [s]	Čas trvania optimalizovaného dekodéru [s]	Ušetrený čas [s]	Ušetrený výpočetný výkon [%]
1.	28,421	23,783	4,638	16,32
2.	28,249	23,611	4,638	16,42
3.	28,559	23,894	4,665	16,33
4.	28,559	23,855	4,704	16,47
5.	28,503	23,857	4,646	16,30
<b>Priemer:</b>				16,37

**Tab. 9:** Trvanie dekodovacieho procesu a ušetrený výpočetný výkon pre videosekvenciu „bridge\_close\_cif“

Poradové číslo merania	Čas trvania pôvodného dekodéru [s]	Čas trvania optimalizovaného dekodéru [s]	Ušetrený čas [s]	Ušetrený výpočetný výkon [%]
1.	20,934	17,471	3,436	16,54
2.	20,968	17,340	3,628	17,30
3.	20,951	17,293	3,658	17,46
4.	20,952	17,252	3,700	17,66
5.	20,887	17,261	3,626	17,36
<b>Priemer:</b>				17,26

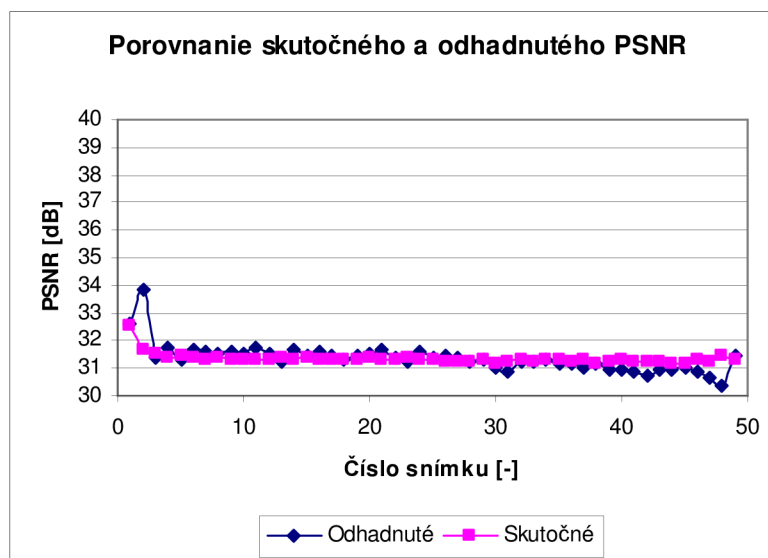
**Tab. 10:** Trvanie dekodovacieho procesu a ušetrený výpočetný výkon pre videosekvenciu „coastguard\_cif“

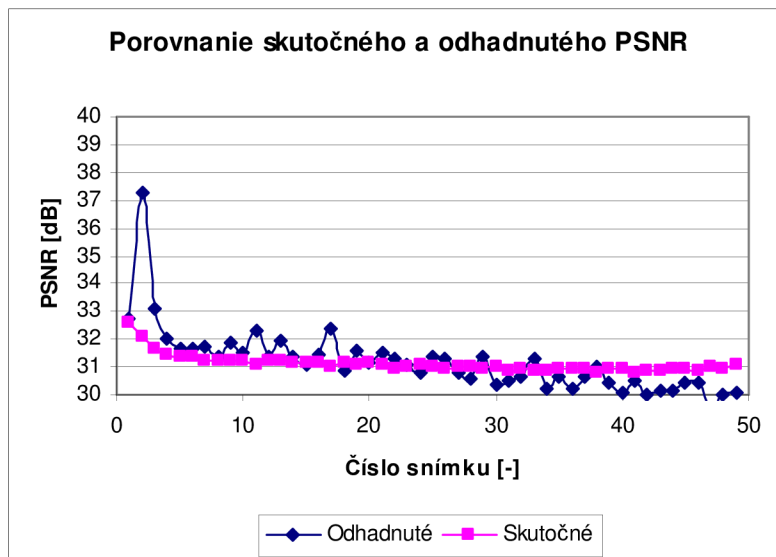
Poradové číslo merania	Čas trvania pôvodného dekodéru [s]	Čas trvania optimalizovaného dekodéru [s]	Ušetrený čas [s]	Ušetrený výpočetný výkon [%]
1.	29,441	23,875	5,566	18,91
2.	29,419	24,050	5,369	18,25
3.	29,466	23,942	5,524	18,75
4.	29,309	24,017	5,292	18,06
5.	29,371	24,064	5,307	18,07
<b>Priemer:</b>				18,41

Vo všetkých troch prípadoch bol ušetrený výpočetný výkon zhruba rovnaký (medzi 16 až 19 %).

