VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ZTRÁTOVÁ KOMPRESE PLENOPTICKÝCH FOTOGRAFIÍ

LOSSY LIGHT FIELD COMPRESSION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **DRAHOMÍR DLABAJA**
AUTHOR

VEDOUCÍ PRÁCE **Ing. DAVID BAŘINA, Ph.D.**
SUPERVISOR

BRNO 2019

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2018/2019

Zadání bakalářské práce

Student: **Dlabaja Drahomír**
Program: Informační technologie
Název: **Ztrátová komprese plenoptických fotografií**
Lossy Light Field Compression
Kategorie: Zpracování obrazu

Zadání:

1. Seznamte se s technologií plenoptických fotografií a formáty pro uložení těchto dat.
2. Seznamte se s technikami, které se používají ke ztrátové kompresi obrazu, videa a volumetrických dat.
3. Vytvořte kompresní postup pro ztrátovou kompresi plenoptických obrazů. S tímto postupem experimentujte. Výstupem tohoto bodu je programová knihovna.
4. Srovnejte účinnost vytvořené metody s existujícími formáty. Jako kritérium použijte kompresní poměr při dané kvalitě.

Literatura:

- Touradj Ebrahimi, Siegfried Foessel, Fernando Pereira, Peter Schelkens, JPEG Pleno: Toward an Efficient Representation of Visual Reality, IEEE Multimedia, Oct-Dec 2016.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2 podrobně doložené technickou zprávou. Funkční kompresní řetězec k bodu 3.

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Bařina David, Ing., Ph.D.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 15. května 2019
Datum schválení: 6. listopadu 2018

Abstrakt

Cílem této práce je návrh, implementace a vyhodnocení ztrátové kompresní metody pro plenoptické fotografie. Navržená metoda zrcadlí metodu JPEG do čtyř rozměrů a přináší nové myšlenky, které vedou k lepšímu kompresnímu výkonu. Korelace mezi pohledy plenoptické fotografie je v obou směrech využita provedením čtyřrozměrné diskrétní kosinovy transformace. Ztráty jsou na obrázek aplikovány kvantizací, podobně jako v metodě JPEG. Navržená metoda je implementována jako programová knihovna v jazyce C++. V této práci je provedeno srovnání navržené metody s videometodou HEVC a s metodami pro kompresi obrazu JPEG, JPEG 2000 a HEVC intra. Navržená kompresní metoda dosahuje lepšího kompresního výkonu v porovnání s ostatními testovanými metodami pro obrázky s větším počtem pohledů. Pro obrázky s menším počtem pohledů a pro velmi nízké datové toky dosahuje lepších výsledků videometoda HEVC.

Abstract

The aim of this paper is to propose, implement and evaluate a new lossy compression method for light field images. The proposed method extends the JPEG method to four dimensions and brings new ideas which lead to better compression performance. Correlation between light field views is exploited in both dimensions by four-dimensional discrete cosine transformation. The lossy part of the encoding is performed by quantization, similarly to the JPEG method. The proposed method is implemented as a program library in a C++ language. This paper compares the proposed method to JPEG, JPEG 2000 and HEVC intra image compression methods and HEVC video compression method. The results show that the proposed method outperforms the reference methods with images with a higher amount of views. HEVC video method is better for images with fewer views or for very low bitrates.

Klíčová slova

světelné pole, ztrátová komprese, plenoptická fotografie, transformační kódování, diskrétní kosinová transformace, JPEG, JPEG 2000, HEVC

Keywords

light field, lossy compression, plenoptic photography, transform coding, discrete cosine transform, JPEG, JPEG 2000, HEVC

Citace

DLABAJA, Drahomír. *Ztrátová komprese plenoptických fotografií.* Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. David Bařina, Ph.D.

Ztrátová komprese plenoptických fotografií

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Davida Bařiny, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Drahomír Dlabaja
14. května 2019

Poděkování

Rád bych poděkoval Ing. Davidu Bařinovi, Ph.D. za poskytnutou odbornou pomoc.

Obsah

Kapitola 1

Úvod

S rozvojem nových technologií pro zachycení scény v prostoru a čase rostou také nároky na paměťovou a přenosovou kapacitu. Příkladem takové technologie je plenoptická fotografie. Dosud se jednalo spíše o experimentální odvětví počítačové grafiky, ale s nástupem výkonnějšího hardwaru v kombinaci s rostoucí dostupností plenoptických fotoaparátů se plenoptická fotografie rozšiřuje i mezi širokou veřejnost.

Taková fotografie nese nejen úhlovou informaci o paprscích světla dopadajících na hlavní čočku fotoaparátu, ale také informaci o poloze dopadu, což společně tvoří mnohem kompletnější reprezentaci zachycené scény. Oproti klasické fotografii se zde neztrácí informace o hloubce, což umožňuje nové způsoby zpracování, které s běžnou fotografií nejsou možné. V pořízeném snímku lze například libovolně měnit rovinu ostření. S tím souvisí možnost nastavení virtuální clony a dokonce lze měnit i úhel pohledu na scénu. Bezesporu největší výhodou je ovšem možnost zachycené světelné pole zpětně reprodukovat, a to jak ve světě digitálním – ve virtuální realitě, tak ve světě reálném – s využitím holografického displeje. To lze navíc provést nezávisle na komplexitě scény a bez nutnosti obrázek převádět na point cloud či jiný 3D formát.

V praxi to naneštěstí znamená mnohonásobně větší nároky na paměťovou a přenosovou kapacitu, efektivní komprese je proto nutností. Tato technologie je poměrně dlouho známá, v minulosti byly pro tento typ dat využívány zejména existující kompresní metody pro kompresi obrazu a videa. Tyto metody však nedokáží naplno využít kompresní potenciál, který plenoptické fotografie mají.

Tato práce si klade za cíl prozkoumat možnosti komprese plenoptických fotografií. Je zde popsán návrh kompresní metody, její implementace a vyhodnocení, včetně srovnání s existujícími metodami. Kompresní metoda navržená v této práci zrcadlí metodu JPEG do čtyř rozměrů. Tím využívá korelace mezi pohledy plenoptické fotografie v obou směrech. Navržená metoda obsahuje řadu nových myšlenek, kterými se přímo odlišuje od metody JPEG a které vedou k lepšímu kompresnímu výkonu. Metoda je implementována jako programová knihovna v jazyce C++. Součástí této práce jsou také nástroje, které tuto knihovnu využívají. Tato metoda je vyhodnocena na třech typech plenoptických fotografií. První část vyhodnocení se věnuje srovnání konfigurací implementovaného kodeku s cílem najít nejlepší konfigurace pro různé typy plenoptických fotografií. Ve druhé části je implementovaná knihovna srovnána s metodami pro kompresi obrazu JPEG, JPEG 2000, HEVC intra a s videometodou HEVC.

Výsledky této práce mohou představovat důležité poznatky pro budoucí výzkum kompresních metod pro plenoptické fotografie, protože znalost, že korelaci mezi jednotlivými

pohledy plenoptické fotografie lze využít rozšířením metody JPEG do čtyř rozměrů, může být přenesena na jiné metody pro kompresi obrazu.

Kapitola 2 se věnuje popisu technologie plenoptických fotografií. Jsou v ní vysvětleny teoretické pojmy související s plenoptickými fotografiemi, způsoby reprezentace a formáty pro uložení těchto dat. Kapitola 3 popisuje obecné principy ztrátové komprese obrazu a videa. V podkapitole 3.3 je podrobně popsán kompresní řetězec metody JPEG, na kterém je založeno kompresní řešení představené v této práci. Tomuto řešení se věnuje kapitola 4. Poslední kapiola 5 se věnuje experimentům, které srovnávají nastavení navržené metody. Navržená metoda je poté srovnána s existujícími metodami pro kompresi obrazu a videa.

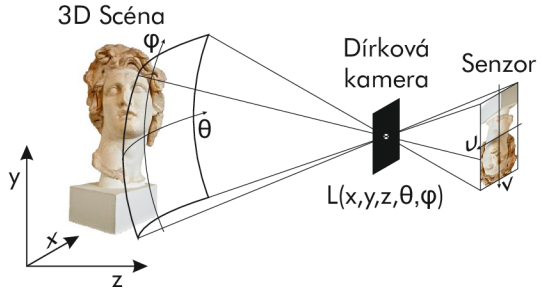
Kapitola 2

Technologie plenoptických fotografií

Tato kapitola podrobně popisuje vše, co stojí za pojmem *plenoptická fotografie*. Plenoptické funkci se věnuje sekce 2.1. Tato funkce definuje rozložení světelných paprsků v prostoru a na jejím pochopení jsou závislé další pojmy. V sekci 2.2 je popsán pojem *4D světelné pole*. Jde o jeden z řezů plenoptické funkce, který slouží jako formální základ pro plenoptickou fotografii. Řezy plenoptické funkce a jejich zachycení je popsáno v sekci 2.3. Sekce 2.4 popisuje technologie pro zachycení plenoptické fotografie. Poslední sekce 2.5 se věnuje přehledu formátů pro uložení plenoptických dat.

