

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

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

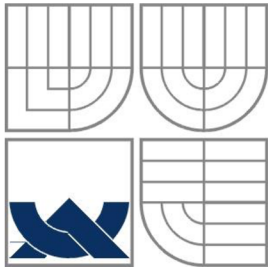
**AKCELERACE KÓDOVÁNÍ VIDEO DO FORMÁTU**  
**H.264 S VYUŽITÍM CUDA**

**BAKALÁŘSKÁ PRÁCE**  
BACHELOR'S THESIS

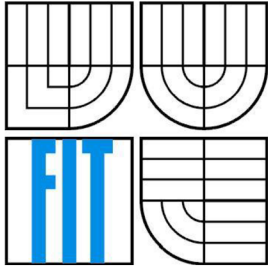
**AUTOR PRÁCE**  
AUTHOR

**CUONG DUONG TUAN**

BRNO 2015



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



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

# **AKCELERACE KÓDOVÁNÍ VIDEO DO FORMÁTU H.264 S VYUŽITÍM CUDA**

CUDA ACCELERATED VIDEO ENCODING TO H.264 FORMAT

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**CUONG DUONG TUAN**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. DRAHOSLAV ZÁŇ**

BRNO 2015

## **ABSTRAKT**

Cílem této bakalářské práce je vytvořit funkční enkodér formátu H.264 akcelerovaného CUDA platformou. H.264 je aktuálně jeden z nejpoužívanějších standardů pro kompresi digitálního videa, jelikož je ale výpočetně náročný, je vhodné ho zpracovat na GPU. Enkodér je psaný v C/C++ a využívá NVCUVID a NVENC API pro dekódování/enkódování videa.

## **KLÍČOVÁ SLOVA**

CUDA, H.264, C, C++, enkodér, NVCUVID, NVENC

## **ABSTRACT**

The goal of this bachelor's thesis is to create functional encoder of H.264 format accelerated by CUDA platform. H.264 is at the moment one of the most popular digital compression standards, but because of its high computational demands, it is suitable to process it on GPU. Encoder is written in C/C++ and uses NVCUVID and NVENC API.

## **KEYWORDS**

CUDA, H.264, C, C++, encoder, NVCUVID, NVENC

DUONG TUAN, Cuong *Akcelerace kódování videa do formátu H.264 s využitím CUDA*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta informačních technologií, Ústav počítačových systémů, 2015. 34 s. Vedoucí práce byl Ing. Drahoslav Záh

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Akcelerace kódování videa do formátu H.264 s využitím CUDA“ jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské 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 autorského 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.

Brno .....

.....

(podpis autora)



## PODĚKOVÁNÍ

Rád bych poděkoval svému vedoucímu Ing. Drahoslavovi Záhovi za cenné informace při vytváření této práce.

Brno .....

.....

(podpis autora)

# OBSAH

Úvod	3
<b>1 Digitální video</b>	<b>4</b>
1.1 Analogové vs digitální video	4
1.2 Video kodek	4
1.3 Ukládání a kontejnery	4
1.4 Barevný prostor	5
1.5 YCbCr	6
1.6 Chroma podvzorkování	7
1.7 Prokládání a snímková frekvence	7
1.8 Rozlišení	7
1.9 Datový tok	8
1.10 Optická média	8
1.10.1 DVD	8
1.10.2 Média pro vysoké rozlišení	9
<b>2 Komprese</b>	<b>10</b>
2.1 Komprese videa	10
2.2 Intra-frame kódování	10
2.3 Inter-frame kódování	11
<b>3 H.264</b>	<b>12</b>
3.1 Historie	12
3.2 Profily a úrovně	12
3.3 Entropické kódování	13
3.3.1 CABAC	14
3.3.2 CAVLC	14
3.4 Makroblok	15
3.5 Enkódovací proces	16
3.5.1 Predikce	16
3.5.2 Transformace a kvantizace	17
3.5.3 Bitstream enkódování	17
3.6 Dekódovací proces	18
3.7 Budoucnost	18
<b>4 GPGPU</b>	<b>19</b>
4.1 CUDA	20
4.2 NVCUVID	20
4.3 NVENC	20
<b>5 Implementace enkodéru</b>	<b>22</b>
5.1 Programovací jazyk a nástroje	22
5.2 Struktura kódu	22
5.2.1 Dekódování	22
5.2.2 Enkódování	23
5.2.3 GUI	24
5.2.4 Hlavní smyčka	24

<b>6 Testování</b>	<b>26</b>
6.1 Testovací sestava	26
6.2 Testovací videa	26
6.3 Alternativní enkodéry	26
6.3.1 x264	26
6.3.2 MainConcept H.264	27
6.4 Testovací metodika	27
6.5 Výsledky měření	27
6.6 Porovnání výkonu	29
<b>7 Závěr</b>	<b>30</b>
<b>Literatura</b>	<b>31</b>
<b>Seznam příloh</b>	<b>33</b>
<b>A GUI programu</b>	<b>34</b>
<b>B Obsah přiloženého CD</b>	<b>34</b>

# ÚVOD

Prakticky veškeré dnes produkované video je v digitálním formátu. S ústupem tradičních analogových technologií pro zápis videa jako např. VHS<sup>1</sup> nebo filmové pásy nastoupila technologie digitálního videa.

S digitálním videem se aktuálně setkáme téměř kdekoliv - ať už sledujeme pozemní, kabelové nebo satelitní televizní vysílání, online videa na službách jako Youtube nebo si prohlédneme záznam z kamery našeho mobilního telefonu.

Digitální video může být uloženo mnoha způsoby - na disku, online službách jako již zmínovaný Youtube nebo Facebook, na fyzickém médiu jako je DVD či Blu-ray. V téměř všech případech je video kompresované z důvodu úspory místa.

Kompresce je téma související s digitálním videem od jeho raného počátku. Nekomprimované video má velmi vysoký datový tok a s ním související vysoké nároky na úložný prostor. Proto je snaha o co největší úsporu dat při nejmenší ztrátě obrazové kvality.

Mezi nejpopulárnější standardy pro kompresi videa patří H.264. Patří k těm formátům, které využívají vysokého výkonu dnešních počítačů pro co nejdokonalejší poměr dat/obrazové kvality. Nicméně čím lepší komprese, tím vyšší je potřebný výkon a mnohdy dnešní procesory nestíhají zpracovat video v dostatečné rychlosti.

Díky rozmachu GPGPU<sup>2</sup> je dnes možné naprogramovat grafické karty tak, aby byly výpočty prováděné na nich místo na CPU. GPU mají navrch oproti CPU v úlohách, které jsou snadno paralelizovatelné. Zpracování videa patří mezi tyto úlohy.

---

<sup>1</sup>Video Home System

<sup>2</sup>General-purpose computing on graphics processing units

# 1 DIGITÁLNÍ VIDEO

## 1.1 Analogové vs digitální video

Digitální video má oproti analogovému řadu výhod[19]:

- Digitální formát jako DVD umožňuje tvorbu bezztrátových kopií - Dvě různé kopie DVD jsou bitově identické, oproti například VHS páskám, kde se při kopírování ztrácí data
- Jednoduchost přenosu dat – jde využít libovolný digitální přenos jako je internet, fyzické médium apod., s tím souvisí možnost použití metod pro ochranu dat jako jsou korekční kódy a tím celkově menší chybovost
- Jednoduchost úpravy videa – Analogové video je velmi náročné na techniku, která je obvykle drahá a nedostupná, oproti digitálnímu videu, které lze upravovat na téměř libovolném stolním počítači nebo i na mobilním telefonu

Analogové video se v poslední době používá už jenom kvůli zpětné kompatibilitě, například se staršími televizní vysílači. V České Republice bylo analogové televizní vysílání definitivně ukončeno dne 30.11.2011 [3].

## 1.2 Video kodek

Video kodek<sup>1</sup> je program či zařízení, které kóduje a dekóduje video do nebo z určitého formátu, obvykle kvůli úspoře dat. Takové video se v praxi běžně ukládá do tzv. multimediálních kontejnerů.

Je třeba rozlišovat pojmy video kodek a video formát. Formát je definovaný standard, který kodek implementuje. Například kodek **x264** (v užším smyslu enkodér) implementuje formát H.264.

Mezi velmi populární kodeky patří open-source kodek **FFmpeg**, který implementuje řadu formátů, od starších jako je H.263/MPEG-4 Part 2 až po nejnovější H.265/MPEG-H HEVC. Podporuje taky řadu jiných kodeků v rámci plug-inů jako například zmiňovaný x264.

## 1.3 Ukládání a kontejnery

Digitální video se může ukládat do tzv. multimediálního kontejneru, které typicky obsahuje jednu videostopu a jednu či více audiostop, popř. titulky s tím, že je zaručena synchronizace. Kontejnery se vzájemně liší podle toho, jaké data dokážou pojmout, obvykle jsou zde limity na rozlišení, snímkovou frekvenci apod. a neposledně také se liší tím, které video/audio formáty podporuje.

Mezi značně populární kontejnery patří Matroska (soubory .mkv), který je velmi populární u warezového<sup>2</sup> obsahu obzvláště díky velmi široké podpoře různých formátů.

---

<sup>1</sup>Vzniklo spojením slov **kó**dování a **dekó**dování

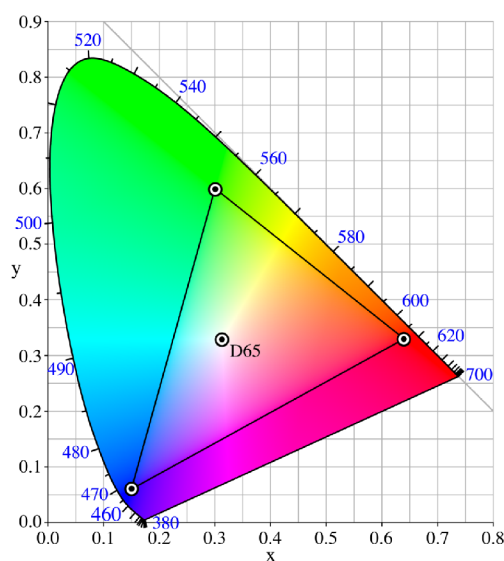
<sup>2</sup>Warez - Obecně autorské díla, se kterými se nakládá nelegálně, např. filmy

Tab. 1.1: Srovnání populárních kontejnerů.

Název	Firma	Video formáty	Audio formáty
AVI	Microsoft	Vše podporující Video For Windows	Vše podporující Audio Compression Manager
Matroska	CoreCodec	Cokoliv	Cokoliv
MP4	MPEG	MPEG-2, H.263, H.264, VC-1	MP3, AC-3, Apple lossless, Vorbis
QuickTime	Apple	Limitované QuickTime	Limitované CoreAudio

## 1.4 Barevný prostor

Barevný prostor je obecně množina barev sloužící k reprezentaci barev založené na nějakém barevném modelu jako jsou například **RGB**<sup>3</sup> nebo **CMYK**<sup>4</sup>. Rozsahem barev v barevném prostoru se definuje **gamut**. Obecně mají modely založené na RGB větší gamut než CMYK, RGB ale nelze použít u tiskáren kvůli aditivnímu míchání barev, proto se před tiskem převádí RGB na CMYK. Mezi nepoužívanější barevné prostory patří například sRGB nebo Adobe RGB.



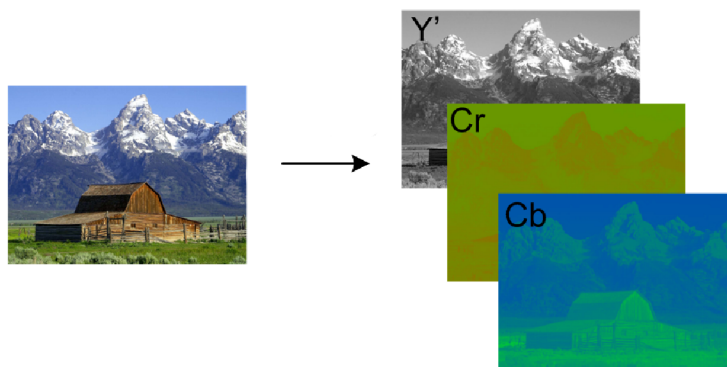
Obr. 1.1: Barevný prostor sRGB [7]. Trojúhelník označuje gamut sRGB v chromatickém diagramu CIE 1931. D65 označuje bílý bod.