#### 7.4 Vplyv rozlíšenia videosekvencie na presnosť odhadu PSNR

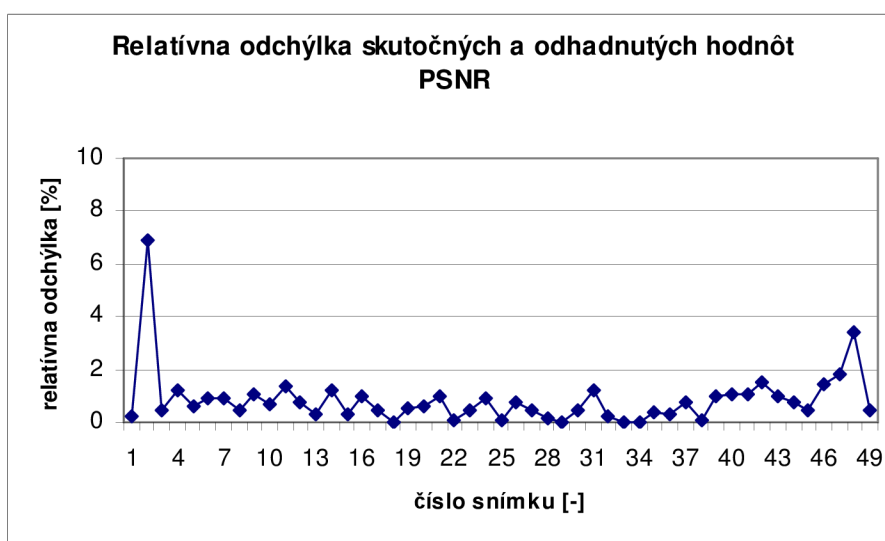
Na toto meranie bola použitá videosekvencia „coastguard\_cif“ v dvoch rozlíšeniach, a to 352 x 288 a 176 x 144. Kvantizačný parameter bol nastavený na hodnotu 32 a kvalita bola v okolí 31 dB u oboch videosekvencií. No a znovu bolo sledované, aké sú rozdiely medzi skutočnými a odhadnutými hodnotami PSNR, aké sú ich relatívne odchýlky a samozrejme trvanie času dekodovania pre obe rozlíšenia. Na obr. 25 a 26 je teda porovnanie skutočných a odhadnutých hodnôt PSNR a na obr. 27 a 28 sú ich relatívne odchýlky.