2.1 Plenoptická funkce

Plenoptická funkce je sedmirozměrná funkce určující intenzitu světla cestujícího podél paprsku definována jako $L(x, y, z, \theta, \phi, \lambda, t)$, kde (x, y, z, t) jsou časoprostorové souřadnice, (θ, ϕ) jsou úhlové souřadnice a λ je vlnová délka světelného paprsku [1]. Tato funkce je často zjednodušena na pět rozměrů, zanedbávaje čas a vlnovou délku světla. Plenoptická funkce je ilustrována na obrázku 2.1.



Obrázek 2.1: Množství světla procházející bodem v prostoru na souřadnicích (x, y, z) pod úhlem (θ, ϕ) odpovídá hodnotě plenoptické funkce $L(x, y, z, \theta, \phi)$. Pro zjednodušení není brán v úvahu čas a vlnová délka světla. Tento obrázek je vektorizovanou verzí původního obrázku dostupného z [5].

Pokud by byla plenoptická funkce definována, bylo by pomocí ní možné vyjádřit každou scénu, každou fotografii a každý film. Všechno, co kdy bylo, je a bude viděno. Toto není v reálném světě možné, proto slouží plenoptická funkce pouze jako konceptuální model

pro různé metody zachycení scény [12]. Příkladem může být třeba první monochromatická fotografie zachycená pomocí dírkové komory. Tuto fotografii lze definovat jako plenoptickou funkci integrovanou přes viditelné spektrum světla se souřadnicemi ve středu otvoru dírkové komory v čase pořízení a v určitém rozsahu úhlů (θ, ϕ). Tyto úhly si lze představit jako vertikální a horizontální souřadnice na výsledné fotografii. Fotografie ve stupních šedi je proto dvourozměrná funkce $L(\theta, \phi)$.

Podobně lze definovat barevnou fotografii jako trojrozměrnou funkci $L(\theta, \phi, \lambda)$, kde oproti monochromatické fotografii není přes vlnovou délku integrováno, λ se tak stává třetím parametrem. Pokud je k funkci dvourozměrné barevné fotografie přidán parametr reprezentující čas, vznikne čtyřrozměrná funkce $L(\theta, \phi, \lambda, t)$, která reprezentuje dvourozměrné barevné video. Dalším příkladem může být funkce $L(x, y, z, \theta, \phi, \lambda)$ která je navíc parametrizována prostorovými souřadnicemi (x, y, z). Tato funkce definuje kompletní holografickou reprezentací zachycené scény.

Důležitost plenoptické funkce tedy spočívá v tom, že pomocí jejich řezů lze definovat různé reprezentace zachycené scény. Jedním z těchto řezů je také 4D světelné pole, kterému se věnuje následující podkapitola.

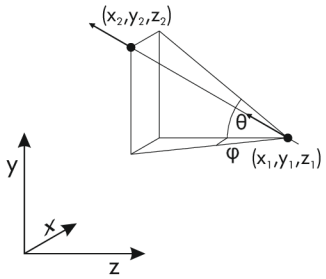
2.2 4D světelné pole

Za předpokladu, že světlo cestuje rovně podél paprsku a jeho intenzita se v průběhu nemění, lze plenoptickou funkci redukovat na čtyřrozměrnou funkci $L(x, y, \theta, \phi)$, zanedbávaje čas a vlnovou délku světla [7, 9]. Tím, že světlo cestuje podél paprsku nezměněno, vzniká v plenoptické funkci redundance, protože jeden světelný paprsek se promítne identicky do všech bodů v prostoru (x, y, z), skrz které tento paprsek prochází. Tato redundance je demonstrována na obrázku 2.2a. Čtyřrozměrné světelné pole oproti tomu zachycuje světelný paprsek právě jednou, a to v bodě, kde se tento paprsek setkává s rovinou (x, y). Případně lze v definici nahradit úhlové souřadnice za prostorové souřadnice a paprsek definovat jako dvojici průniků s rovinami (u, v) a (s, t). To demonstruje obrázek 2.2b. Parametrizace pomocí dvou rovin není nic neobvyklého, obrázek 2.3 demonstruje světelné pole jako čtyřrozměrnou funkci $L(u, v, s, t)$, kde (u, v) představuje rovinu pozorovatele a (s, t) představuje rovinu obrazu. Výhodou této reprezentace světelného pole je nahrazení úhlových souřadnic prostorovými, což je vhodné zejména při počítačovém zpracování. Nevýhodou je, že touto reprezentací nelze vyjádřit paprsek rovnoběžný s rovinou (u, v). Čtyřrozměrné světelné pole tedy obsahuje kompletní reprezentaci statické scény pozorované z roviny v prostoru a slouží jako formální aparát pro plenoptické fotografie.

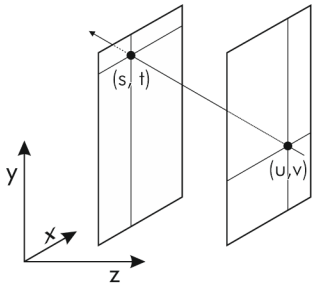
2.3 Zachycení řezů plenoptické funkce

V předchozích podkapitolách jsou popsány řezy plenoptické funkce a jejich souvislost s metodami zachycení scény. Příkladem je třeba barevná fotografie, kterou lze definovat funkcí $L(\theta, \phi, \lambda)$, kde (θ, ϕ) jsou prostorové úhly a λ je vlnová délka světla. Jelikož je reálný svět spojitý, je spojitá i plenoptická funkce. Pokud tedy vychází definice barevné fotografie z plenoptické funkce, znamenalo by to, že scéna je zachycena pro každý úhel pohledu a s nekonečnou přesností.

Většina běžných fotoaparátů nedokáže vyfotografovat kompletní panoramatickou reprezentaci scény, je proto možné funkci zachytit pouze pro určitý interval hodnot úhlů (θ, ϕ). Stejný princip se aplikuje i na další parametry plenoptické funkce. Na běžné fotografii

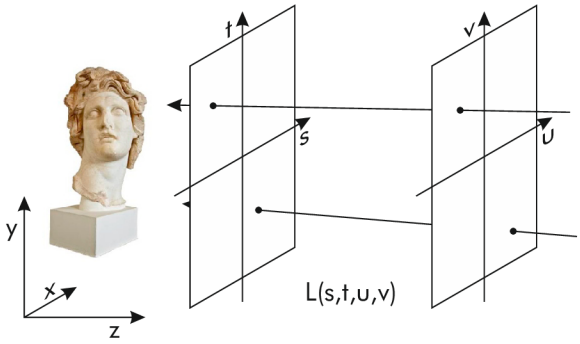


(a) Plenoptická funkce obsahuje redundantní informaci. Paprsek pod úhlem (θ, ϕ) se v bodě (x_1, y_1, z_1) projeví stejnou hodnotou plenoptické funkce, jako v bodě (x_2, y_2, z_2).



(b) Stejný paprsek přitom lze definovat jako dvojici průniků s rovinami (u, v) a (s, t). Tím je redundance odstraněna, protože jeden paprsek nelze definovat více souřadnicemi.

Obrázek 2.2: Redundance v plenoptické funkci, kterou řeší čtyřrozměrné světelné pole.



Obrázek 2.3: Parametrizace čtyřrozměrného světelného pole pomocí průniků paprsků s rovinami (u, v) a (s, t).

například není zachyceno elektromagnetické vlnění mimo viditelné spektrum. Pro různé profesionální účely naopak můžou existovat kamery, které zachycují vlnění i mimo viditelné spektrum.

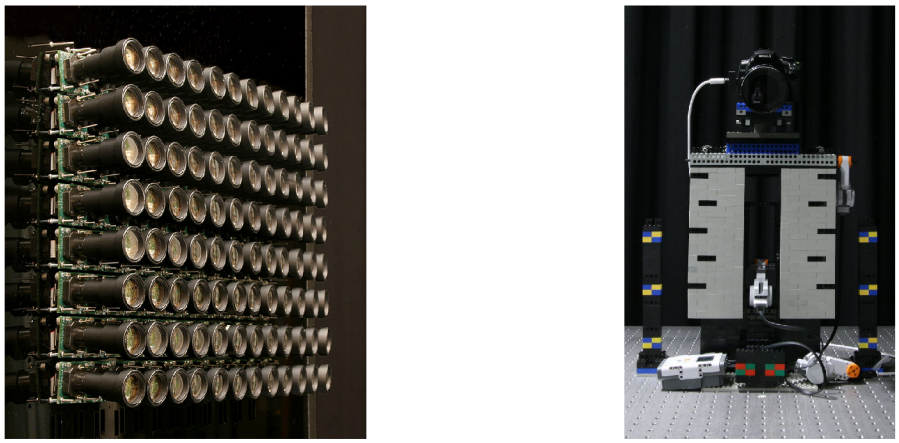
Zároveň není možné dosáhnout nekonečné přesnosti, plenoptická funkce je při zachycení *vzorkována*. Opět platí, že pro každý rozměr plenoptické funkce a pro různé účely zachycení se vzorkování liší. U běžné fotografie každý ocení vysoké rozlišení, tedy jemné vzorkování úhlů (θ, ϕ). U videa je navíc důležitá snímková frekvence, tedy vzorkování času t. Oproti tomu například vlnovou délku světla lze agregovat do trojice hodnot reprezentující červenou, zelenou a modrou barevnou složku bez viditelné ztráty informace. Všechny tyto parametry jsou dány výrobcem fotoaparátu, objektivu či jiného zařízení pro zachycení scény. Stejná pravidla se aplikují i na plenoptické fotografie.