<sup>3</sup>Red, Green, Blue - aditivní míchání

<sup>4</sup>Cyan, Magenta, Yellow, Black - substraktivní míchání

## 1.5 YCbCr

$Y C_B C_R$  je digitální variantou barevného modelu  $YUV$ , který vznikl pro potřeby analogové televize, jelikož se ale s nimi pracuje podobně, jsou často zaměňovány. Spojuje je především to, že mají jasovou složku  $Y$  a dva barevné posuny  $C_B$  a  $C_R$ .



Obr. 1.2: Repräsentace obrázku v YCbCr.  $Y'$  je jasová (luma) složka,  $Cr$  a  $Cb$  barevná, tzv. chroma složka. [5]

YCbCr není absolutní barevný prostor<sup>5</sup> jako např. RGB, je fakticky způsobem, jak jinak zapsat RGB informaci. Proto barva v YCbCr závisí na interpretaci zobrazovacího média. Problémem je taky to, že převod do YCbCr z RGB je ztrátový, tzn. že při konverzi mezi barevnými prostory je ztracena část informací o barvě. Je to způsobeno zaokrouhlováním při konverzi.

Převod z RGB do YCbCr je dán rovnicí:

$$\begin{aligned} Y &= 16 + \frac{65.738 \cdot R}{256} + \frac{129.057 \cdot G}{256} + \frac{25.064 \cdot B}{256} \\ C_B &= 128 - \frac{37.945 \cdot R}{256} - \frac{74.494 \cdot G}{256} + \frac{112.439 \cdot B}{256} \\ C_R &= 128 + \frac{112.439 \cdot R}{256} - \frac{94.154 \cdot G}{256} - \frac{18.285 \cdot B}{256} \end{aligned} \quad (1.1)$$

Daná rovnice platí pro převod 8-bitové RGB informace (tzn. že hodnota jednotlivých složek nabývá rozsahu od 0 do 256) do YCbCr s chroma podvzorkováním 4:4:4. Nicméně se v praxi 4:4:4 podvzorkování příliš nepoužívá.

Rovnice pro zpětný převod z YCbCr do RGB je následující:

$$\begin{aligned} R &= \frac{298.082 \cdot Y}{256} + \frac{408.583 \cdot C_R}{256} - 222.921 \\ G &= \frac{298.082 \cdot Y}{256} - \frac{100.291 \cdot C_B}{256} - \frac{208.120 \cdot C_R}{256} + 135.576 \\ B &= \frac{298.082 \cdot Y}{256} + \frac{516.412 \cdot C_B}{256} - 276.836 \end{aligned} \quad (1.2)$$

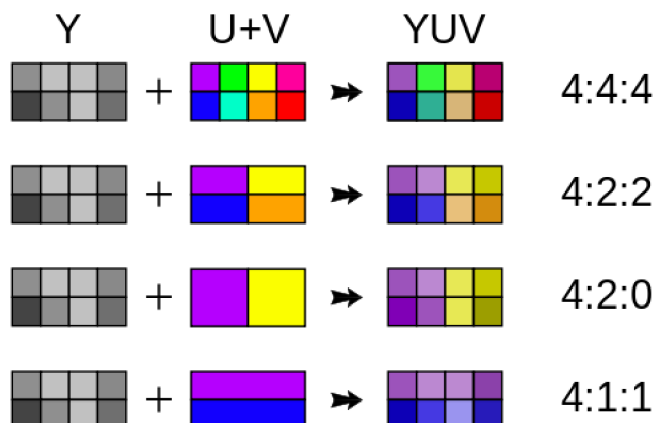
<sup>5</sup>Absolutní barevný prostor má přesně definovanou chromatičnost barevných složek, u YUV a YCbCr je dána posunem

## 1.6 Chroma podvzorkování

**Chroma podvzorkování** je forma komprese obrazu již při práci s barevným prostorem. Využívá se faktu, že lidské oko je více citlivé na jas než na barvu, tudíž se dá částečně redukovat barevná složka YCbCr obrazu bez pozorovatelné ztráty kvality. Tato technika se využívá mimo komprese videa i u komprese JPEG a JPEG 2000.

Druh chroma podvzorkování se obvykle definuje jako poměr  $J:a:b$ :

- J – Horizontální rozlišení vzorkování (šířka bloku), obvykle 4
- a – Počet chroma vzorků v horní řadě pixelů J
- b – Počet chroma vzorků ve spodní řadě pixelů J



Obr. 1.3: Běžné poměry chroma podvzorkování. 4:4:4 vzorkování je maximální, 4:2:0 je běžně používáno ve většině video formátů. [6]

## 1.7 Prokládání a snímková frekvence

Videa se dělí podle typu snímků na 2 typy:

- **Prokládané** (interlaced) – Každý snímek je rozdělen na dva "půlsnímky", kdy se v jednom okamžiku nepřenáší celý obraz, pouze liché nebo sudé řádky. Fakticky se jedná o kompresi obrazu, jelikož se takto dá použít dvojnásobná snímková frekvence při zachování stejné velikosti videa, bohužel ale takto vzniká řada artefaktů především při rychlém pohybu.
- **Progresivní** – Přenáší se pouze jeden snímek naráz, je takto uložena většina digitálního videa.

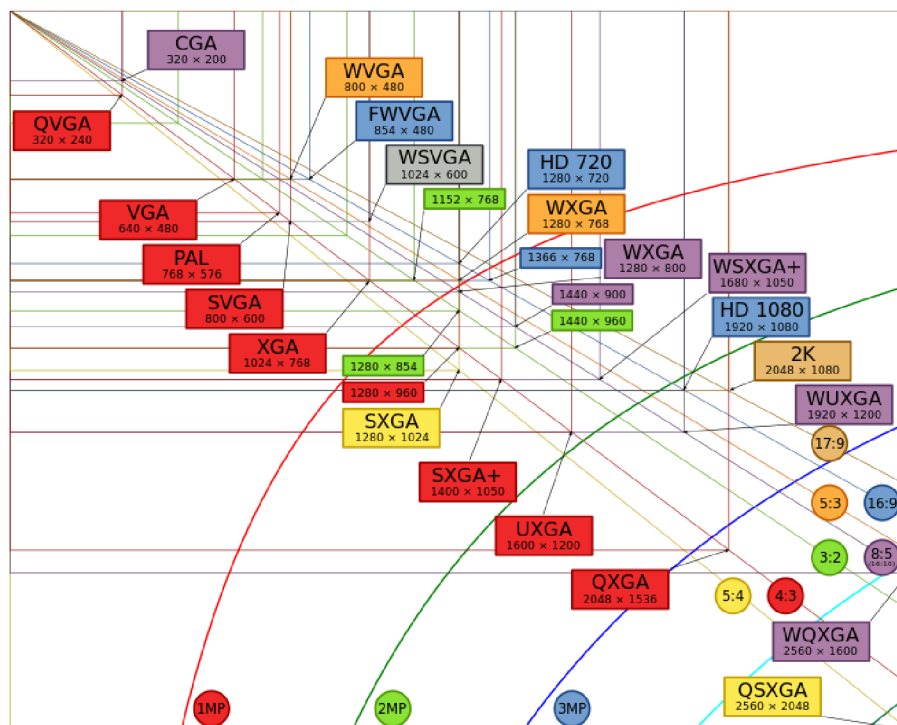
Snímková frekvence definuje počet snímků za vteřinu, může být proměnlivá v různých částech videa. Udává se v FPS (frames per second).

## 1.8 Rozlišení

Definuje šířku a výšku zobrazovacího média. Jisté kombinace šířky a výšky jsou standardizovány a pojmenovány. Poměr mezi šířkou a výškou udává poměr stran (aspect ratio). Obvykle je zadáváno jako **šířka x výška** v pixelech.

Mezi nejčastější rozlišení u současných zařízení patří 1280x720(HD), 1366x768 (HD Ready) a 1920x1080 (Full HD).





Obr. 1.4: Srovnání různých rozlišení [9]. Různé poměry stran jsou rozlišeny barvou – 4:3 je označené červeně, 16:9 modře apod.

## 1.9 Datový tok

Datový tok představuje množství dat přenesených za sekundu, definuje se obvykle v kbps nebo Mbps (kilobits/Megabits per second). U videa se dá dělit na 2 typy:

- **Constant Bit Rate (CBR)** – Datový tok je stejný ve všech částech videa, obecně je méně náročné na výpočet, ale více náročné na data při zachování stejné kvality obrazu oproti VBR. Standardně se video enkóduje jednorůchodově.
- **Variable Bit Rate (VBR)** – Datový tok se různí podle náročnosti snímků, na rozdíl od CBR bývá video enkóováno ve 2 nebo více průchodech - při prvním průchodu se určí části, které lze komprimovat lépe a enkóduje se až při dalším průchodu. Je dána snaha o nejlepší kvalitu videa při co nejnižším možném datovém toku.

## 1.10 Optická média

Optická média jsou fyzická média schopné záznamu na optických discích. Mezi optické disky první generace patří např. CD nebo LaserDisc, ale k masivnímu využití pro digitální video došlo až při nástupu DVD.

### 1.10.1 DVD

DVD vstoupilo v roce 1996 na trh jako nástupce CD, které bylo dimenzované především na audio – zejména kvůli kapacitě média. 700 MB tehdy nebylo dostatečné pro video distribuci, proto bylo vyvinuto DVD a s ním kompresní formát MPEG-2. Má s CD totožné

rozměry disku z důvodu zachování kompatibility přehrávačů (takřka všechny DVD přehrávače podporují i CD). DVD má kapacitu 4.7 GB v případě jednostranného média nebo 8.5 GB v případě oboustranného média. Jeho hlavním kompresním standardem formátem je MPEG-2<sup>6</sup>. Podporuje vysoké rozlišení až do 1920x1080<sup>7</sup>, 30 fps s chroma podvzorkováním 4:2:2.

DVD médium není stavěno na tak vysoké rozlišení z důvodu vysokého datového toku, tudíž se podle standardu DVD-Video používá nižší rozlišení, závislé na standardu PAL/NTSC<sup>8</sup>. Při frekvenci 50 Hz (PAL) se používá rozlišení 720x576 a při frekvenci 60 Hz (NTSC) 720x480.

### 1.10.2 Média pro vysoké rozlišení

S pomalým nástupem obsahu ve vysokém rozlišení vzniká i potřeba nového média a nového kompresního formátu, jelikož MPEG-2 je prostorově velmi náročný. Jako dvě konkurenční média se objevila **HD DVD** a **Blu-ray**. Obě média podporují jak starší formát MPEG-2, tak novější VC-1 a MPEG-4 AVC, neboli H.264. Obě média jsou také rozměrově identické s DVD ze stejného důvodu, jako je DVD rozměrově identické s CD – zachování kompatibility.

Obě média podporují videozáznam ve Full HD při 30 snímcích za sekundu progresivně a až 60 snímků za sekundu prokládaně. Blu-ray podporuje vyšší datový tok díky vyšší kapacitě (25 GB u jednovrstvého média oproti 15 GB u HD DVD).

HD DVD bylo vyvinuto firmami Toshiba, NEC a Sanyo, Blu-ray firmami Sony a Philips. Blu-ray a HD DVD spolu soupeřili v tzv. formátové válce, kde se bojovalo o majoritní podíl na trhu. Blu-ray disk měl převahu zejména díky herní konzoli Sony Playstation 3<sup>9</sup>. Na začátku roku 2008 bylo firmou Toshiba oznámen konec výroby HD DVD a tím se fakticky ustanovil Blu-ray jako jediný standard nové generace optických médií.

---

<sup>6</sup>Jinak se také označuje jako H.262

<sup>7</sup>Full HD

<sup>8</sup>PAL/NTSC jsou standardem kódování analogového videa. NTSC je používán především v Severní Americe, Japonsku a Jižní Koreji, PAL je typicky využíván v Evropě, Asii a Austrálii. Používá se ještě PAL kompatibilní SECAM ve Francii a státech bývalého Sovětského Svazu.

<sup>9</sup>Playstation 3 byl nějakou dobu po svém uvedení nejlevnější (a jeden z nejlepších) Blu-ray přehrávačů na trhu z důvodu dotace Sony.

## 2 KOMPRESSE

Kompresse (jinak také komprimace) videa znamená podobně jako u komprese tradičních dat zmenšení jejich dat při zachování informací. Kompresse se provádí kompresním algoritmem, kterých existuje několik typů podle zaměření – existují jiné kompresní algoritmy pro text, hudbu, video apod.