**Obr. 25** Porovnanie skutočného a odhadnutého PSNR pre videosekvenciu „coastguard\_cif“ s rozlíšením 352 x 288

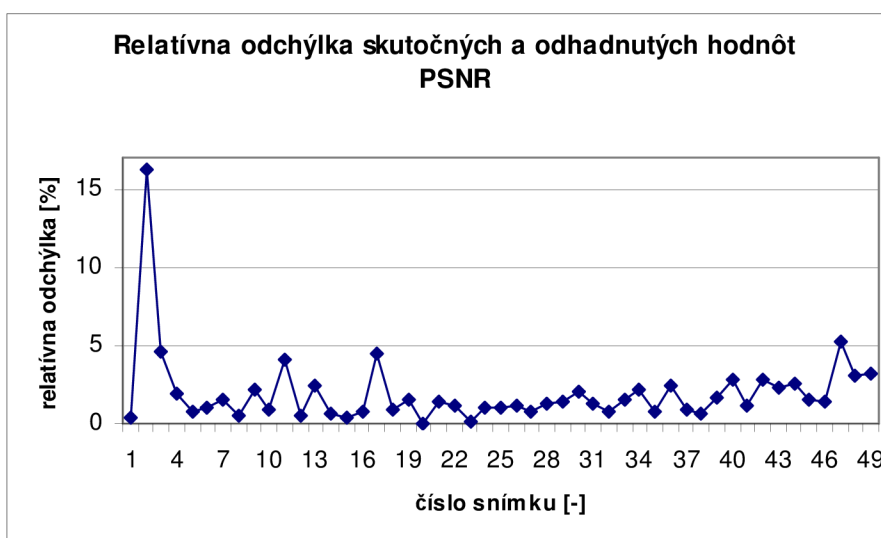


**Obr. 26** Porovnanie skutočného a odhadnutého PSNR pre videosekvenciu „coastguard\_cif“ s rozlíšením 176 x 144

Z porovnania hodnôt PSNR je vidieť, že pri väčšom rozlíšení je odhad jednoznačne lepší, a z toho samozrejme vyplývajú nižšie hodnoty relatívnej odchýlky pre túto videosekvenciu. Len jedna odhadnutá hodnota PSNR sa výrazne líši od tej skutočnej, kdežto u videosekvencie s rozlíšením 176 x 144 je to podstatne horšie. Najväčšia realívna odchýlka je až 16,21 %, čo už je veľmi nepresné.



**Obr. 27** Retalívna odchýlka skutočných a odhadnutých hodnôt PSNR pre videosekvenciu „coastguard\_cif“ s rozlíšením 352 x 288



**Obr. 28** Relatívna odchýlka skutočných a odhadnutých hodnôt PSNR pre videosekvenciu „coastguard\_cif“ s rozlíšením 176 x 144

V tab. 11 a 12 sú výsledky meraní trvania dekodovacieho procesu pre oba referenčné dekodéry a znova je vyčíslené aj ušetrenie výpočetného výkonu pre videosekvenciu s vyšším aj nižším rozlíšením.

**Tab. 11:** Trvanie dekodovacieho procesu a ušetrený výpočetný výkon pre videosekvenciu „coastguard\_cif“ s rozlíšením 352 x 288

Poradové číslo merania	Čas trvania pôvodného dekodéru [s]	Čas trvania optimalizovaného dekodéru [s]	Ušetrený čas [s]	Ušetrený výpočetný výkon [%]
1.	23,987	18,783	5,204	21,70
2.	23,703	18,783	4,920	20,76
3.	23,938	18,799	5,139	21,45
4.	23,831	18,779	5,052	21,20
5.	23,860	19,126	4,734	19,84
<b>Priemer:</b>				20,99

**Tab. 12:** Trvanie dekodovacieho procesu a ušetrený výpočetný výkon pre videosekvenciu „coastguard\_cif“ s rozlíšením 176 x 144

Poradové číslo merania	Čas trvania pôvodného dekodéru [s]	Čas trvania optimalizovaného dekodéru [s]	Ušetrený čas [s]	Ušetrený výpočetný výkon [%]
1.	13,138	12,096	1,042	7,93
2.	12,980	11,958	1,022	7,87
3.	13,012	12,062	0,950	7,30
4.	12,988	12,063	0,925	7,12
5.	13,107	11,975	1,132	8,64
<b>Priemer:</b>				7,77

Pre videosekvenciu s rozlíšením 352 x 288 vyšlo ušetrenie výpočetného výkonu na 20,99 %, čo je podobné ako v predchádzajúcich meraniach. U rovnakej videosekvencii ale

s polovičným rozlišením v oboch smeroch (176 x 144) už vyšlo oveľa menšie číslo, a to 7,77 %.

## Záver

V tejto práci bol predstavený štandard H.264/MPEG-4 AVC, jeho štruktúra, jeho neoddeliteľné súčasti a princíp funkcie enkodéru a dekodéru.

Ďalej tu bol predstavený a implementovaný algoritmus pre vyhodnotenie kvality videosekvencií na základe údajov získaných zo skreslenej videosekvencie. To znamená, že nie je potrebná referencia. Algoritmus vyhodnocuje objektívnu metriku špičkový pomer signálu k šumu PSNR. Matematická zložitosť je veľmi malá, a preto je možná implementácia v reálnom čase. Pri procese dekódovania nie je nutné, aby bola vytvorená dekódovaná videosekvencia, a preto je zložitosť dekodéru menšia ako u bežných dekodérov. Boli tu uskutočnené testy s videosekvenciou „foreman.yuv“, na základe ktorých sa dá tvrdiť, že tento algoritmus pre odhad hodnôt PSNR je dostatočne presný. Korelácia výsledkov skutočných a odhadnutých hodnôt PSNR vyšla 96,20 %.