2.4 Zachycení světelného pole

Zachycení plenoptické fotografie se zpravidla provádí jemným vzorkováním úhlů (θ, ϕ) vůči hrubšímu vzrokování prostorových souřadnic (x, y). Rozsahy parametrů jsou dány metodou zachycení. Pro zachycení plenoptické fotografie se využívají nejčastěji dva přístupy.

První z těchto přístupů se vyznačuje relativně velkým rozsahem parametrů (x, y) s velkými rozestupy mezi jednotlivými pohledy. Zachycení tímto přístupem je docíleno dvourozměrným polem klasických fotoaparátů, které zachycují celou scénu v jeden moment [11]. Příklad

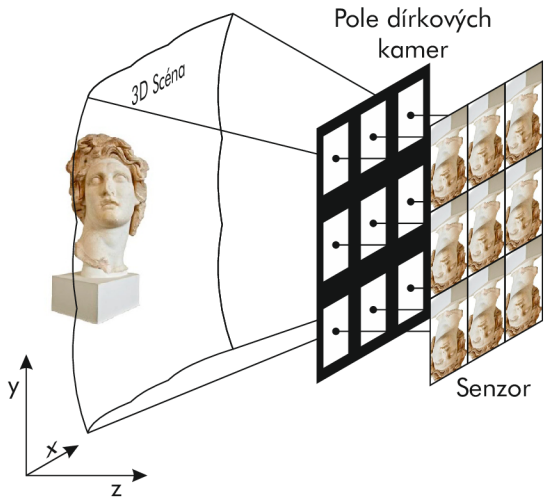
takového pole je demonstrován na obrázku 2.4a. Stejný princip lze simulovat použitím jednoho fotoaparátu, který se mezi fotografiemi pohledů přesouvá po pravidelné mřížce. Tento přístup je levnější, avšak časově náročnější a nelze pomocí něj zachytit dynamické scény. Zařízení s jedním fotoaparátem, které slouží k zachycení plenoptických fotografií, je demonstrováno na obrázku 2.4b. Princip zachycení plenoptické fotografie pomocí pole fotoaparátů je schematicky demonstrován obrázkem 2.5.



(a) Pole fotoaparátů.

(b) Portálové zařízení.

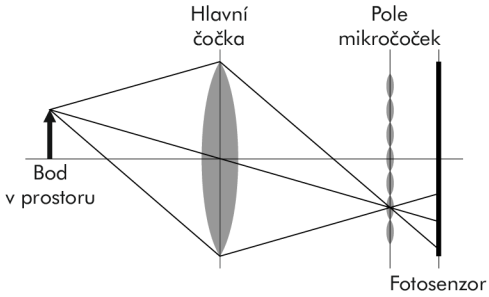
Obrázek 2.4: Zařízení zachycující plenoptické fotografie sestrojené na univerzitě Stanford.



Obrázek 2.5: Zachycení plenoptické fotografie si lze představit jako zachycení scény polem dírkových komor sežazených v rovině (x, y).

Druhý přístup naopak zaznamenává menší rozsah prostorových dimenzí (x, y). Tento přístup se uplatňuje v menších zařízeních, kde všechny pohledy snímá jeden fotosenzor. Takové zařízení se příliš neliší od klasického fotoaparátu. Hlavním rozdílem je pole mikročoček umístěné mezi hlavní čočkou a fotosenzorem. První plenoptický fotoaparát dostupný široké veřejnosti Lytro je na obrázku 2.6a Princip plenoptického fotoaparátu založeného na poli mikročoček je demonstrován na obrázku 2.6b.

(a) První plenoptický fotoaparát založený na poli mikročoček, který byl dostupný široké veřejnosti, Lytro.



(b) Princip plenoptického fotoaparátu. Paprsky směřující do fotoaparátu jsou před zachycením na světlocitlivý senzor rozptýleny polem mikročoček.

Obrázek 2.6: Plenoptický fotoaparát, který funguje na principu pole mikročoček.

2.5 Formát plenoptických fotografií

Plenoptická data můžou mít různé formáty a každý formát je vhodný pro jiné využití. Na jednu stranu je potřeba s těmito daty pracovat rychle a efektivně, na stranu druhou se musí řešit paměťové nároky na uložení těchto dat a s tím související komprese.

Při práci s plenoptickým fotoaparátem se lze nejčastěji setkat s daty v lenslet[1] formátu. Jde o surová data, která jsou zachycena světlocitlivým senzorem v plenoptickém fotoaparátu. Tato data reflektují rozložení mikročoček ve fotoaparátu. Pro jejich zpracování jsou k samotné bitmapě přiložena metadata o konstrukci fotoaparátu a světelných podmínkách při zachycení. Mimo metadata, která ukládá klasický fotoaparát, se zde zaznamenávají zejména kalibrační údaje k poli mikročoček.

Alternativním formátem je dvourozměrné pole pohledů na scénu, kde každý pohled reprezentuje scénu z jiného bodu pozorování. Tento formát je obvykle výstupem zachycení plenoptické fotografie pomocí pole fotoaparátů. Jako metadata jsou v tomto případě uloženy koordináty jednotlivých pohledů, a to buď ve formě vertikálního a horizontálního indexu, nebo pomocí jednotek reálného světa. Druhá varianta umožňuje určit měřítko zachycené scény a tím i rozměry objektů, které se ve scéně nacházejí. Tento formát je vhodný zejména pro kompresi založenou na 4D transformaci. Příznivým faktorem pro tuto třídu kompresních algoritmů je silná korelace mezi sousedícími pixely uvnitř jednotlivých pohledů a zároveň mezi sousedícími pohledy. Proto se plenoptické fotografie do tohoto formátu často převádí.

[1] Lenslet – mikročočka, která je prvkem pole mikročoček.

Kapitola 3

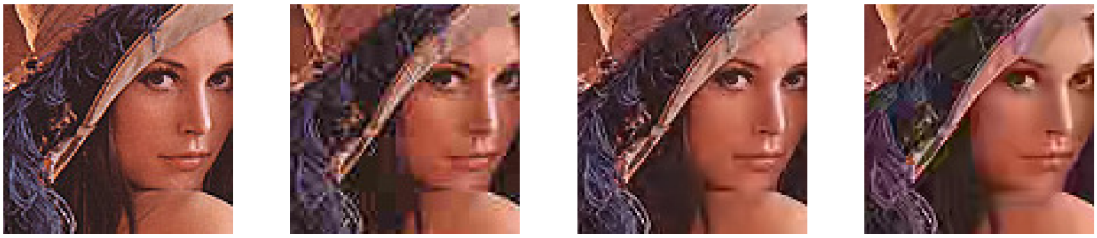
Ztrátová komprese

Komprese je ztrátová, pokud při ní dochází k nezvratné ztrátě informací. Rekonstruovaná data tedy pouze aproximují data původní. Výhodou dobře navržené ztrátové komprese oproti bezeztrátové kompresi je signifikantní nárůst kompresního výkonu na úkor často zanedbatelné degradace dat. Díky nedokonalosti lidských smyslů je ztrátová komprese hojně využívaná v oblasti obrazových a zvukových signálů.

V sekci 3.1 jsou popsány metody pro ztrátovou kompresi obrazu. Sekce 3.2 popisuje metody pro ztrátovou kompresi videa. V poslední sekci 3.3 je podrobně popsán kompresní řetězec metody JPEG, na kterém je založeno řešení pro plenoptické fotografie navržené v této práci.

3.1 Ztrátová komprese obrazu

Při ztrátové kompresi obrazu je využíváno zejména toho, že lidské oko je méně citlivé na barvu než na jas a zároveň je méně citlivé na vysoké frekvence v obraze. Při ztrátové kompresi dochází ke ztrátě právě těchto informací [2]. Na obrázku 3.1 jsou zobrazeny artefakty, které vznikají při ztrátové kompresi různými metodami.



(a) Originál. (b) JPEG. (c) JPEG 2000. (d) HEVC intra.

Obrázek 3.1: Porovnání artefaktů kompresních metod při nízkých datových tocích.