Kompresse dat se obecně dělí na 2 typy:

- **Bezeztrátová** – Při kompresi nedochází ke ztrátě dat, která by znemožňovala zpětnou rekonstrukci komprimovaných dat. Používá se velmi často při ukládání souborů. Typickým příkladem je formát ZIP.
- **Ztrátová** – Při kompresi jsou některá data odstraněna a nelze získat všechny původní informace. Bývá efektivnější než bezztrátová komprese. Typickým příkladem je hudební formát MP3.

### 2.1 Kompresse videa

Nekomprimované video má standardně velmi vysoký datový tok:

Tab. 2.1: Datový tok nekomprimovaného videa [19]. Bitová hloubka je 12 bitů/barva, Chroma podvzorkování je 4:4:4.

Formát	Rozlišení	Snímková frekvence	Datový tok
QCIF	176 x 144	15 fps	4 562 kbps
CIF	352 x 240	30 fps	30 413 kbps
ITU-R 601	720 x 480	30 fps	124 416 kbps
HD	1280 x 720	30 fps	331 776 kbps
Full HD	1920 x 1080	30 fps	746 496 kbps

Kvůli úspoře místa existuje řada metod obrazové komprese. Nicméně je nutné podotknout, že ztrátová komprese není žádaná u případů, kde je nutné mít veškeré původní data pro co nejkvalitnější práci - typicky při produkci filmových snímků (stříhání, postprocess efekty apod.). Komprimuje se obvykle až při distribuci - na internet, digitální televizi či fyzická média.

V zásadě se dá komprese videa dělit na 2 typy

- Intra-frame – V rámci snímku
- Inter-frame – Mezi snímky

### 2.2 Intra-frame kódování

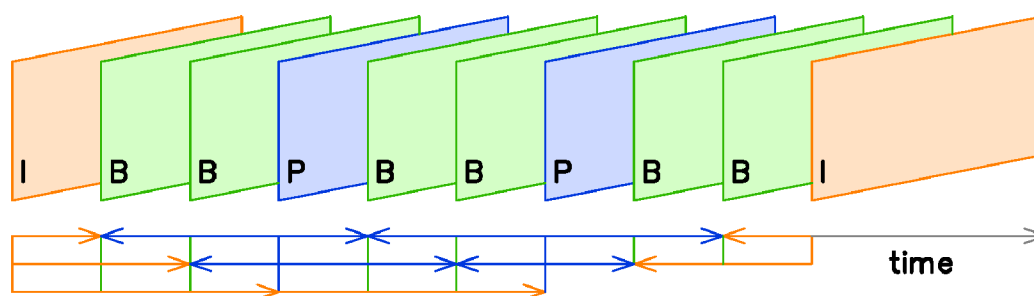
**Intra-frame** neboli I Frame kódování je jeden ze dvou typů komprese videa. Video se kóduje jako sled nezávislých snímků, to znamená, že každý snímek je kódován nezávisle na ostatních a není snaha použít podobnosti následujících snímků pro kompresi [11]. Tento typ kódování je velmi jednoduchý a nenáročný na výpočetní sílu. Nevýhodou je vysoký datový tok. Mezi typické zástupce patří MJPEG a Apple ProRes.

## 2.3 Inter-frame kódování

Inter-frame kódování je druhý typ komprese videa. Pro enkódování využívá změny mezi sousedními snímky a predikce snímků, resp. využití redundance. Video se kóduje jako sled GOP<sup>1</sup>. GOP obsahuje tyto snímky:

- **I snímek** – Snímek, který je kódován nezávisle na ostatních a může být referenční. Intra-frame kódování je v podstatě sled těchto snímků.
- **P snímek** – Obsahuje informaci o **motion-compensation**<sup>2</sup> relativně k předchozímu snímku. Je také označován jako *delta snímek*, jelikož definuje jenom rozdíl mezi snímky. Může referencovat I nebo jiný P snímek.
- **B snímek** – Stejně jako P snímek obsahuje informaci o rozdílu předchozího snímku, ale navíc obsahuje informaci o snímku následujícím. Může referencovat I nebo P snímek.

Struktura GOP je obvykle definovaná jako dvojice čísel, obvykle M, N. M definuje vzdálenost mezi nejbližšími I nebo P snímky a N definuje vzdálenost mezi I snímky. Například pro M=3 a N=12 je struktura GOP IBBPBBPBBPBI.



Obr. 2.1: Příklad Group of Pictures. Šipky pod snímky ukazují reference jednotlivých snímků.[11]

Při příliš vysoké vzdálenosti mezi jednotlivými I snímky může docházet k tvorbě artefaktů<sup>3</sup> při dekódování. Nicméně vysoký počet I snímků značně zvyšuje datový tok. Vzhledem k podobné velikosti I snímků ale I snímky stabilizují datový tok – proto je žádoucí používat Intra-frame kompresi nebo Inter-frame kompresi s vysokým počtem I snímků pro případy, kdy je žádoucí stabilní, nekolíšavý datový tok – typicky online přenosy videa.

Inter-frame komprese není příliš praktická pro editaci videa, jelikož se při střihání musí počítat s ostatními snímky a tím se zvyšuje náročnost na výpočetní sílu.

<sup>1</sup>Group Of Pictures – skupina snímků

<sup>2</sup>Algoritmus, který predikuje snímek na základě pohybu předchozích snímků

<sup>3</sup>Výrazné vizuální zkreslení obrazu

## 3 H.264

H.264 je průmyslový standard pro kompresi digitálního videa, jiným označením také MPEG-4 Part 10, Advanced Video Coding (MPEG-4 AVC). Je faktickým nástupem standardu MPEG-2 a momentálně je jedním z nejpoužívanějších formátů pro kompresi videa.

### 3.1 Historie

H.264 vznikl na podnět agentury ITU-T<sup>1</sup> společně s MPEG<sup>2</sup>. Vznikl jako pokračovatel linie formátů H.26X a byl vyvíjen pod názvem H.26L. Vzhledem k společné tvorbě pod ITU-T a MPEG je formát označován jako H.264 i MPEG-4 AVC, kombinovaně **H.264/MPEG-4 AVC**. Vývoj H.264 začal v roce 1998 pod názvem H.26L za cílem zdvojnásobit enkódovací efektivitu oproti libovolnému existujícímu standardu v té době (tehdy H.262/MPEG-2 Part 2 a H.263). V roce 2001 se ITU-T a MPEG spojily do Joint Video Team (JVT) kvůli finalizaci standardu, která byla dokončena v březnu 2003. Od té doby bylo do standardu přidáno dalších 19 dodatků, přidávající mimo jiné podporu SVC<sup>3</sup> nebo MVC – 3D obraz.

### 3.2 Profily a úrovně

Standard definuje několik různých profilů, z nichž každý podporuje část vlastností pro využití v různých aplikacích. To prakticky znamená, že dekodér pro H.264 video nemusí podporovat všechny profily, ale stačí jenom část nižších profilů (například dekodér pro mobilní telefon nemusí podporovat profil pro 4K 60fps video). Tím se ušetří značné prostředky pro vývoj kodeků, jelikož značná část kodeků, především pro koncové zařízení jsou založená na HW bázi jako jsou FPGA nebo ASIC, např. dedikovaný HW akcelerátor na SoC<sup>4</sup> v mobilním telefonu.

	Baseline	Extended	Main	High	High 10	High 4:2:2	High 4:4:4 Predictive
I and P Slices	Yes	Yes	Yes	Yes	Yes	Yes	Yes
B Slices	No	Yes	Yes	Yes	Yes	Yes	Yes
Multiple Reference Frames	Yes	Yes	Yes	Yes	Yes	Yes	Yes
In-Loop Deblocking Filter	Yes	Yes	Yes	Yes	Yes	Yes	Yes
CAVLC Entropy Coding	Yes	Yes	Yes	Yes	Yes	Yes	Yes
CABAC Entropy Coding	No	No	Yes	Yes	Yes	Yes	Yes
Interlaced Coding (PicAFF, MBAFF)	No	Yes	Yes	Yes	Yes	Yes	Yes
8x8 vs. 4x4 Transform Adaptivity	No	No	No	Yes	Yes	Yes	Yes
Quantization Scaling Matrices	No	No	No	Yes	Yes	Yes	Yes
Separate Cb and Cr OP control	No	No	No	Yes	Yes	Yes	Yes
Separate Color Plane Coding	No	No	No	No	No	No	Yes
Predictive Lossless Coding	No	No	No	No	No	No	Yes
	Baseline	Extended	Main	High	High 10	High 4:2:2	High 4:4:4 Predictive

Obr. 3.1: Srovnání podpory jednotlivých vlastností H.264 v různých profilech. Obecně platí, že "vyšší"profil automaticky podporuje vlastnosti nižších profilů.[19]

<sup>1</sup>ITU-T je telekomunikační složka ITU, která je specializovaná agentura OSN zabývající se problematikou informačních a komunikačních technologií.

<sup>2</sup>Moving Picture Experts Group, skupina založená skupinami ISO a IEC pro zavedení standardů v audiovizuálních formátech.

<sup>3</sup>Scalable Video Coding, podpora video proudu obsahující další, menší video proud.

<sup>4</sup>System on Chip – integrovaný obvod, integruje všechny potřebné počítačové komponenty na jediném čipu.

Všechny profily standardně podporující práci v 4:2:0 chroma podvzorkování a 8 bitový rozsah barev. Mezi nejvíce používané profily patří tyto:

- **Baseline Profile** (BP) – Profil vyžadující nejjednodušší implementaci, typicky využíván pro videokonference a mobilní telefony a jiné aplikace nevyžadující vysokou kvalitu.
- **Extended Profile** (XP) – Profil vycházející z Baseline, používá se v online video přenosech díky větší datové robustnosti
- **Main Profile** (MP) – Profil využíván pro vysílání digitální televize standardu DVB. Nicméně se nepoužívá pro vysílání ve vysokém rozlišení (HDTV) a byl z větší části nahrazen High profilem.
- **High Profile** (HiP) – Profil využíván především pro obsah ve vysokém rozlišení, jako je Blu-ray disk nebo DVB HDTV vysílání.
- **High 4:2:2 Profile** (Hi422P) – Nadmnožina High profilu umožňující práci v 4:2:2 chroma podvzorkování a 10 bitový rozsah barev.
- **High 4:4:4 Predictive Profile** (Hi444PP) – Nadmnožina High 4:2:2 profilu umožňující práci v nejvyšším chroma podvzorkování 4:4:4 a 14 bitový rozsah barev. Navrch podporuje kódování snímku jako 3 rozdílné barevné plochy.

Úroveň (Level) je specifická sada omezení (velikost snímků, makrobloky apod.) pro jednotlivé profily. Společně s profilem jsou povinným nastavením pro kodek a je třeba je dodržovat pro zachování kvality přehrávání videa na koncových zařízeních. Je to zapisováno jako Profil@Úroveň, např. Main@L4.1.

Tab. 3.1: Tabulka hodnot omezení pro jednotlivé úrovně pro profil High.

Úroveň	Makrobloků/Snímek	Maximální datový tok	Maximální rozlišení
1	99	80 kbps	176x144@15 fps
2	396	2,500 kbps	352x288@30 fps
3	1620	12,500 kbps	720x576@25 fps
3.1	3600	17,500 kbps	1280x720@30 fps
4	8192	25,000 kbps	2048x1024@30 fps
4.1	8192	62,500 kbps	2048x1024@30 fps
5	22080	168,750 kbps	3672x1536@26.7 fps
5.1	36864	300,000 kbps	4096x2304@26.7 fps

### 3.3 Entropické kódování

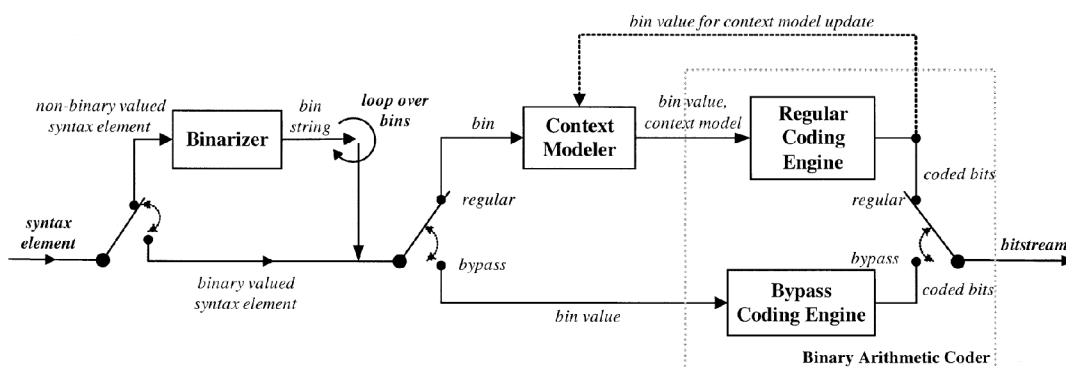
Entropické kódování je metoda bezeztrátové komprese dat. Funguje tak, že přiřazuje prefixový kód každému unikátnímu symbolu na vstupu. Komprese probíhá tak, že se každý vstupní symbol s pevnou délkou zamění za korespondující prefixový kód, jehož délka je nepřímě úměrná s pravděpodobností výskytu. To znamená, že nejčastější symboly mají nejkratší kód.