Bol tu predstavený referenčný softvér, ktorý obsahuje enkodér aj dekodér štandardu H.264/AVC. Algoritmus bol implementovaný do referenčného dekodéru, ktorý bol následne optimalizovaný tak, aby jeho jedinou funkcionalitou bol výpočet odhadnutých hodnôt PSNR. Na záver tu boli uskutočnené rozsiahlejšie testy algoritmu s ďalšími tromi videosekvenciami, pri ktorých bola analyzovaná závislosť presnosti odhadu PSNR na rôznych vlastnostiach videosekvencie, ako napríklad: rozlíšenie, obsah, či kvalita. Okrem toho bol meraný aj čas dekódovacieho procesu pôvodného dekodéru a optimalizovaného dekodéru. Z toho bol následne vypočítaný ušetrený výpočetný výkon. Na základe týchto meraní je možné povedať, že presnosť odhadu PSNR závisí na kvalite videosekvencie. Čím je videosekvencia kvalitnejšia, tým je odhad presnejší. Relatívne odchýlky odhadnutých hodnôt PSNR od tých skutočných sa pri veľmi kvalitnej videosekvencii (PSNR okolo 52 dB) pohybovali do 2 %, čo je vynikajúci výsledok. Pri tých stredne kvalitných (PSNR okolo 35 dB) bola relatívna odchýlka do 4 %, čo je ešte stále priateľné číslo. Čo sa týka obsahu videosekvencie, tak ako najvhodnejšie sa pre tento algoritmus ukázali statické scény, kde nie je veľa pohybu (PSNR do 1,22 %). Naopak, čím dynamickejšia scéna, tým menej presný odhad PSNR. No a napokon bola testovaná zmena rozlíšenia rovnakej videosekvencie. Všetky doteraz analyzované videosekvencie mali rozlíšenie 352 x 288 bodov. Bola ale vyskúšaná videosekvencia s polovičným rozlíšením v oboch smeroch (176 x 144 bodov) a výsledok bol taký, že odhad PSNR sa podstatne zhoršil, relatívna odchýlka sa vyšplhala až do 16,21 %, čo už je veľmi nepresné.

Čo sa týka ušetreného výpočetného výkonu po optimalizácii dekodéru, tak u väčšiny testov vychádza ušetrenie v okolí 20 % (plus mínus 3 %). Výrazne iné číslo vychádza len v prípade veľmi kvalitnej videosekvencie (PSNR okolo 52 dB), tam to vyšlo len cca 6,41 % a v prípade videosekvencie s rozlíšením 176 x 144, kde vyšiel ušetrený výpočetný výkon na úrovni cca 7,77 %.



## Použitá literatura

- [1] EDEN, A. No-Reference Estimation of the Coding PSNR for H.264-Coded Sequences. *IEEE Transactions on Consumer Electronics*, 2007, vol. 53, no. 2, p. 667-674.
- [2] SÜHRING, K. *The T.264/MPEG-4 AVC Reference Software - JM 15* [online] [cit. 19. prosince 2008]. Dostupné na www: <http://iphome.hhi.de/suehring/tml/download/>
- [3] RICHARDSON, E.G. *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*. John Wiley & Sons, 2003. ISBN: 0-470-84837-5.
- [4] Prispievatelia wikipédie. Medzinárodná telekomunikačná únia [online]. 13.12.2009 [cit.29.12.2009]. Dostupné na www: [http://sk.wikipedia.org/wiki/Medzin%C3%A1rodn%C3%A1\\_telekomunika%C4%8Dn%C3%A1\\_%C3%BAnia](http://sk.wikipedia.org/wiki/Medzin%C3%A1rodn%C3%A1_telekomunika%C4%8Dn%C3%A1_%C3%BAnia)
- [5] Slanina, M. *Komprimace digitálních obrazových signálů, JPEG, MPEG-1, MPEG-2: přednáška z predmetu MVDK*. Brno: VUT Fakulta elektrotechniky a komunikačních technologií, 2009. 43 s.
- [6] Kratochvíl, T. *Měření kvality obrazu v transportním toku MPEG-2 TS při příjmu DVB-T: návod na laboratorní úlohu z predmetu MDTV*. Brno: VUT Fakulta elektrotechniky a komunikačních technologií, 2010. 6 s.