Komprese obvykle začíná převodem RGB dat do barevného modelu, který odděluje složku jasu (chroma) a barevného odstínu (luma). Pokud jsou vstupem kompresoru data ve stupních šedi, převod se neprovádí, stupně šedi již představují složku jasu. Komponenty barevné složky jsou volitelně podvzorkovány dle požadavků na výslednou kvalitu. Na každou z těchto komponent je aplikována transformace, která odděluje důležité informace (nízké frekvence) od méně důležitých (vysoké frekvence). Na méně důležitá data jsou aplikovány ztráty dle požadavků na výslednou kvalitu nebo kompresní poměr. Koeficienty jsou dále bezeztrátově kódovány entropickými kodéry.

Nejrozšířenějším standardem pro ztrátovou kompresi obrazu se stala metoda JPEG z roku 1992. Tato metoda je založená na diskrétní kosinové transformaci (DCT) bloků obrazu o 8×8 vzorcích. Koeficienty transformace jsou kvantizovány maticemi založenými na vlastnostech lidského oka. Na stejnosměrnou složku (DC) je aplikována diferenční pulzní kódová modulace (DPCM), pro měnící se složku (AC) se provádí run-length kódování (RLE). Vše završuje Huffmanovo nebo aritmetické kódování. Metodě JPEG se podrobně věnuje podkapitola 3.3. Na této metodě je také založen kompresní řetězec pro plenoptické fotografie, který je podrobně popsán v kapitole 4.

Pomyslným nástupcem formátu JPEG, který se ovšem svému předchůdci popularitou nikdy nevyrovnal, je metoda JPEG 2000. Tato metoda je založena na diskrétní vlnkové transformaci (DWT). Tato transformace rozděluje signál na dva podsignály reprezentující vysoké a nízké frekvence. Diskrétní vlnková transformace se dále rekurzivně aplikuje na jeden z nich. JPEG 2000 je v mnoha ohledech lepší než jeho předchůdce, ale kvůli patentům se nikdy nerozšířil. Zajímavostí je, že JPEG 2000 podporuje také mód bezeztrátové komprese.

Za zmínku také stojí HEVC intra kódování, veřejnosti známé jako nedílná součást metody pro kódování videa H.265 nebo obrázek v kontejneru HEIC. Tato metoda provádí predikce, které obraz efektivně dekorelují před tím, než jsou aplikovány kompresní metody podobné metodám již zmíněným. Ačkoli aktuálně patří HEVC intra mezi jednu z nejefektivnějších metod pro kódování obrazu, některé její technologie jsou pokryty patenty. Nahrazení metody JPEG širokou veřejností se proto stále nekoná.

3.2 Ztrátová komprese videa

Pro kompresi videa je klíčovým faktorem předpoklad, že jednotlivé snímky se od sebe moc neliší. Této vlastnosti se říká časová redundance. To je kombinováno s předpokladem, že hodnoty sousedících vzorků uvnitř jednoho snímku jsou také velmi podobné. Této vlastnosti se říká prostorová redundance. Stejně jako u komprese obrazu jsou pro ztrátovou kompresi videa využity nedokonalosti lidského vidění. Komprese videa kombinuje přístupy využití časové redundance (inter snímkové kódování) a prostorové redundance (intra snímkové kódování) [10].

Intra snímkové kódování využívá pro kompresi prostorové redundance. Intra snímkové kódování odpovídá metodám pro kompresi obrazu popsaným v kapitole 3.1. Existují dva přístupy pro kódování snímků. Transformační kódování na snímek aplikuje transformaci, která oddělí důležité informace od méně důležitých. Méně důležité informace je možné redukovat bez viditelné ztráty kvality, proto je využíváno zejména pro ztrátovou kompresi. Druhým přístupem je prediktivní kódování. Prediktivní kódování se snaží předpovědět hodnotu aktuálního vzorku na základě hodnot vzorků v jeho okolí. Tento rozdíl je entropicky kódován. Prediktivní kódování se využívá zejména v bezeztrátové kompresi.

Inter snímkové kódování využívá pro kompresi časové redundance. Vychází z předpokladu, že jednotlivé snímky v přirozeném videu se jen málo liší. Proto se místo samotných snímků kóduje pouze porovnání vůči snímkům předchozím, případně následujícím. Tomuto procesu se říká predikce. Existuje mnoho způsobů, jak predikce provádět, přičemž každý se liší svojí efektivitou a složitostí. Nejméně složitý způsob je pracovat s rozdílem aktuálního snímku vůči snímku předchozímu. Toto kódování ovšem není moc efektivní. Mnohem lepší výsledky nesou metody kompenzace pohybu. Ty můžou být prováděny na úrovni celého obrázku, nebo jeho částí v podobě bloků. Tyto části navíc můžou mít různou velikost v závislosti na komplexitě scény. Úlohou kompenzace pohybu je najít v odkazovaných

snímcích části podobné právě kompenzovanému bloku, vybrat tu s nejmenším rozdílem a kompenzovaný blok poté vyjádřit pomocí pohybového vektoru a rozdílu vůči nalezené části. Pro zvýšení přesnosti se kompenzace pohybu provádí na subpixelové úrovni. Jelikož by prohledávání celého obrázku vedlo k vysoké časové složitosti, prohledává se pouze okolí původního bloku. Existují optimalizační techniky s mnohonásobně menší složitostí, ovšem za cenu suboptimálního řešení.

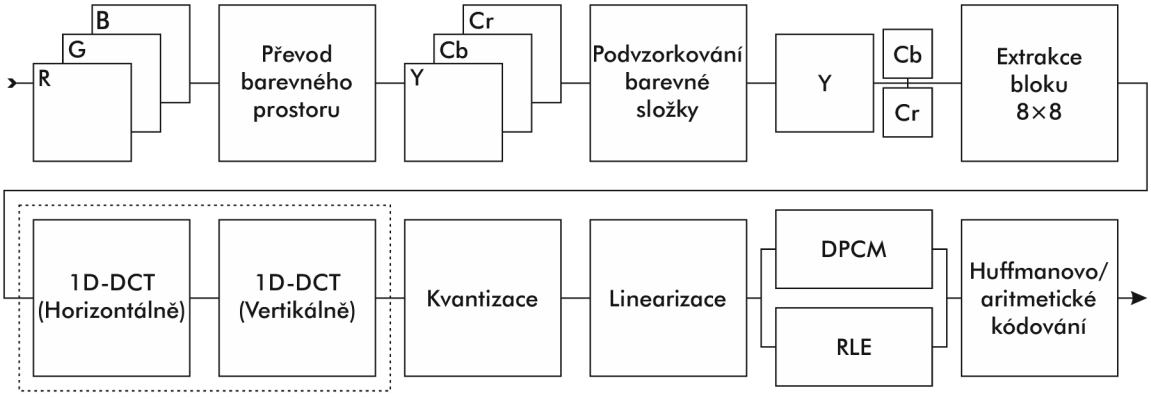
Tyto přístupy jsou ve videu kombinovány s cílem dosažení optimální komprese. Existuje několik druhů snímků, které tyto přístupy využívají. Popis typů snímků je dostupný v tabulce 3.1. Snímky můžou být ve výsledném streamu přeskupeny tak, aby každý snímek referoval vždy jen na snímky předcházející. To je výhodné z hlediska dekomprese. Pořadí snímků je při dekompresi navráceno do původní podoby.

Tabulka 3.1: Typy snímků využívané v kompresi videa.

Typ	Popis
I	Klíčové snímky, které jsou intra kódovány. Tyto snímky jsou nezávislé na jiných snímcích, lze je proto dekódovat samostatně.
P	Snímky inter kódovány v závislosti na předcházejících I nebo P snímcích.
B	Snímky inter kódovány v závislosti na budoucích i předcházejících I, P a B snímcích.

3.3 Kompresní řetězec metody JPEG

V této podkapitole je popsán kompresní řetězec metody JPEG. Metoda JPEG vznikla roku 1992 jako doporučení ITU-T T.81, od té doby se drží na příčce nejpopulárnější metody pro kompresi obrazu. Schéma kompresního řetězce metody JPEG je na obrázku 3.2.



Obrázek 3.2: Schéma kompresního řetězce metody JPEG. Vstupní RGB obrázek je převeden do barevného prostoru YCbCr. V závislosti na požadované kvalitě jsou podvzorkovány kanály reprezentující barevnou složku. Každý kanál je zpracováván samostatně. Komponenta je rozdělena na bloky 8×8. Na tyto bloky je aplikována 2D-DCT. koeficienty v bloku jsou kvantizovány kvantizační maticí a linearizovány do jednorozměrné struktury. Na DC koeficient je aplikována diferenční pulzní kódová modulace, na AC koeficienty se aplikuje run-length kódování. Posledním krokem je Huffmanovo či aritmetické kódování.