Tento typ kódování využívá mimo jiné aritmetické a Huffmanovo kódování. Huffmanovo kódování je využito ve starších standardech MPEG-2 a MPEG-4 Part 2. Pro H.264 byly vyvinuty dva nové typy entropického kódování – CABAC a CAVLC.

### 3.3.1 CABAC

**CABAC (Context-adaptive binary arithmetic coding)** je jedním z typů entropického kódování používaném v H.264/MPEG-4 AVC<sup>5</sup>. Vychází, jak už název napovídá, z aritmetického kódování. Oproti předchozím používaným algoritmům (aritmetické, Huffman) nabízí mnohem lepší kompresi a je klíčovým algoritmem pro kvalitní kompresi H.264. Nicméně pro jeho vysokou náročnost na výpočet oproti CALVC je dostupný pouze v profilech Main a High.



Obr. 3.2: Schéma algoritmu entropického kódování CABAC. [20]

Algoritmus probíhá v následujících fázích:

1. **Binarizace** – CABAC využívá binární aritmetické kódování, proto se nebinární vstupní symboly jako např. vektory musí prvně převést na binární kód. Tento proces je podobný převádění symbolu na kód variabilní délky (Huffmanovo kódování), tento kód je ale ještě dodatečně kódován.
2. **Kontextový model** – kontextový model je pravděpodobnostní model pro jeden či více bitů binarizovaného symbolu. Tento model je generován ze statistik nedávno kódovaných symbolů. Kontextový model ukládá pravděpodobnost výskytu jednotlivých bitů.
3. **Aritmetické kódování** – Aritmetický kóder enkóduje vstupní bity podle zvoleného kontextového modelu.
4. **Aktualizace pravděpodobnosti** – Po každém symbolu se aktualizuje pravděpodobnost výskytů bitů v symbolu v kontextovém modelu používaném v bodě (2).
5. **Výstup** – Výstupem tohoto kódování je proud bitů, který se později enkóduje kódem do NAL<sup>6</sup>

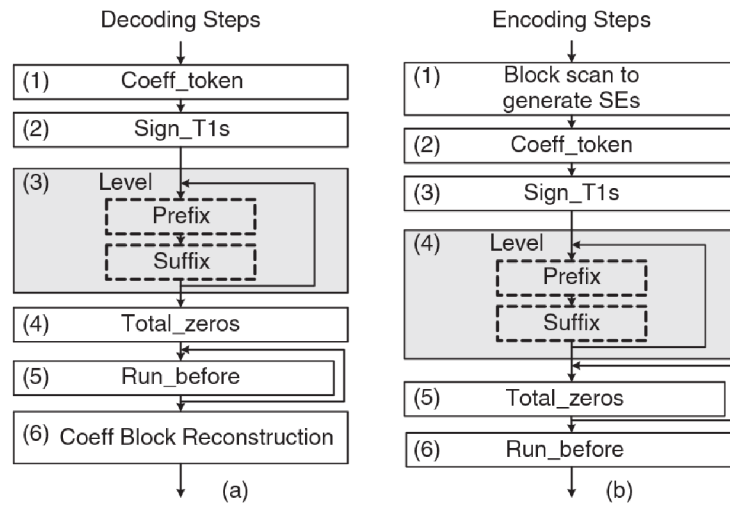
### 3.3.2 CAVLC

**CAVLC (Context-adaptive variable-length coding)** je druhý typ bezztrátové komprese používaný v H.264/MPEG-4 AVC založený na entropickém kódování. Pro jeho nižší

<sup>5</sup>Používá se i u jeho nástupce H.265/MPEG-H HEVC, kde se používá u všech profilů jako komprese.

<sup>6</sup>Network Abstraction Layer, základní jednotka H.264 video proudu

náročnost na výpočet se používá místo CABAC. Nicméně nedosahuje jeho kvalit v efektivitě komprese. CAVLC je podporován ve všech profilech na rozdíl od CABAC a je hlavní kompresí v profilech Baseline a Extended.



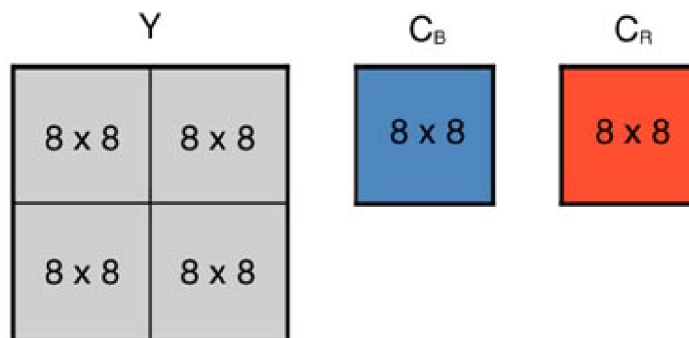
Obr. 3.3: Diagramové schéma algoritmu entropického kódování CAVLC: (a) Dekodér a (b) enkodér[21]

### 3.4 Makroblok

**Makroblok** je blok pixelů ve snímku, s nimiž pracují obrazové komprese pracující na diskrétní kosinové transformaci.<sup>7</sup>

Makroblok má typicky 16x16 pixelů a je nadále dělen na transformační nebo predikční bloky. Makrobloky se dělí na jednotlivé YCrCb složky, tudíž jsou ovlivněné chroma podvzorkováním.

H.264 umožňuje použití i menších makrobloků, např. 4x4 pixely.



Obr. 3.4: Makroblok v 4:2:0 vzorkování [10]. Y označuje jasovou složku, C<sub>B</sub> modrý posun a C<sub>R</sub> červený posun.

<sup>7</sup>Typicky H.26x řada – H.262/MPEG-2 Part 2, H.263, H.264/MPEG-4 AVC.

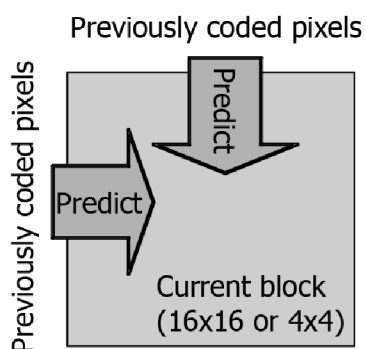


## 3.5 Enkódovací proces

### 3.5.1 Predikce

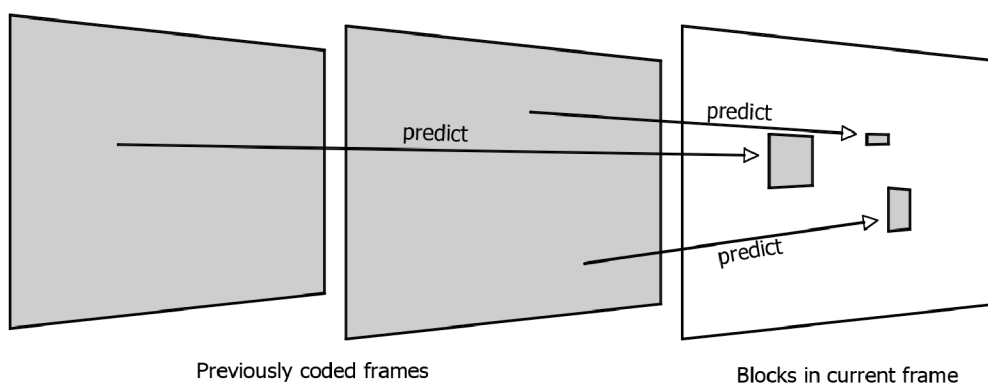
Enkodér zpracovává snímek videa jako jednotky makrobloků. Tvoří predikci makrobloku založeném na dříve zpracovaných datech buď ze současného snímku (**intra** predikce) nebo z jiných snímků, který již byly enkódovány (**inter** predikce). Využívá se zde *motion compensation*. Predikční metody podporované v H.264 umožňují použití různých velikostí makrobloků najednou pro dosažení větší přesnosti predikce a není zde omezení na použití pouze 16x16 makrobloků.

**Intra predikce** používá 16x16 a 4x4 bloky pro predikci makrobloku z okolních, dříve zpracovaných bloků pixelů ve stejném snímku.



Obr. 3.5: Intra predikce. Pro současně zpracovaný blok se používá predikce z dříve zpracovaných bloků. [2]

**Inter predikce** používá **dynamickou velikost bloků** - od 16x16 do 4x4 pro predikci pixelů v současném snímku z podobných částí v dříve zpracovaných snímcích. Inter-frame predikce v starších kompresních standardech dovolovala použít pouze jeden (u I a P snímků) nebo dva (u B snímků) referenční snímky. H.264 navyšuje počet referenčních snímků až na 16.



Obr. 3.6: Inter predikce. Je zde vidět různá velikost jednotlivých makrobloků. Používají se zde bloky z předchozích snímků posunuté kompenzací pohybu. [2]

### 3.5.2 Transformace a kvantizace

Blok predikovaných vzorků je transformován pomocí 4x4 nebo 8x8 **celočíslné transformace**, která je aproximovaná forma diskrétní kosinové transformace. Výstup transformace je sada koeficientů.

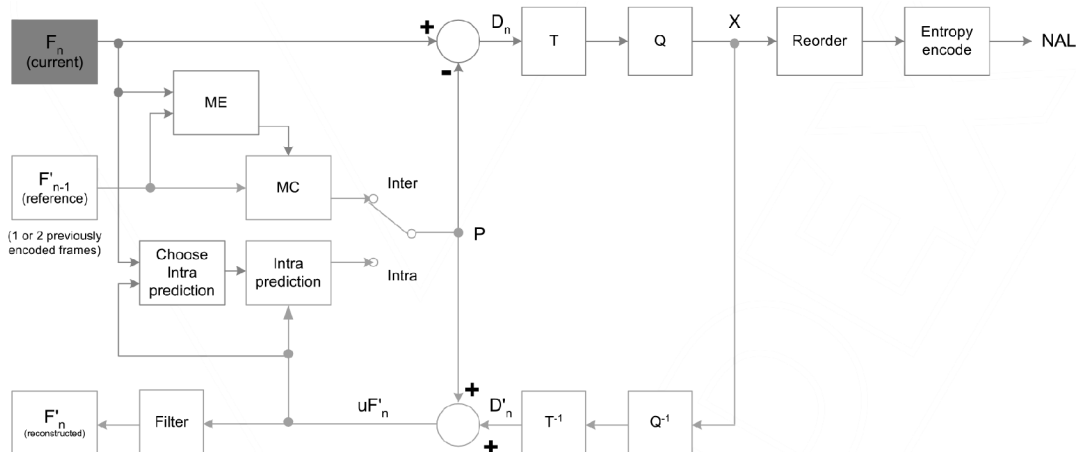
Výsledek transformace je kvantizován<sup>8</sup>. Kvantizace redukuje přesnost transformačních koeficientů podle tzv. **kvantizačního parametru (QP)**. Typický výsledek je blok, ve kterém je většina koeficientů nulová. Čím větší je kvantizační parametr, tím je více koeficientů nulových, čímž se zvyšuje komprese za úkor kvality dekodovaného obrazu.

### 3.5.3 Bitstream enkódování

Předchozí kroky enkódování tvoří sadu hodnot, které musí být enkódovány do výsledného komprimovaného bitstreamu<sup>9</sup>. Mezi tyto hodnoty patří:

- Kvantizované transformační koeficienty
- Informace pro enkodér umožňující zpětnou rekonstrukci predikce
- Informace o struktuře komprimovaných dat a metod komprimace
- Informace o kompletní video sekvenci - rozlišení, snímková frekvence apod.

Všechny tyto hodnoty jsou následně zakódovány pomocí CABAC/CAVLC do výsledného bitstreamu ve formátu NAL - Network Abstraction Layer.



Obr. 3.7: Celkové schéma enkódování H.264.  $F$  označuje snímek,  $ME$  je motion estimation - predikce pohybu,  $MC$  kompenzace pohybu,  $T$  je celočíselná transformace a  $Q$  je kvantizace. Vrchní část schématu popisuje enkódování současného snímku, spodní část popisuje rekonstrukci předchozího snímku pro použití inter predikce. Po kvantizaci jsou hodnoty seřazeny a enkódovány vhodným entropickým enkódováním - CABAC/CAVLC.[22]

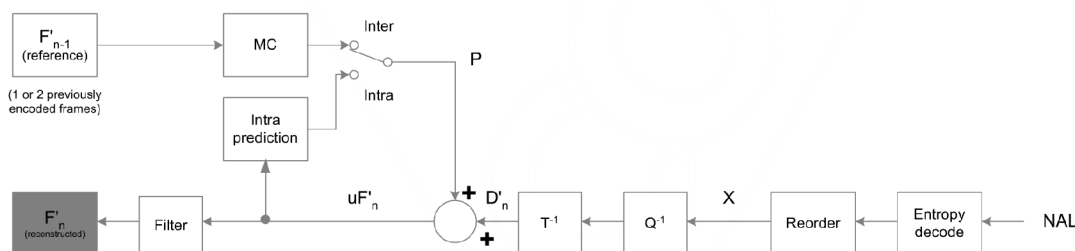
<sup>8</sup>Každý koeficient je dělen celočíselnou hodnotou.

<sup>9</sup>Bitstream - proud bitů.

## 3.6 Dekódovací proces

Dekódování obstarává sada kroků, která reflektuje enkódování v opačném pořadí:

1. Dekódování vstupního bitstreamu z NAL a jeho dekomprimace
2. Rozřazení dat pro získání sady kvantizovaných koeficientů, které jsou následně dekvantizovány a inverzně transformovány
3. V tomto kroku máme dekódovaný snímek. Pokud se jedná o I snímek, není třeba rekonstruovat predikci z jiných snímků a pokračuje se dále, jinak (v případě P a B snímků) se rekonstruuje predikce z informací v hlavičce bitstreamu
4. Na snímku může být aplikován filtr pro redukcii artefaktů



Obr. 3.8: Celkové schéma dekódování H.264.  $F$  označuje snímek,  $MC$  je kompenzace pohybu,  $T^{-1}$  je inverzní celočíselná transformace a  $Q^{-1}$  je dekvantizace. Schéma je velmi podobné schématu pro rekonstrukci snímku 3.7 na stránce 17.[22]

## 3.7 Budoucnost

H.264 bude v budoucnu nahrazen standardem **H.265/MPEG-H HEVC**, který je stejně jako H.264 vyvíjen společně pod ITU-T a MPEG. První verze standardu byla finalizována v lednu 2013 a specifikace byla formálně ratifikována 13. dubna 2013.

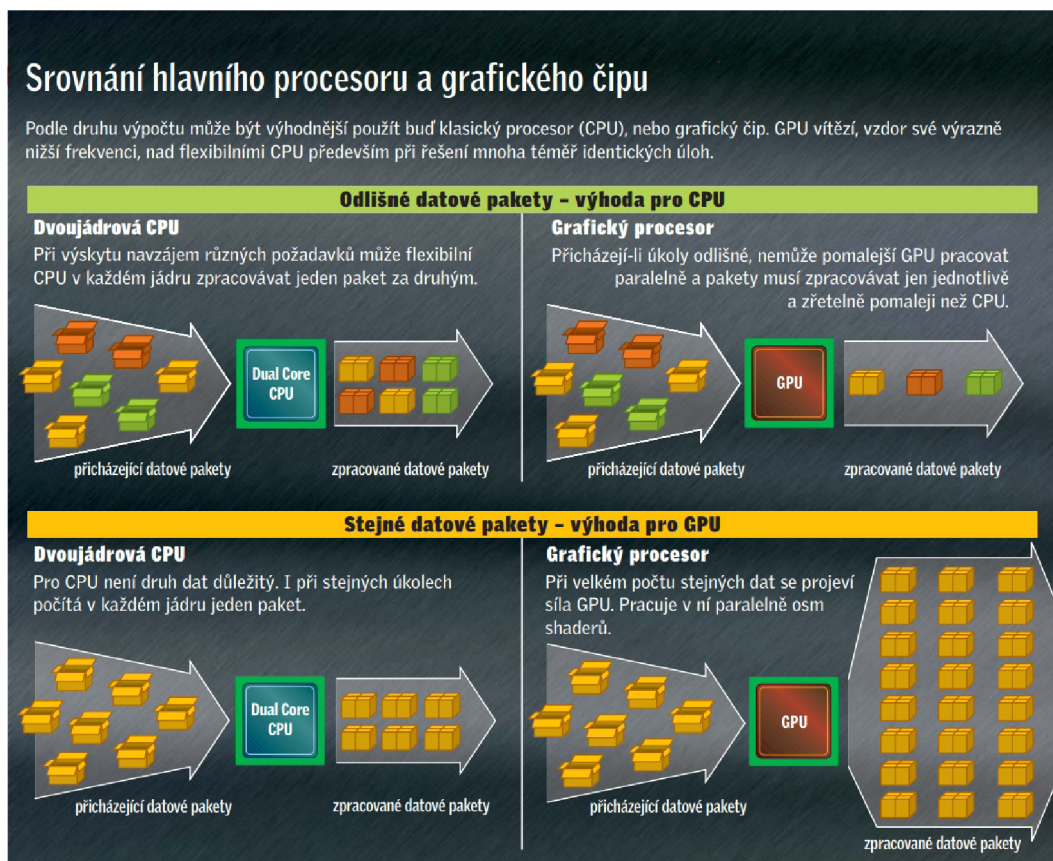
H.265 vychází z H.264, který byl designován především na video ve Full HD a 2K rozlišení, nicméně videa ve vyšším rozlišení jako je 4K v něm dosahují velmi vysokých datových toků. H.265 vznikl za účelem zdvojnásobit kompresní efektivitu H.264 při stejné kvalitě videa. Kompresní efektivita umožňuje podporu **8K Ultra HD** rozlišení - až 8192x4320 a mnohem lepší kvalitu 4K videa.

Z H.264 přebírá mnoho vlastností - např. CABAC je primárním entropickým kódováním pro všechny profily H.265. Nicméně vylepšuje některé základy - např. makrobloky jsou nahrazeny pokročilými CTU - coding tree unit.

Na rozdíl od H.264, kterému trvalo několik let se prosadit, H.265 už má podporu v řadě zařízeních - např. mobilní telefon iPhone 6 již má HW podporu pro přehrávání H.265 videa a několik populárních video přehrávačů jako je VLC už nabízí přehrávání. NVIDIA přidala HW podporu pro enkódování H.265 videa v GPU řady Maxwell.

## 4 GPGPU

**GPGPU** (General-purpose computing on graphics processing units) je obecně způsob využití paralelizace na GPU<sup>1</sup> k výpočtu obecných algoritmů. GPU se primárně využívá na vykreslování dat na obrazovku, nicméně se v posledních letech začala využívat jejich lepší efektivita oproti CPU<sup>2</sup> např. v algoritmech na zpracování obrazu nebo algoritmů využívající paralelizaci.



Obr. 4.1: Srovnání zpracování dat na CPU a GPU. V horní části je popsána úloha se špatnou možností paralelizace, např. složité fyzikální rovnice s navzájem závislými hodnotami, kde je malý počet silných CPU jader výkonnější. V dolní části je případ algoritmu s vysokou možností paralelizace.[14]

Technické předpoklady pro GPGPU přineslo zavedení grafických karet podporující DirectX 8 – od tohoto momentu je možné využít volně programovatelné shadery<sup>3</sup>, nicméně o reálném využití se dá mluvit až od nástupu DirectX 9.0C a GPU s unifikovanými shadery<sup>4</sup>.

Mezi standardy využívající GPGPU patří mimo CUDA i např. OpenCL či Stream (pro grafické karty AMD).

<sup>1</sup>Graphics Processing Unit – grafický akcelerátor je specializovaný procesor nacházející se obvykle na grafické kartě.

<sup>2</sup>Central Processing Unit – procesor.

<sup>3</sup>Výpočetní jednotky GPU.

<sup>4</sup>Předtím byly shadery rozdělené na pixel a vertex shadery.

## 4.1 CUDA

CUDA<sup>5</sup> je hardware/software platforma vyvinutá firmou NVIDIA, která umožňuje spouštět programy v C/C++ a FORTRAN na NVIDIA GPU. Poprvé byla představena v roce 2006 a první SDK ve verzi 1.0 bylo představeno o rok později pro karty NVIDIA Tesla architektury G80. Původně byla CUDA vyvinuta pouze pro profesionální karty (Tesla, Quadro), nicméně SDK bylo nakonec uvolněno i pro běžné mainstreamové karty GeForce.

Ve verzi SDK 3.0 (architektura Fermi) byla přidána nativní podpora pro double precision výpočty a podpora rekurze. SDK 4.0 přidává podporu MultiGPU (SLI).

## 4.2 NVCUVID

NVCUVID je API<sup>6</sup> nabízející přístup k hardwarově akcelerovanému dekódování na GPU firmy NVIDIA podporující CUDA Compute 1.1 (řada Tesla a vyšší). V současnosti podporuje hardware akceleraci těchto video kodeků:

- MPEG-2/VC-1
  - Akcelerace dekódování pro čipy G8X a G9X
  - Podpora plného dekódování od řady GF1xx a čipové sady MCP79
  - Podpora Full HD (1080p) přehrávání včetně Blu-ray obsahu
- H.264/AVCHD
  - Baseline, Main, High profil do úrovně 4.1
  - Podpora plného dekódování Full HD obsahu včetně Blu-ray
  - Podpora B snímků, datový tok až 45 Mbps

NVCUVID může běžet jak na VP<sup>7</sup>, tak na samotném GPU dle řady grafické karty a kodeku. Jako součást CUDA je provázané s grafickým výstupem, tudíž podporuje 3D API jako OpenGL a DirectX a umožňuje výstup dekódovaného obrazu na monitor, jelikož je ale jeho výstup v standardním nekomprimovaném formátu NV12, lze tento výstup použít jako vstup pro enkódování videa do jiných kodeků.

## 4.3 NVENC

NVENC je ASIC<sup>8</sup>, který je součástí NVIDIA GPU od řady GeForce 600 založené na řadě Kepler. Přímo konkuruje Intel Quick Sync Video (Intel) a Video Codec Engine (AMD).

NVENC je přímým nástupcem NVCUVENC, který narozdíl od NVENC fungoval přímo na CUDA jednotkách, ne na samotném čipu. To například omezuje využití všech CUDA jednotek, u NVENC lze použít jak CUDA jednotky, tak samostatný NVENC čip zároveň, jak je tomu v napsaném programu, který využívá CUDA pro dekódování (NVCUVID) a NVENC pro enkódování.

Dle NVIDIA jsou výhody enkóderu založeného na HW následující:

- Nízká spotřeba – ASIC má nízkou spotřebu oproti CPU/GPU
- Nízká odezva – Snížení režie pro přenos dat mezi CPU/GPU a operační paměť
- Vysoký výkon

---

<sup>5</sup>Compute Unified Device Architecture

<sup>6</sup>Application Programming Interface – rozhraní pro přístup ke knihovnam/jiným programům

<sup>7</sup>Video Processor – součást NVIDIA GPU, byl odstraněn z jádra v řadě GF200 a nahrazen technologií PureVideo.