Transformace barevného prostoru

Pokud je vstupem komprese RGB obrázek, je před transformací převeden do barevného prostoru, který odděluje složku jasu od složek barevného odstínu. K tomuto se vybízí barevný prostor YCbCr, který data dělí na jednu složku luminance Y (jas) a dvě složky chrominance CbCr (barevný odstín). Jelikož je lidské oko citlivé více na jas než na barvu, je možné chromatické složky komprimovat více drasticky bez viditelné ztráty kvality. Rovnice

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0{,}299 & 0{,}587 & 0{,}114 \\ -0{,}168736 & -0{,}331264 & 0{,}5 \\ 0{,}5 & -0{,}418688 & -0{,}081312 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.1}$$

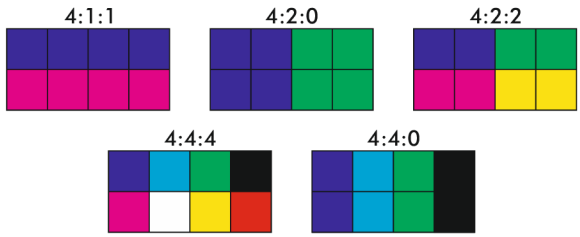
specifikuje výpočet hodnot YCbCr z hodnot RGB použitý v metodě JPEG. Rozsah vstupních hodnot RGB je z intervalu $\langle 0; 255 \rangle$, rozsah výstupního hodnot YCbCr je také z intervalu $\langle 0; 255 \rangle$. Po převodu do vhodného barevného prostoru je každý kanál komprimován zvlášť.

Posun hodnot do znaménkové reprezentace

Pro výpočet diskrétní kosinové transformace je vhodné, aby byly hodnoty vstupních vzorků posunuty do znaménkové reprezetace. To je provedeno odečtením hodnoty 2^{P-1} od hodnoty vzorku, kde P je bitová hloubka vzorku. Pro obrázky s 8 bity na kanál je od vzorků odečtena hodnota $2^{8-1} = 128$. Tento krok nemá vliv na hodnoty AC koeficientů, pouze způsobí posun DC koeficientu do znamenénkové reprezentace. Tento krok je v metodě JPEG zaveden zejména kvůli minimalizaci zaokrouhlovací a kumulační chyby, která může při výpočtu DCT nastat. DC koeficient lze posunout až po transformaci, nebo nemusí být posunut vůbec. V takovém případě nese DC koeficient oproti znaménkovým AC koeficientům bezeznaménkovou hodnotu.

Podvzorkování barevné složky

Lidské oko je více citlivé na jas než na barevný odstín. Informaci o barvě je proto možné redukovat bez viditelné ztráty kvality. Toho je využito ve většině kompresních metod v kroku zvaném podvzorkování barevné složky. V kompresním řetězci JPEG jsou nejčastěji využívány tři módy podvzorkování. Mód 4:2:0 redukuje barevnou složku na polovinu v horizontálním i vertikálním směru. To znamená, že na na jeden vzorek barevné složky (Cb a Cr) připadnou čtyři vzorky jasu (Y). Mód 4:2:2 redukuje barevnou složku na polovinu pouze ve vertikálním směru, na každý vzorek barvy proto připadnou dva vzorky jasu. Mód 4:4:4 neredukuje barevnou složku vůbec, ke každému vzorku barvy tak přísluší jeden vzorek jasu. Demonstrace módů podvzorkování barevné složky je na obrázku 3.3.



Obrázek 3.3: Demonstrace módů podvzorkování barevné složky.

Dělení do bloků

Data jednoho komprimovaného kanálu jsou převedeny do bloků o 8×8 sousedících vzorcích, které jsou následně komprimovány individuálně. U obrázků s rozměry beze zbytku nedělitelnými osmi vzniká problém krajních bloků. Obrázek se v tomto případě rozšíří na nejbližší rozměry, jež jsou osmi beze zbytku dělitelné. Existuje několik metod, jak vyplnit nově vzniklé krajní vzorky. Naivně lze vyplnit prázdné místo jednotnou hodnotou, např. nulami. Nevýhodou tohoto přístupu jsou vysoké frekvence, které vznikají skokovým přechodem na okraji obrázku. Tento problém dokáže částečně řešit přístup, kdy se vzorky doplní hodnotou nejbližších krajních vzorků původního obrázku. Alternativou může být také zrcadlení původního obrázku nebo jeho opakování – svinutí obrázku do pomyslného torusu.

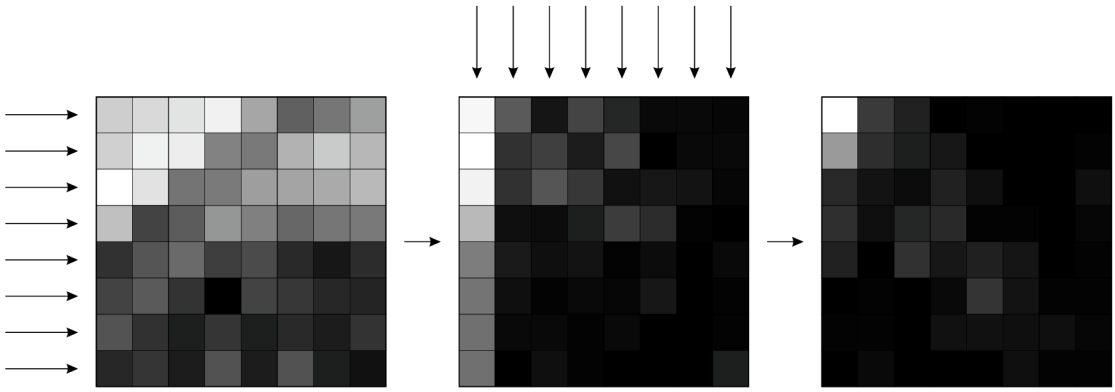
2D-DCT

Na každý blok se aplikuje diskrétní kosinová transformace (DCT). Dopředná DCT rozloží blok na DC koeficient reprezentující průměrnou hodnotu vzorků v bloku, a AC koeficienty reprezentující zastoupení jednotlivých frekvencí v bloku. Jde o transformaci podobnou diskrétní Fourierově transformaci, na rozdíl od ní ale produkuje pouze reálné koeficienty. Diskrétní kosinová transformace je využívána díky její vlastnosti koncentrovat energii poblíž DC koeficientu, což je z hlediska ztrátové komprese výhodné. Dvourozměrnou diskrétní kosinovou transformaci lze vyjádřit separabilně pomocí sady jednorozměrných DCT, která je specifikována rovnicí

$$X_k = \sqrt{\frac{2}{N}} c_k \sum_{n=0}^{N-1} x_n \cos\left[\frac{(2n+1)\,k\,\pi}{2N}\right] \qquad (3.2)$$

$$c_k = \begin{cases} 1/\sqrt{2} & \text{pro } k\ = 0 \\ 1 & \text{jinak} \end{cases}$$

Nejprve se 1D-DCT aplikuje na každý řádek vzorků, poté na každý sloupec koeficientů vycházejících z předchozího kroku. Tento postup je demonstrován na obrázku 3.4.



Obrázek 3.4: Separabilní provedení 2D-DCT. V prvním průchodu je 1D-DCT provedena na všechny řádky v bloku. V druhém průchodu je provedena na všechny sloupce. Výsledkem je blok koeficientů 2D-DCT.

Kvantizace

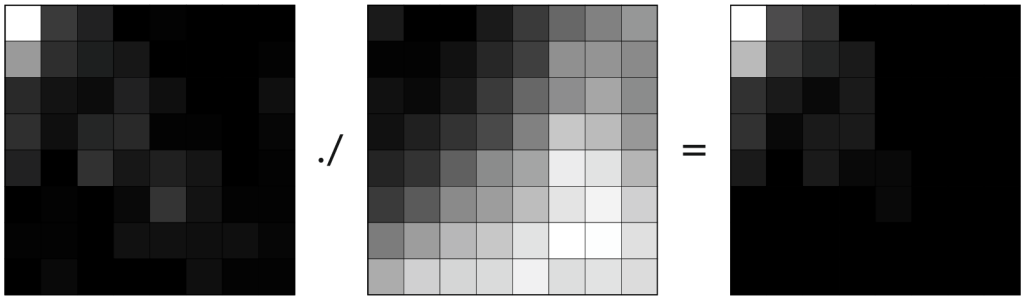
Transformovaný blok se dále kvantizuje dle požadavků na výslednou kvalitu. Ke kvantizaci se využívá kvantizační matice. Příkladem je kvantizační matice

$$Q_{\text{luma}} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \tag{3.3}$$

Tato matice byla vytvořena na základě experimentů zkoumajících vlastnosti lidského oka. Využívá se faktu, že lidské oko je méně citlivé na vysoké frekvence, které je možné kvantizovat více, než frekvence nízké. Kvantizace probíhá podělením DCT koeficientů v bloku příslušející hodnotou v kvantizační matici a následným zaokrouhlením podle rovnice

$$X_{q\mathbf{k}} = \text{round}\left[\frac{X_{\mathbf{k}}}{Q_{\mathbf{k}}}\right] \tag{3.4}$$

Krok kvantizace je vizuálně demonstrován na obrázku 3.5.



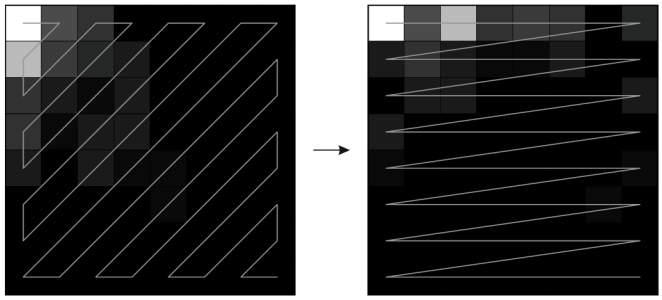
Obrázek 3.5: Kvantizace bloku koeficientů. Koeficienty jsou děleny hodnotami kvantizační matice prvek po prvku.

Linearizace

Dále je potřeba koeficienty v bloku seřadit tak, aby vznikly shluky, které je dále možné efektivně entropicky kódovat. K tomu se využívá průchod způsobem cik-cak podle matice

$$\begin{bmatrix} 0 & 1 & 5 & 6 & 14 & 15 & 27 & 28 \\ 2 & 4 & 7 & 13 & 16 & 26 & 29 & 42 \\ 3 & 8 & 12 & 17 & 25 & 30 & 41 & 43 \\ 9 & 11 & 18 & 24 & 31 & 40 & 44 & 53 \\ 10 & 19 & 23 & 32 & 39 & 45 & 52 & 54 \\ 20 & 22 & 33 & 38 & 46 & 51 & 55 & 60 \\ 21 & 34 & 37 & 47 & 50 & 56 & 59 & 61 \\ 35 & 36 & 48 & 49 & 57 & 58 & 62 & 63 \end{bmatrix} \tag{3.5}$$

Tento průchod zajišťuje, že koeficienty zastupující nejnižší frekvence budou koncentrovány na jednom konci jednorozměrné struktury a naopak koeficienty zastupující nejvyšší frekvence jsou seřazeny na konci druhém. Jelikož byly vysoké frekvence v předchozím kroku redukovány, vzniknou na jednom konci struktury shluky nul, které můžou být efektivně komprimovány v dalším kroku. Obrázek 3.6 demonstruje linearizaci na bloku kvantizovaných koeficientů.



Obrázek 3.6: Demonstrace linearizace koeficientů. Na vstupu je blok kvantizovaných koeficientů. Na výstupu jsou koeficienty přeskupeny tak, aby nenulové koeficienty zůstaly ve shluku i v jednorozměrné struktuře.

Diferenční pulzní kódová modulace DC koeficientů

Na DC koeficienty je aplikováno kódování, kdy se namísto hodnoty dále pracuje s rozdílem hodnoty aktuálního DC koeficientu a hodnoty DC koeficientu bloku předchozího. Tomuto kódování se říká diferenční pulzní kódová modulace (DPCM). Při tomto kódování se využívá toho, že DC koeficient vyjadřuje průměrnou hodnotu původního bloku. Bloky, které spolu sousedí, mají v běžných fotografiích podobnou průměrnou hodnotu. Vzniká proto redundance, které je při tomto kódování využito.

Run-length kódování AC koeficientů

Seřazené AC koeficienty v bloku jsou zakódovány tak, že každý nenulový koeficient je převeden na dvojici, kde první hodnota určuje počet nul, které koeficientu předcházely, a druhá hodnota je samotný koeficient. Tomuto kódování se říká run-length kódování (RLE).

Toto kódování zahrnuje dva speciální symboly. Prvním speciálním symbolem je značení, že veškeré zbylé koeficienty v bloku jsou nulové. Tento symbol se nazývá end-of-block (EOB) a je značen jako dvojice nul. Druhý symbol řeší situaci, kdy je před kódovaným nenulovým koeficientem více než 15 nul. V tomto případě je nutné zavést opatření, které zajistí, že hodnotu počtu nul bude možné uložit do 4 bitů. Druhý speciální symbol proto reprezentuje šestnáct nulových koeficientů za sebou. Tato dvojice se značí jako dvojice (15, 0), tedy patnáct nul předcházejících koeficient s nulovou hodnotou. Pokud se na vstupu objeví např. sedmnáct nul a dvojka, budou výstupem dvojice (15, 0) a (1, 2).

Entropické kódování

Entropické kódování je metoda bezeztrátové komprese, při které se sekvence symbolů o pevné délce kóduje do sekvence s proměnnou délkou v závislosti na frekvenci výskytů vstupních symbolů. Sekvence jsou voleny tak, že symbol s nejčastějším výskytem je reprezentován nejkratší sekvencí. Metoda JPEG umožňuje dvě metody entropického kódování.

První metodou je výpočetně nenáročné Huffmanovo kódování, druhou metodou je mírně náročnější aritmetické kódování, které ovšem dosahuje lepších výsledků.

Při Huffmanově kódování se vstupní symboly o pevné délce mapují na výstupní bitové sekvence o variabilní délce. Tyto dvojice jsou uloženy v kódové knize. Pro kódování může být využita buď existující kódová kniha, nebo je možné vypočítat optimální kódovou knihu ze vstupního obrázku. Optimální kódová kniha vede k lepšímu kompresnímu výkonu za cenu jednoho průchodu obrázkem navíc pro získání frekvencí jednotlivých symbolů. V metodě JPEG jsou vstupem do Huffmanova kodéru osmibitové symboly, ve kterých horní 4 bity reprezentují počet nul získaný z předchozího RLE a dolní 4 bity obsahují hodnotu minimálního počtu bitů, do kterých lze uložit hodnotu amplitudy. Samotná amplituda není entropicky kódovaná, ale ukládá se jen na zakódovaný potřebný počet bitů. Záporné hodnoty amplitudy se ukládají v jedničkovém doplňku.

V metodě JPEG se obvykle používají zvlášť Huffmanovy tabulky pro AC a DC koeficienty a zvlášť pro jasovou a barevnou složku. Celkem tedy (AC, DC)×(luma, chroma) čtyři Huffmanovy tabulky. Tyto tabulky jsou uloženy ve výsledném souboru. Komprese Huffmanovy tabulky je možná při použití kanonického Huffmanova kódování.

Aritmetické kódováni oproti Huffmanovu kódování nekóduje symboly do bitových sekvencí samostatně, ale kóduje zprávu jako celek. Funguje na principu dělení intervalu na menší intervaly, kdy rozsah každého intervalu poměrně odpovídá frekvenci výskytu kódovaného symbolu. Začíná se dělením intervalu $\langle 0, 1)$. Dále se vždy dělí interval odpovídající zakódovanému symbolu. Aritmetické kódování dosahuje mírně lepšího kompresního výkonu, ovšem za cenu větší výpočetní náročnosti, proto není tak často využívané v praxi.

Kapitola 4

Kompresní řešení pro plenoptické fotografie založené na 4D-DCT

V této kapitole je představena navržená kompresní metoda pro plenoptické fotografie založená na 4D-DCT, která úzce zrcadlí metodu JPEG. Jsou zde řešeny problémy, které přináší komprese 4D dat. Dále jsou zde představeny nové myšlenky a rozšíření původní metody JPEG.

Sekce 4.1 popisuje navržený kompresní řetězec. Text je zaměřen zejména na rozdíly vůči původní metodě JPEG, ze které toto řešení vychází. V sekci 4.2 je popsána programová knihova v jazyce C++, která navrženou metodu implementuje. Zejména je zde kladen důraz na popis aplikačního rozhraní knihovny.

4.1 Popis kompresního řetězce

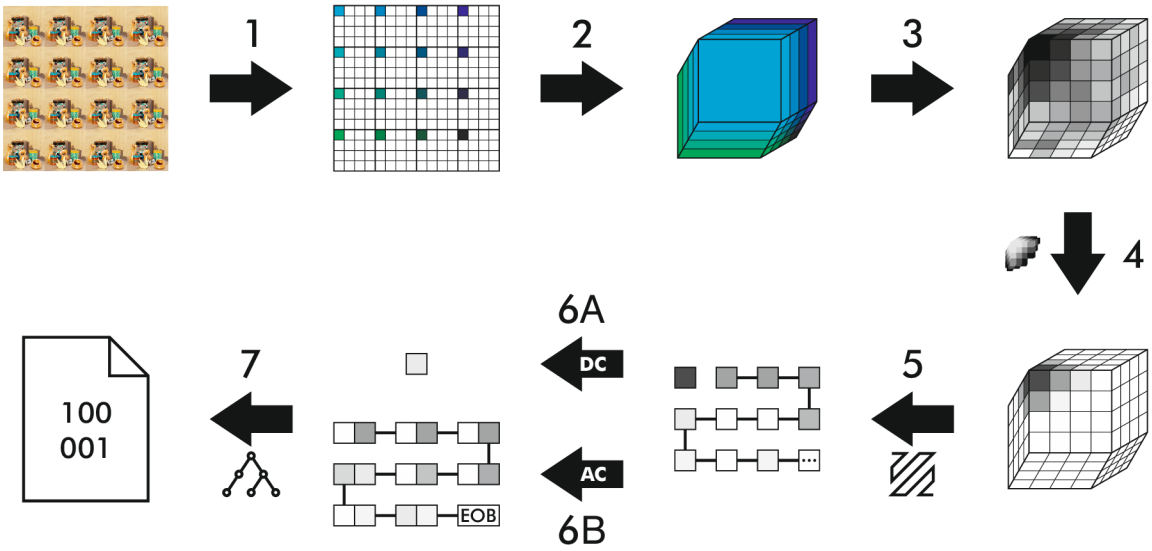
Kompresní řetězec je ilustrován na obrázku 4.1. Od metody JPEG se navržený řetězec liší pouze v detailech jednotlivých kroků, v této kapitole jsou proto popsány zejména tyto diference. Podrobný popis jednotlivých kroků metody JPEG je dostupný v kapitole 3.3.