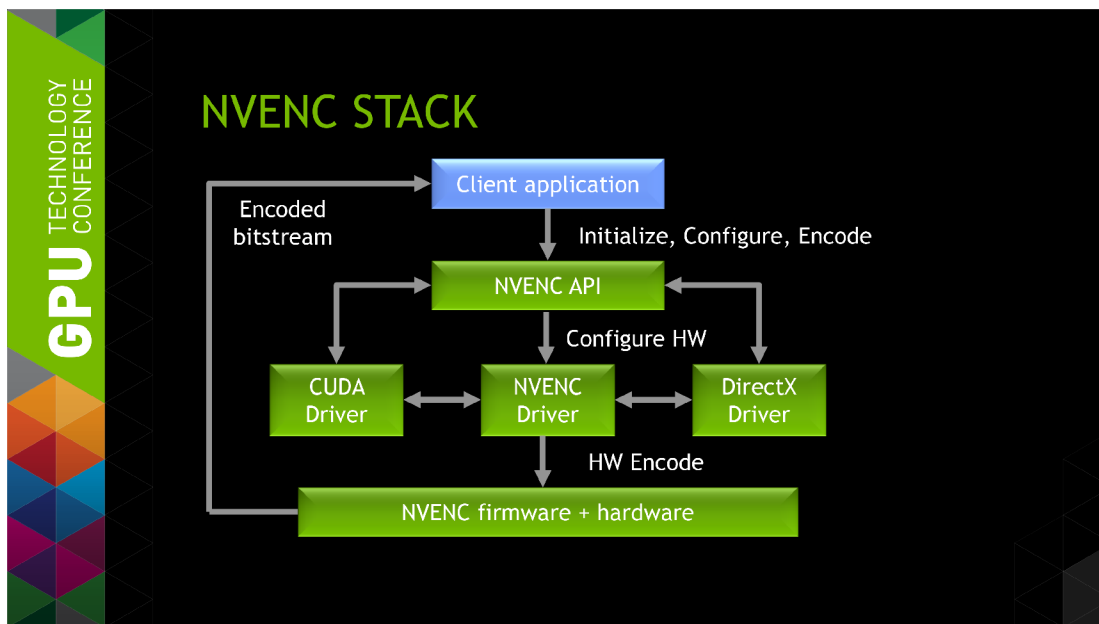
<sup>8</sup>Application-specific integrated circuit – integrovaný obvod tvořený na míru k nějakému účelu, narozdíl od CPU, který je univerzální



- Snadný vývoj – Přístup skrze API, není potřeba vyvíjet samotný enkodér

API pro NVENC momentálně nabízí enkódování jak H.264, tak H.265 (HEVC, nástupce H.264) na kartách s čipy Kepler a Maxwell. Na běžných GeForce kartách je možné enkódovat dva streamy souběžně na jednom systému. Na GRID, Tesla a Quadro kartách je to limitované pouze množstvím čipů a dostupnou video pamětí. API je unifikováno pro Windows a Linux, podporované CPU architektury jsou x86/x64.

Celý enkodér je naprogramován v HW, ke kterému se přistupuje skrze API. Tím odpadá složité programování enkodéru a mimo jiné unifikaci vývoje, jelikož není potřeba optimalizovat vývoj pro různé řady GPU.



Obr. 4.2: Tok programu pro NVENC. Klientská aplikace volá NVENC API – inicializuje potřebné proměnné a posílá data na zpracování. NVENC API následně zařídí zavolání příslušných ovladačů pro HW, na kterém se následně data enkódují.[17]

## 5 IMPLEMENTACE ENKODÉRU

### 5.1 Programovací jazyk a nástroje

Zvoleným jazykem implementace byl jazyk C++ zejména pro jeho nativní podporu pro CUDA volání a knihovny. Druhým faktorem byla vysoká rychlost výsledného programu oproti jiným programovacím jazykům.

Pro tvorbu GUI je využita knihovna Qt ve verzi 5.3. Ta byla zvolena především díky jednoduchosti a rychlosti a neposledně kvůli jednoduché kombinaci společně s MS Visual Studio. Neposledně je Qt také multiplatformní, díky čemuž je možné program snadno portovat na Unixové či Linuxové systémy, kde je CUDA také podporováno.

Hlavní IDE<sup>1</sup> bylo Microsoft Visual Studio ve verzi Professional 2013. To bylo zvolené pro jednoduchou provázanost s CUDA díky NVIDIA pluginu a jejímu vlastnímu kompilátoru nvcc. Při tvorbě GUI byl využit Qt Visual Studio Add-in, který umožňuje kompilovat GUI přímo ve Visual Studiu. GUI bylo tvořeno v Qt Designeru.

### 5.2 Struktura kódu

Program vychází z volně dostupného sample *NvTranscoder*, který je součástí NVENC 5.0 SDK. Z něj byla převzata základní struktura a především použité knihovny.

#### 5.2.1 Dekódování

Funkce a datové struktury pro enkódování vstupního souboru, pracující s NVCUVID jsou uloženy v souborech `VideoDecoder.cpp/h`. Třída `CudaDecoder` implementuje dekodér vstupního souboru, kterým může být video formátu MPEG-2 či H.264.

Při inicializaci dekodéru metodou `InitVideoDecoder()` se jako první vytvoří ukazatel na video soubor API funkcí `cvidCreateVideoSource()`. Touto metodou se zároveň uloží metadata o souboru do proměnné `oVideoSourceParameters`, se získanými metadaty se vytvoří dekodér šitý na míru danému videosouboru metodou `cvidCreateDecoder()`. Samotné načítání ze souboru řeší parser tvořený metodou `cvidCreateVideoParser()`.

Inicializace tedy vypadá zhruba takto:

```
void CudaDecoder::InitVideoDecoder()
{
    ...
    // otevri soubor a vytvor na nej ukazatel
    cvidCreateVideoSource(&m_videoSource);
    // zjist parametry videa
    cvidGetSourceVideoFormat();
    // vytvor dekodér
    cvidCreateDecoder();
    // vytvor parser pro soubor
    cvidCreateVideoParser();
    ...
}
```

<sup>1</sup>Integrated Development Enviroment – vývojové prostředí

## 5.2.2 Enkódování

Funkce a datové struktury pro enkódování nekomprimovaných snímků, které využívají NVENC jsou uloženy v souborech `VideoEncoder.cpp/h`. Třída `VideoEncoder` implementuje vlastní enkodér. Pro vytvoření objektu enkodéru je třeba vytvořený CUDA kontext, který je argumentem konstruktoru.


Enkodér inicializuje NVENC HW po tom, co dostane informace od dekodéru o vstupním souboru, především rozlišení, aby bylo možné vytvořit buffer v adekvátní velikosti, jehož velikost je mimo rozlišení značně ovlivněna i počtem B snímků. Také zjišťuje parametry pro enkódování, to dostává od uživatele v GUI. Kontroluje se zároveň, zda-li je preset podporovaný na grafické kartě – některé pokročilé nastavení jsou dostupné až od GPU řady Maxwell, například enkódování ve vyšším chroma podvzorkování než 4:2:0.

Před enkódováním je třeba vytvořit si I/O buffer v metodě `AllocateIOBuffers()`. Ten se tvoří jenom jeden, je ale možné ho rozšířit pro podporu enkódování více jak jednoho video proudu zároveň.

Enkódování snímku začne, jakmile je v Frame Queue (datová struktura pro jednotlivé snímky) nějaký snímek. Zavolá se metoda `EncoderFrame()`, ve které se prvně vyčká na uvolnění bufferu, který se následně zaplní snímkem. Následně se zamkne kontext funkcí `cuvidCtxLock()`, jelikož se následně buffer a parametry nakopírují do CUDA paměti funkcí `cuMemcpy2D()`. Potom se může kontext odemknout funkcí `cuvidCtxUnlock()` a nakonec se snímek enkóduje metodou `NvEncEncodeFrame()`.

Po posledním enkódovaném snímku se pošle "prázdný snímek" enkodéru, kterým se signalizuje konec enkódování a tím se vyprázdní buffer metodou `FlushEncoder()`.

Parametry enkódování jsou nastaveny zvolením presetu v GUI.



### QUERY ENCODER PRESETS

PRESET	Encoder Settings	APPLICATION
HIGH QUALITY	B Frames, CABAC, 8x8 Transform, All Intra Modes, All Inter Modes*, VBR RC, GopLength 30	TRANSCODING HIGH BITRATE
HIGH PERFORMANCE	No B Frames, CAVLC, P16x16, Intra16x16 and Intra4x4 Modes, VBR, GopLength 30	MULTIPLE TRANSCODING
LOW LATENCY HQ	No B Frames, CABAC, All Intra, All Inter Modes, Single frame VBV 2 PASS, Infinite GOP,	CLOUD GAMING, MIRACAST, VIDEO CONFERENCING
LOW LATENCY HP	No B Frames, CABAC, All Intra and Inter Modes, Single frame VBV 2 PASS, Infinite GOP, Smaller Search Range compared to LOW LATENCY HQ	CLOUD GAMING, MIRACAST

Obr. 5.1: Dostupné presetu pro NVENC. High quality je nejkvalitnější preset – vhodný pro enkódování jednoho videa s vysokým datovým tokem, high performance nejvýkonnější – vhodný pro více souběžných enkódování. Jejich low latency verze jsou vhodné pro aplikace vyžadující nízkou odezvu, jako jsou například videokonference nebo streamování online her. [17]



Samotné enkódování vypadá zjednodušeně takhle:

```
NVENCSTATUS VideoEncoder::EncodeFrame()
{
    ...
    // dostan buffer
    *pEncodeBuffer = m_EncodeBufferQueue.GetAvailable();
    // pokud není dostupný, počkej
    if (!pEncodeBuffer)
    {
        pEncodeBuffer = m_EncodeBufferQueue.GetPending();
    }
    // zamkni kontext
    cuvidCtxLock();
    // nakopíruj data do CUDA paměti
    cuMemcpy2D(&memcpy2D);
    // odemkni kontext
    cuvidCtxUnlock();
    // enkoduj snímek na HW
    m_pNvHWEncoder->NvEncEncodeFrame();
    ...
}
```

### 5.2.3 GUI

GUI je definované souborem `nvencgui.ui`, který je fakticky XML soubor ve formátu, se kterým pracuje Qt Designer.

Kód pro definici jednotlivých metod pro třídu GUI `NVENCGUI` jsou v souborech `nvencgui.cpp/h`. Třída `NVENCGUI` je podtřídou třídy `QMainWindow`, která tvoří hlavní okno v Qt GUI.

Pro jednotlivé akce jsou vytvořeny spárované signály/sloty. Enkódování se spouští tlačítkem `Encode!`, které pošle signál do slotu `NVENCGUI::EncodeButtonClick()`. Ten spouští metodu `NVENCGUI::StartTranscoding`, která zkontroluje, zda-li jsou zadané všechny platné parametry, pokud ne, tak vyhodí chybovou hlášku. Pokud ano, spouští hlavní smyčku programu `nvencgui::Transcode()`. Tato metoda se spouští v jiném vlákně kvůli výkonu a odezvě GUI. Zároveň se zamknou jednotlivé části GUI, aby se nedalo omylem spustit enkódování vícekrát.

### 5.2.4 Hlavní smyčka

Jádrem celého programu je metoda `NVENCGUI::Transcode()`, ve které se provádí veškeré dekodování/enkódování.

Tato metoda postupně inicializuje všechny části `NVCUVID` a `NVENC API`:

1. Inicializuje `CUDA` na `GPU` funkcí `cuInit()`
2. Zkontroluje, zda-li jsou vstupní soubory platné video soubory
3. Vytvoří `CUDA` kontext, na kterém se bude kód spouštět
4. Přepne kontext na `NVCUVID` a inicializuje dekodér a frontu snímků
5. Inicializuje `NVENC API` na `CUDA` kontextu, nastaví enkodéru parametry nastavené v GUI
6. Vytvoří vlákno, na kterém se spustí dekodér, ukládající snímky do fronty snímků

7. Spustí smyčku, kde se vyčkává na snímky z dekodéru

Enkódovací smyčka programu vypadá následovně:

```
// dokud není video dekodováno a frame queue není prázdná
while (!(pFrameQueue->isEndOfDecode() && pFrameQueue->isEmpty()))
{
    ...
    // načti snímek a dekoduj ho
    cuvidMapVideoFrame();
    // vezmi snímek a enkoduj ho
    EncodeFrame();
    // vypust snímek
    cuvidUnmapVideoFrame();
    // vyprázdní frame queue
    pFrameQueue->releaseFrame();
    ...
}
```

Po dokončení se vyprázdní všechny buffery, smaže dekodér/enkodér, uvolní NVCUVID a CUDA kontext a vrátí se režie zpět GUI, kde se může spustit další enkódování.

## 6 TESTOVÁNÍ

### 6.1 Testovací sestava

Tab. 6.1: Testovací sestava

Procesor	Intel Core i5-4690K – 4.3 GHz
Operační paměť	8 GB DDR3
Grafická karta	nVidia GTX 660
Disk	Samsung 840 EVO 240 GB
Operační systém	Windows 8.1 Pro
Ovladač GPU	350.12

### 6.2 Testovací videa

Pro testování jsem zvolil tři volně dostupná videa s různou délkou a rozlišením:

Tab. 6.2: Testovací videa

	Big Buck Bunny	Sintel	The Simpsons – Trailer
Rozlišení	1920x1080	1280x544	1280x544
Snímková frekvence	24 fps	24 fps	23.976 fps
Datový tok	9.3 Mbps	5.5 Mbps	4.3 Mbps
Délka	9 min 56 s	14 min 48s	2 min 17 s
Počet snímků	14315	21312	3288
Profil	Main@L4.1	High@L4.1	Main@L5.1

### 6.3 Alternativní enkodéry

#### 6.3.1 x264

x264 je open-source kodek pro enkódování H.264 videa. Má otevřený zdrojový kód šířený pod licencí GNU GPL a je vyvíjena společně s projektem VideoLAN, pod který patří například populární VLC media player. Tento kodek používá CPU pro enkódování. Pro přístup k tomuto kodeku je použita utilita FFmpeg.