Vstupní data

Vstupem kompresního řetězce je plenoptický obrázek reprezentován jednotlivými pohledy ve čtvercové mřížce. Obrázek je v barevném prostoru RGB, případně ve stupních šedi. Navržený kompresní řetězec zvládá oproti metodě JPEG komprimovat vstupní obrázek o libovolné bitové hloubce až do 16 bitů na kanál.

Konverze barevného prostoru

Prvním krokem je konverze barevného prostoru. Tento krok je identický s kompresním řetězcem metody JPEG. Obrázek je převeden do barevného prostoru, který odděluje složku jasu a barevného odstínu. Pro vstupní data ve stupních šedi se konverze neprovádí. Hodnoty vzroků jsou posunuty do znaménkové reprezentace odečtením 2^{P-1}, kde P je bitová hloubka vstupního obrázku v rozsahu $\langle 1; 16 \rangle$.



Obrázek 4.1: Schéma navrženého kompresního řetězce. 1) Vstupní obrázek je konvertován do vhodného barevného prostoru (YCbCr), vzorky jsou posunuty do znaménkové reprezentace. Řetězec kóduje každý kanál zvlášť. 2) Barevná komponenta je rozdělena na čtyřrozměrné bloky. Barevné čtverce v mřížce na levé straně tohoto kroku představují dvourozměrné bloky o velikosti 4×4 vzorků, kde každý blok přísluší stejné pozici v každém pohledu. Tyto 2D bloky jsou zkombinovány do jednoho 4D bloku. 3) Na blok je aplikována 4D-DCT. Energie je koncentrována v okolí DC koeficientu. 4) Blok je kvantizován čtyřrozměrnou kvantizační maticí. 5) Koeficienty jsou linearizovány do jednorozměrné struktury. 6A) Na DC koeficienty je aplikována diferenční pulzní kódová modulace. 6B) Na AC koeficienty je aplikováno run-length kódování. 7) DC koeficient a AC páry jsou pomocí Huffmanova kódování bezeztrátově zakódovány do výstupního bistreamu.

Dělení do bloků

Komprimovaná komponenta plenoptického obrázku je rozdělena na bloky o velikosti $N \times N \times N \times N$ vzorků, kde N je hodnota určená před samotnou kompresí a při kompresi se nemění. Pokud blok přesahuje rozměry dané komponenty, jsou přesahující vzorky doplněny nejbližšími hodnotami z původního obrázku.

Ačkoli se v metodě JPEG využívají pouze bloky o velikosti 8×8, pro plenoptické fotografie není pevně daná velikost bloku vhodná. Pokud je například obrázek o 9×9 pohledech, kde každý pohled má 512×512 vzorků, rozdělen na bloky o velikosti $N = 8$ vzorků, vznikne $\lceil 9/8 \rceil \times \lceil 9/8 \rceil \times \lceil 512/8 \rceil \times \lceil 512/8 \rceil = 16\,384$ bloků. Tyto bloky dohromady obsahují $16\,384 \times 8^4 = 67\,108\,864$ vzorků. Původní obrázek má $9^2 \times 512^2 = 21\,233\,664$ vzorků. To znamená, že jen 31,6 % vzorků jsou vzorky původní a zbylých 68,4 % jsou vzorky doplněné. Oproti tomu pro bloky o velikosti $N = 9$ vzorků bude komprimovaný počet bloků 3 249. Tyto bloky budou obsahovat 21 316 689 vzorků, což je jen 0,4 % doplněných vzorků. Volba velikosti bloku je proto velmi důležitá.

4D-DCT

Stejně jako u metody JPEG je na blok aplikována diskrétní kosinová transformace. Ta je v tomto případě čtyřrozměrná. Při separabilní implementaci to znamená, že se provede

$N^3 \times 4$ 1D-DCT. Společně s počtem dimenzí bloku se oproti metodě JPEG mění také maximální amplituda výsledných koeficientů. Ta je navíc závislá i na velikosti bloku. Vzorec pro výpočet minimálního počtu bitů, do kterých lze uložit nejvyšší koeficient, vychází ze samotné kosinové transformace a je definován rovnicí

$$\text{depth}_{\text{DCT}} = \left\lceil \log_2(N^D) \right\rceil + P - (3D/2); \tag{4.1}$$

kde N je velikost hrany bloku, D je počet dimenzí a P je bitová hloubka vstupních vzorků. Tento vzorec předpokládá, že vstupní vzorky byly posunuty do znamenénkové reprezentace. Znalost velikosti koeficientu transformace pro dané parametry je klíčová v následujících krocích.

Kvantizace

Stejně jako v metodě JPEG jsou koeficienty kosinové transformace kvantizovány. Jelikož jsou bloky koeficientů čtyřrozměrné a o předem neznámých rozměrech, nelze přímo použít standardní dvourozměrné kvantizační matice. Proto je součástí této práce návrh 4D kvantizačních matic. Je kladen důraz zejména na způsoby rozšíření existujícíh matic na požadované rozměry.

Nejjednodušším možným schématem je uniformní kvantizace. Všechny koeficienty jsou kvantizovány stejnou hodnotou, která odpovídá požadované kvalitě komprese. Další schémata jsou založená na rozšiřování dvourozměrných matic.

Pro rozšíření matice na požadované rozměry jsou navrženy dva přístupy. První přístup rozšířené koeficienty doplňuje nejbližší hodnotou z původní kvantovací matice. Pokud je prováděno zmenšení matice, jsou přebytečné koeficienty ořezány. Druhým přístupem je roztažení matice pomocí diskrétní kosinové transformace. Matice je transformována do frekvenční domény, kde je následně rozšířena a doplněna nulami. Poté je zpětně transformována pomocí inverzní kosinové transformace. Rozšířená matice má požadovaný rozměr a podobné rozložení koeficientů, jako matice původní.

Pro rozšíření matice na požadovaný počet dimenzí jsou navrženy také dva přístupy. V prvním přístupu se provede kopie dvourozměrné matice do čtyřrozměrného bloku. Druhý přístup kvantovací matici vytváří z průměrných hodnot diagonál původní kvantovací matice. Ve dvourozměrné matici se vypočtou průměrné hodnoty diagonál, tyto hodnoty se použijí na příslušející diagonály ve 4D matici. Jelikož 4D matice obsahuje více diagonál než 2D matice, jsou zbývající diagonály doplněny hodnotou největší dostupné diagonály.

Linearizace

Po kroku kvantizace je nutné koeficienty v bloku vhodně seřadit do jednorozměrné sekvence, kterou bude dále možné bezeztrátově komprimovat. V metodě JPEG se provádí cik-cak průchod. V rámci této práce bylo navrženo několik variant průchodu 4D bloku. Jedna z těchto variant pracuje s průchodem bloku metody JPEG. Další varianty blok procházejí na základě nějakého pravidla či vzorce. Poslední variantou je průchod založený na průměrném rozložení koeficientů v blocích.

Varianta přímo rozšiřující metodu JPEG je čtyřrozměrný cik-cak průchod. Cik-cak průchod pro blok o D dimenzích lze vyjádřit rekurzivně jako sérii navazujících cik-cak průchodů o $D - 1$ dimenzích provedených nad každou diagonálou. Algoritmus, který toto implementuje, je založený na obecném nenavazujícím průchodu diagonál, který je kombinován s rotacemi linearizovaného bloku. Tato kombinace zajistí, že jednotlivé diagonály na sebe navazují a výsledný průchod odpovídá průchodu cik-cak, který je známý ze dvou dimenzí.

V této práci bylo experimentováno s dalšími metodami průchodu bloku. První testovanou variantou je průchod bloku podle Eukleidovské vzdálenosti od DC koeficientu. To odpovídá průchodu koeficientů po kružnicích se středem v DC koeficientu. Nevýhodou této metody je, že mezi koeficienty vznikají skoky, průchod není spojitý, což není pro linearizaci výhodné. Další testovanou metodou je obecný průchod po diagonálách. Tento průchod je jednodušší variantou cik-cak průchodu. Tento průchod také trpí nespojistostí, ta je však menší než u průchodu podle Eukleidovské vzdálenosti. Další variantou je průchod založený na kvantizační matici. Vychází z předpokladu, že v místech, kde je hodnota kvantizační matice nejnižší, bude výsledný koeficient nejvyšší a naopak. Tato varianta je závislá na zvolené kvantizační matici, např. uniformní kvantizační matice by byla pro tento způsob linearizace naprosto nevhodná.