### 6.3.2 MainConcept H.264

MainConcept H.264 je kodek vyvinutý firmou MainConcept GmbH, dceřinnou společností známé firmy DivX. Tento kodek používá CPU i GPU pro enkódování. Pro přístup k této knihovně je použit program Adobe Media Encoder. GPU podpora je přidána plug-inem Rovi TotalCode.

## 6.4 Testovací metodika

V NVENC Enkodéru se měřil čas interním časovačem zabudovaném v programu. U x264 je čas měřen jednoduchým skriptem, kdy se vypsala čas okamžitě před a po dokončení skriptu, který zároveň volal i utilitu FFMpeg a z těchto časů se vypočítal výsledný čas. Čas u MainConcept H.264 byl získán z logu programu Adobe Media Encoder.

Všechny měření byly opakované 5x a z nich byl vypočítán průměr pro nejlepší přesnost. Pro porovnání pouze videa bylo vypnuté exportování audia.

Parametry pro jednotlivé enkodéry byly zadané takto:

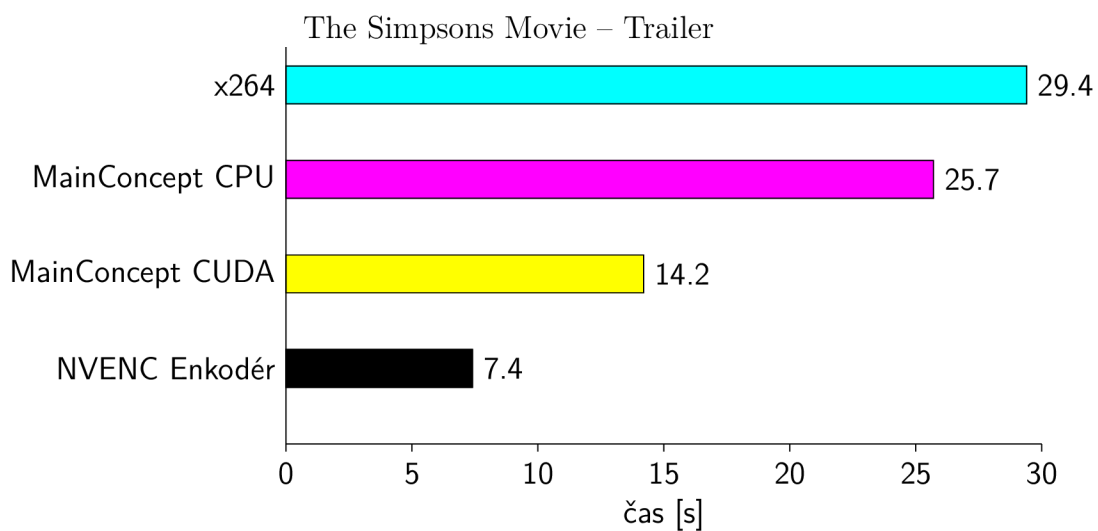
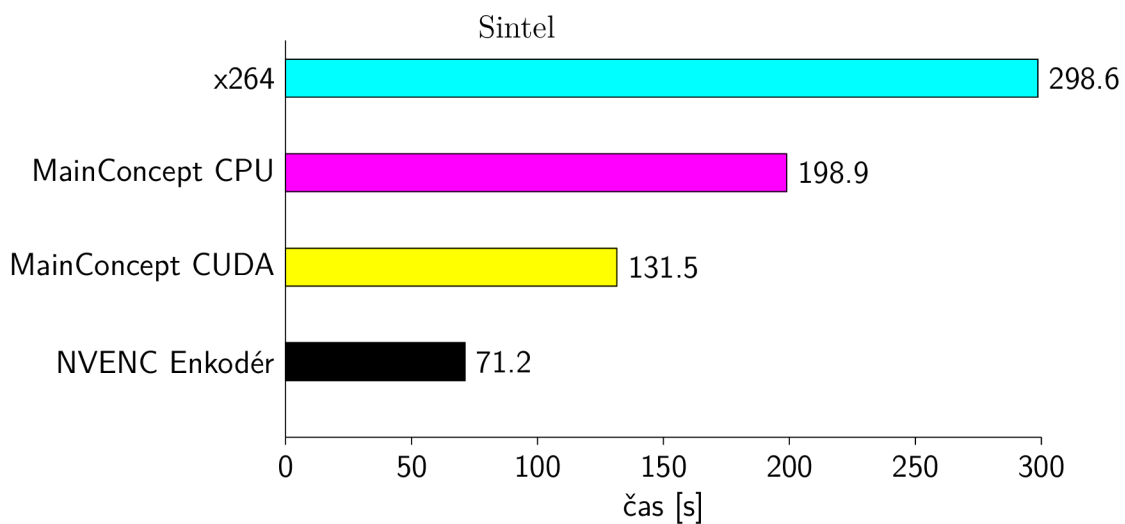
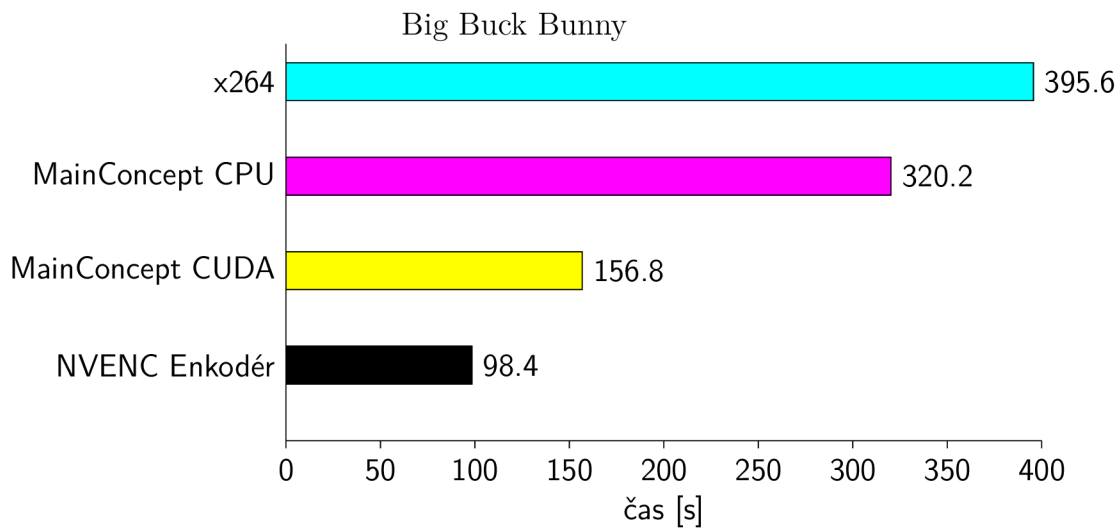
- NVENC Enkodér
  - Preset: High Performance
  - Preset GUID: Constant Bit Rate
  - Bitrate: 10000/5000 kbps
  - Resolution/Framerate: Same as source
- MainConcept H264
  - Frame rate: Same as source
  - Field Order: None (Progressive)
  - Profile: High
  - Level: 4.0
  - Bitrate Encoding: CBR
  - Bitrate Level: Custom 10000/5000 kbps
- x264 – FFMpeg byl volán s následujícími parametry: -c:v libx264 -b:v 5000k -an

## 6.5 Výsledky měření

Za základ, ze kterého se počítal celkový výkon se bere výsledek enkodéru x264. Jednotlivé výsledky byly sečteny a z nich vypočítán průměrný výkon. Výsledky byly zaokrouhleny na jednu desetinnou čáru.

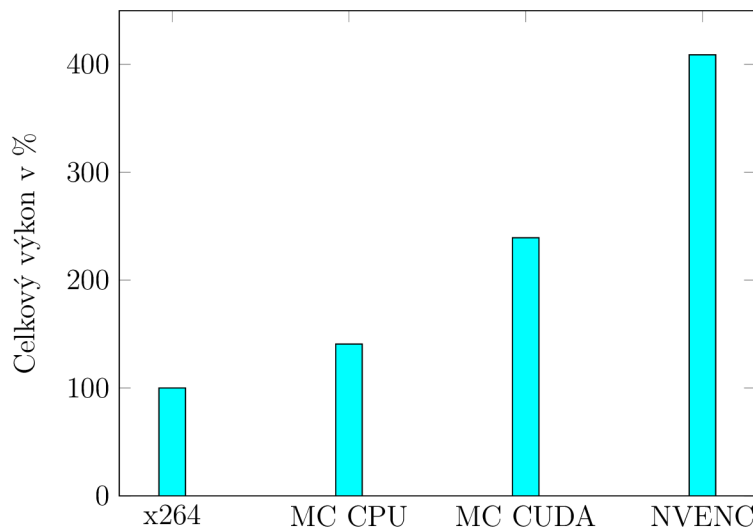
Tab. 6.3: Tabulka výsledků

	Big Buck Bunny	Sintel	The Simpsons	Výkon
x264	395.6 s	298.6 s	29.4 s	100%
MainConcept CPU	320.2 s	198.9 s	25.7 s	140.7%
MainConcept CUDA	156.8 s	131.5 s	14.2 s	239.2%
NVENC Enkodér	98.4 s	71.2 s	7.4 s	408.8%

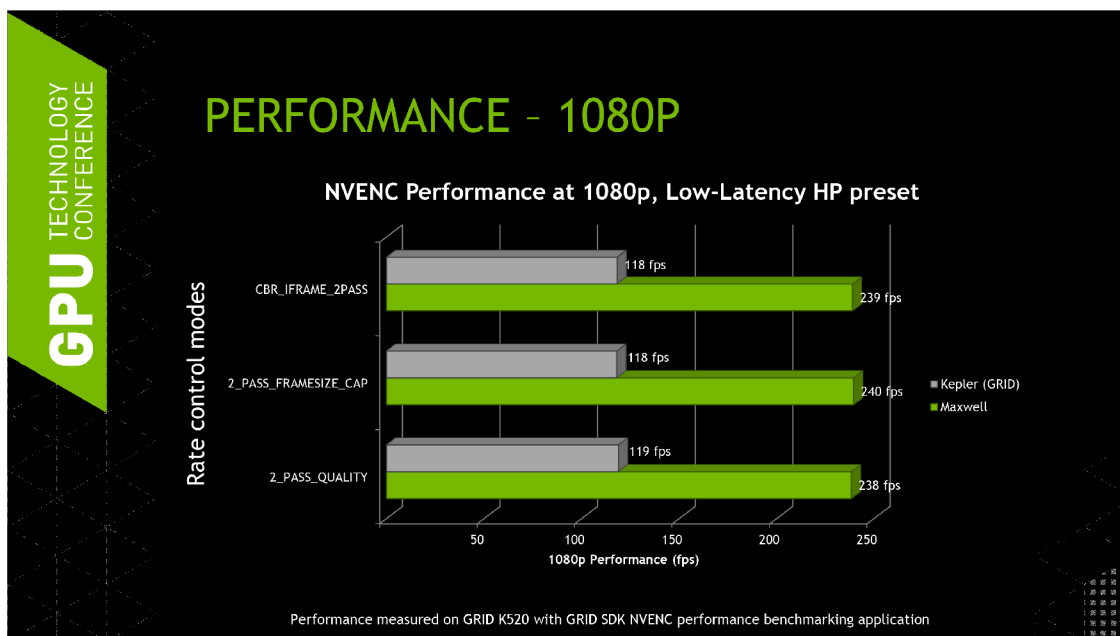


## 6.6 Porovnání výkonu

Z tabulky a grafů je jasně vidět značné zrychlení NVENC enkodéru oproti enkodérům založených na CPU (x264, MainConcept CPU) i oproti enkodéru založeném na čistém CUDA (MainConcept CUDA).



Oproti enkodéru MainConcept (CPU) je NVENC řešení v průměru 2,9x rychlejší. Oproti populárnímu enkodéru x264 je NVENC dokonce více jak 4x rychlejší. Oproti čistému CUDA řešení (MainConcept CUDA) je NVENC řešení 1,7x rychlejší. Výsledky jsou výborné pro CUDA a NVENC i přes to, že je použit velmi silný přetaktovaný procesor a GPU řady Kepler. GPU řady Maxwell mají dle NVIDIA více jak 2x vyšší výkon oproti řadě Kepler v enkódování videa.



Obr. 6.1: Srovnání výkonu Kepler a Maxwell. Maxwell je v průměru 2x výkonnější než Kepler ve všech rate control módech – Constant Bitrate a Variable Bitrate.[17]

## 7 ZÁVĚR

Tato práce měla za úkol vytvořit program implementující enkodér formátu H.264 s využitím GPGPU, konkrétně platformy CUDA. Tato implementace si kladla za cíl nabídnout lepší výkon jako běžně dostupné CPU implementace enkodéru. Implementaci enkodéru na GPU umožňuje NVENC API, které je součástí grafických karet NVIDIA a CUDA od řady Kepler. NVENC je nicméně pouze enkodér a pokud není na vstup přivedeno čisté nekomprimované video v YUV/NV12 formátu, bylo třeba napsat i dekodér – pro ten bylo zvoleno NVCUVID API, které umožňuje dekódování vstupního videa na CUDA HW. Tyto API byly zvoleny pro jejich výkon, zejména NVENC – toto API je tak výkonné, že NVIDIA odstranila podporu pro starší API (NVCUVENC) v ovladačích R337.

Dekodér i enkodér byly implementovány v podstatě současně – jejich funkce se navzájem prolíná i přes to, že každý pracuje s jiným API. Klíčovým prvkem obou prvků byla struktura snímkové fronty, která umožňovala při běhu dvou vláken (jedno pro dekodér, jedno pro enkodér) bezchybnou synchronizaci – enkodér okamžitě dostal dekódovaný snímek na vstup. Následně bylo implementováno GUI, které bylo vytvořeno zejména pro jednoduché užití programu a možnost budoucího rozšíření.

Výsledky rychlosti enkodéru byly nadmíru uspokojivé – NVENC enkodér byl oproti CPU enkodéru x264 více jak 4x rychlejší, přestože byl použit přetaktovaný aktuálně nejsilnější čtyřjádrový procesor od Intelu a GPU se starší verzí NVENC. Oproti enkodéru využívající starší API je rychlost téměř dvojnásobná, čímž se potvrdila lepší výkonnost na dedikovaném ASIC oproti čistě CUDA jádrům.

Program jako takový je možný dále optimalizovat – například přednačítáním snímků do bufferu, jež je ale velmi náročné na operační paměť. Je možné je také snadno rozšířit – třeba přidáním podpory více vstupních video formátů, aktuálně je podporováno pouze MPEG-2 a H.264. Velmi snadno lze enkodér H.264 převést na H.265 enkodér za předpokladu, že je v systému přítomna GPU řady Maxwell. Nicméně je podpora H.265 v NVENC stále v plenkách.

## LITERATURA

- [1] Richardson, I.E. *The H.264 Advance Video Compression Standard. Second Edition.* Chichester, UK: John Wiley & Sons, Ltd, 2010. ISBN 9780470516928
- [2] Richardson, I.E. *An Overview of H.264 Advanced Video Coding* [online] Dostupné z URL <http://www.vcodex.com/images/uploaded/469323879727520.pdf> [cit. 2015-29-04]
- [3] Česká Tisková Kancelář *Analogové vysílání v Česku definitivně skončilo. Budoucnost bude patřit hybridnímu* [online] Dostupné z URL <http://byznys.ihned.cz/c1-53897210-analogove-vysilani-v-cesku-definitivne-skoncilo-budoucnost> [cit. 2015-30-04]
- [4] Wiegand, T.; Sullivan, G. J.; Bjontegaard, G.; Luthra, A. *Overview of the H.264/AVC Video Coding Standard* [online] Dostupné z URL [http://ip.hhi.de/imagecom\\_G1/assets/pdfs/csvt\\_overview\\_0305.pdf](http://ip.hhi.de/imagecom_G1/assets/pdfs/csvt_overview_0305.pdf) [cit. 2015-12-05]
- [5] Wikipedia, The Free Encyclopedia *YCbCr* [online] Dostupné z URL <http://en.wikipedia.org/wiki/YCbCr>, [cit. 2013-12-05]
- [6] Kerr, D. A. *Chrominance Subsampling in Digital Images* [online] Dostupné z URL <http://dougkerr.net/Pumpkin/articles/Subsampling.pdf> [cit. 2015-13-05]
- [7] Wikipedia, The Free Encyclopedia *Color space* [online] Dostupné z URL [http://en.wikipedia.org/wiki/Color\\_space](http://en.wikipedia.org/wiki/Color_space), [cit. 2015-05-09]
- [8] Richardson, I.E. *H.264 and MPEG-4 Video Compression* Chichester, UK: John Wiley & Sons, Ltd, 2003. ISBN 0470848375
- [9] Wikipedia, The Free Encyclopedia *Display resolution* [online] Dostupné z URL [http://en.wikipedia.org/wiki/Display\\_resolution](http://en.wikipedia.org/wiki/Display_resolution), [cit. 2015-05-09]
- [10] Líška, D. *Digitální terestriální televize DVB-T: Technické minimum - MPEG 2* [online] Dostupné z URL [http://www.digitalnitelevize.cz/magazin/dvb-t/dvb-t-technologie/technicke\\_minimum\\_mpeg2.html](http://www.digitalnitelevize.cz/magazin/dvb-t/dvb-t-technologie/technicke_minimum_mpeg2.html), [cit. 2015-05-07]
- [11] Marshall, D. *MPEG Video* [online] Dostupné z URL <http://www.cs.cf.ac.uk/Dave/Multimedia/node256.html>, [cit. 2015-05-08]
- [12] Grant *Video Formáty a kodeky videa* [online] Dostupné z URL <http://www.grantvideo.cz/kodeky.html>, [cit. 2015-05-08]
- [13] NVIDIA Corporation *NVIDIA VIDEO CODEC SDK* [online] Dostupné z URL <https://developer.nvidia.com/nvidia-video-codec-sdk>, [cit. 2015-04-29]
- [14] Littschwager, T. *GPGPU: 200x rychleji než CPU* [online] Dostupné z URL <http://www.chip.cz/soubory/dokumenty/83a948e01eb1fc11fac84205f198daa8.pdf>, [cit. 2015-05-01]
- [15] NVIDIA Corporation *NVENC - NVIDIA HARDWARE VIDEO ENCODER* [online] Dostupné z URL [http://developer.download.nvidia.com/compute/nvenc/v4.0/NVENC\\_AppNote.pdf](http://developer.download.nvidia.com/compute/nvenc/v4.0/NVENC_AppNote.pdf), [cit. 2015-05-11]



- [16] Patait, A. *High performance video encoding using NVIDIA GPUs* GPU Technology Conference, 2014
- [17] Mohapatra, S. *Detail overview of NVENC Encoder API* [online] Dostupné z URL <http://on-demand.gputechconf.com/gtc/2014/presentations/S4654-detailed-overview-nvenc-encoder-api.pdf>, [cit. 2015-04-28]
- [18] Wilt, N. *The CUDA handbook : a comprehensive guide to GPU programming* Upper Saddle River, New Jersey: Addison-Wesley, 2013. ISBN 978-0-321-80946-9
- [19] Igarta, M. *A STUDY OF MPEG-2 AND H.264 VIDEO CODING* [online] Dostupné z URL <https://engineering.purdue.edu/~ace/thesis/igarta/thesis-igarta.pdf>, [cit. 2015-05-10]
- [20] Marpe, D., Schwarz, H., Wiegand, T. *Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard* IEEE Trans. Circuits and Systems for Video Technology, Vol. 13, No. 7, stránka 620–636, Červen 2003.
- [21] Tian X., Le, T.M., Lian, Y. *Entropy Coders of the H.264/AVC Standard* 2011. ISBN 978-3-642-14702-9
- [22] Richardson, I.E. *A Technical Introduction to H.264 / AVC* [online] Dostupné z URL <http://vcodex.com/images/uploaded/342454811635986.pdf>, [cit. 2015-5-14]

# SEZNAM PŘÍLOH

A GUI programu	34
B Obsah přiloženého CD	34

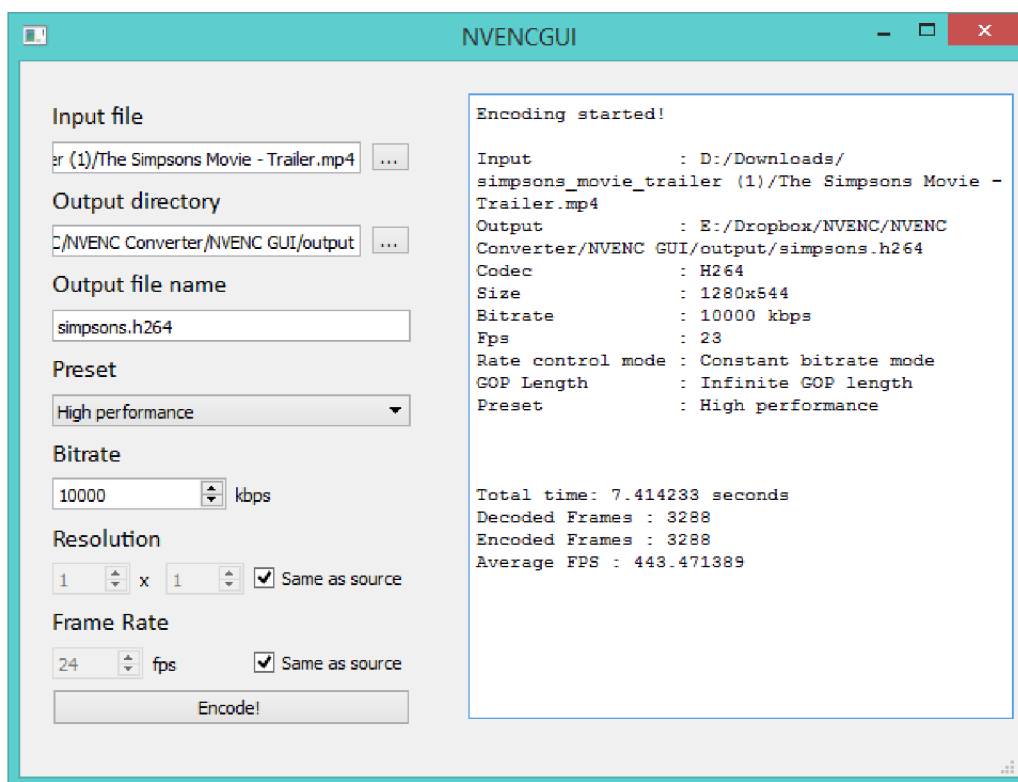
## A GUI PROGRAMU

Jednotlivé parametry, které lze zadat, jsou následující:

- Input File - Vstupní soubor, může být libovolného kontejneru obsahující H.264 video
- Output Directory - Výstupní složka
- Output file name - Jméno výstupního souboru, doporučená koncovka je .h264 kvůli kompatibilitě s většinou video přehrávačů
- Preset - Preset nastavení enkodéru, pro karty řady Kepler může být High Performance / Quality, pro Maxwell je přidána možnost Low Latency High Performance / Quality
- Bitrate - Datový tok videa v kbps
- Resolution - Rozlišení v pixelech, může být zadáno ručně nebo nastavit stejné jako vstupní soubor. Maximální rozlišení je 1920x1200 kvůli omezení NVCUVID
- Frame Rate - Snímková frekvence, může být 1 až 240 fps

Při začátku enkódování se vypíší zadané parametry do textového pole v pravé části programu. Po dokončení enkódování se vypíší metriky enkódování:

- Total time - Celková doba enkódování
- Decoded Frames - Počet dekodovaných snímků
- Encoded Frames - Počet enkódovaných snímků
- Average FPS - Průměrná rychlost enkódování v snímcích za sekundu, počítá se jako celková doba enkódování / počet enkódovaných snímků



Obr. A.1: GUI programu

## B OBSAH PŘILOŽENÉHO CD

CD obsahuje:

- Zdrojový kód tohoto programu ve složce "src"
- Zdrojový kód LaTeXu ve složce "latex"
- Spustitelný program
- Tuto práci ve formátu pdf
- Návod ke kompilaci programu