Alternativou je průchod, který bere v potaz rozložení koeficientů v právě komprimovaném obraze. Pro komponentu jsou spočteny průměrné hodnoty koeficientů v blocích, blok je poté linearizován na základě těchto průměrných hodnot tak, aby největší koeficienty připadly na začátek struktury a nejmenší hodnoty na konec. Tento průchod většinou není spojitý, ale jelikož je známo průměrné rozložení koeficientů, nepředstavuje to takový problém.

Diferenční pulzní kódová modulace DC koeficientů

DC koeficienty jsou kódovány stejně jako v metodě JPEG, tedy jako rozdíl DC koeficientu bloku aktuálního vůči DC koeficientu bloku předchozího ze stejné komponenty. S výslednou hodnotou je dále pracováno při Huffmanově kódování.

Run-length kódování AC koeficientů

Run-length kódování odpovídá metodě JPEG. Koeficienty jsou seskupovány do dvojic, kde prvním prvkem dvojice je počet nul před nenulovým koeficientem a druhým prvkem dvojice je daný nenulový koeficient. Taktéž je zde uplatňována speciální dvojice EOB, která značí, že veškeré koeficienty do konce bloku jsou nulové.

Rozdíl nastává u speciální dvojice pro vyjádření dlouhé sekvence následujících nul. V metodě JPEG jsou pro tuto hodnotu vyhrazeny 4 bity, takže maximální délka sekvence nul je 15. V navržené metodě je maximální délka sekvence nul zvolena dynamicky v závislosi na maximální možné hodnotě diskrétní kosinové transformace pro danou velikost bloku N, bitové hloubce P a zvolené velikosti kódového symbolu při Huffmanově kódování, které následuje po run-length kódování.

Huffmanovo kódování

posledním krokem, který zpracované koeficienty převede do výsledného bitstreamu, je Huffmanovo kódování. Kódování v navrženém postupu se od postupu v metodě JPEG liší pouze velikostí kódovaného symbolu.

Metoda JPEG kóduje symboly o velikosti 8 bitů, kde horní čtyři bity připadají na počet nul před nenulovým koeficientem a dolní čtyři bity jsou vyhrazeny na velikost tohoto koeficientu. To je dáno tím, že maximální koeficient diskrétní kosinové transformace v metodě JPEG lze podle vzorce (4.1) uložit minimálně do 11 bitů. Tuto hodnotu lze uložit v symbolu minimálně do $\lceil \log_2 (11 + 1) \rceil = 4$ bitů.

Pomocí stejného postupu lze určit rozdělení bitů v symbolu pro čtyřrozměrnou transformaci s danými parametry. Uvažujme kódování obrázku s bitovou hloubkou $P = 12$ bitů

na kanál s bloky o velikosti $N = 15$ pomocí čtyřrozměrné kosinové transformace $D = 4$. Dle vzorce (4.1) je pro uložení maximální hodnoty 4D-DCT potřeba minimálně 22 bitů. Hodnotu 22 lze uložit minimálně do 5 bitů. To znamená, že pět bitů kódového symbolu bude sloužit k vyjádření velikosti amplitudy a zbytek pro vyjádření velikosti shluku nul. Při velikosti symbolu 8 bitů to znamená, že velikosti shluku nul přísluší pouze tři bity, maximálně tedy lze vyjádřt symbol který shlukuje $2^3 - 1 = 7$ nul. V kombinaci s kódovaným blokem, který obsahuje 15^4 koeficientů, není maximální délka shluku nul dostačující, protože dlouhé shluky nul se při run-length kódování namapují na sekvenci speciálních párů značících maximální shluk nul. Shluky nul různé délky v takovém případě není možné optimálně Huffmanově kódovat.

Proto je v navrhovaném řešení použita velikost kódového slova 16 bitů. Jelikož velikosti amplitudy přísluší 5 bitů, na velikost shluku zůstane dostupných 11 bitů. To znamená maximální délku shluku nul 2047, což je považováno za dostačující. Tato změna se projeví dvojnásobnou velikostí kódové knihy ve výstupním bitstreamu, která je však při využití kanonického Huffmanova kódování zanedbatelná.

4.2 Implementace

Navržený kompresní řetězec byl implementován jako programová knihovna v jazyce C++. Knihovna se skládá z řady modulů, tyto moduly jsou popsány v tabulce 4.1.

Tabulka 4.1: Moduly implementované knihovny.

Název	**Popis**
bitstream	Čtení a zápis v jednotkách bitů.
encoder/decoder	Kroky komprese/dekomprese.
block	Extrakce/vložení bloku vzorků ze/do souvislé paměti.
colorspace	Převod mezi barevnými prostory.
constpow	Pomocná funkce pro výpočet compile-time mocniny.
dct	Výpočet diskrétní kosinové transformace.
endian	Funkce pro čtení/zápis hodnot, které provádí endian konverze.
huffman	Huffmanovo kódování.
quant table	Generování kvantizačních matic.
runlength	Run-length kódování.
traversal	Generování linearizačních matic.
zigzag	Algoritmus pro generování cik-cak linearizační matice.

Drtivá většina těchto modulů využívá šablonového metaprogramování. Rozhraní této knihovny je také šablonové, část knihovny je proto kompilována se specifikovanými parametry až na straně klienta. Knihovna dokáže pracovat se sadou obrázků jako s jedním monolitickým n-dimenzionálním tělesem, nebo jako se sadou několika menších těles. Kompresi 4D light field fotografií je proto možné pomocí této knihovny provést hned několika způsoby.

Vstupem kodéru je funkční objekt, pomocí kterého je přistupováno k jednotlivým barevným vzorkům. Tento přístup umožňuje správu paměti na straně klienta. S tímto krokem souvisí

převod do vhodného barevného prostoru, který je prováděn na straně klienta. K převodu je možné využít modul colorspace, který je součástí knihovny. Kodér ke vzorkům přistupuje nezávisle na jejich bitové hloubce a uložení v paměti pomocí rozhraní definovaném tímto funkčním objektem.

Implementovaný kodér provádí dvou až tří průchodovou kompresi. Pokud chce klient použít linearizační matici založenou na průměrném rozložení koeficientů v blocích, musí provést průchod, který toto rozložení z obrázku získá. Pokud klientovi stačí cik-cak průchod, tento krok není potřeba. Klientovi je také umožněno do kodéru vložit vlastní linearizační matici, kterou má např. uchovanou z předchozí komprese. Průchod po zkonstruování linearizačních matic slouží k získání četností symbolů pro optimální Huffmanovo kódování. Klient má opět možnost tuto četnost dodat bez nutnosti provádění průchodu a tím rapidně zvýšit rychlost komprese. Veškeré tabulky, matice a informace o obrázku jsou před posledním průchodem zapsány do hlavičky výsledného streamu. Pro zajištění kompatibility jsou hodnoty ukládány ve formátu big endian. Poslední průchod kóduje obrázek do bitstreamu. Pro tento průchod je nutné mít sestrojenou linearizační matici a Huffmanovu kódovou knihu.

Dekodér provádí dekompresi v jednom průchodu. Veškeré informace potřebné k dekódování jsou přečteny z hlavičky vstupního streamu, přičemž je provedena konverze z big endian formátu do lokální endianity. Dále dekodér zpracovává a dekóduje vstupní bitstream. Dekodér vrací výstup pomocí funkčního objektu. Klient tak má možnost vzorky průběžně zpracovávat.

4.3 Aplikační rozhraní

Aplikační rozhraní knihovny sestává ze struktur pro uchování stavu a parametrů kodéru a dekodéru společně s řadou funkcí, které nad těmito strukturami pracují. Veškeré rozhraní přijímá dva šablonové parametry, první specifikuje velikost bloku, druhý specifikuje počet dimenzí. Šablonové parametry kombinují výhodu parametrizace kodéru/dekodéru na straně klienta společně s možností provedení pokročilých optimalizací na straně překladače.

Kodér

Aplikační rozhraní kodéru poskytuje hlavičkový soubor `lfif_encoder.h`.

```
template<size_t BS, size_t D>
struct LfifEncoder {
  uint8_t color_depth;
  uint64_t img_dims[D+1];

  ...
};
```

Struktura `LfifEncoder` slouží pro definici parametrů komprese. V této struktuře je také uchován stav kompresoru mezi jednotlivými průchody. Parametry šablony `<size_t BS, size_t D>` určují velikost hrany, resp. počet dimenzí kódovaného bloku. Položka `color_depth` určuje barevnou hloubku vstupního obrázku v jednotce počtu bitů na kanál.

Pole `img_dims` obsahuje rozměry komprimovaného obrázku. Pro kompresi plenoptického obrázku ve čtyřech dimenzích $D = 4$ odpovídají první dva prvky pole šířce, resp. výšce pohledu v počtu vzorků. Následující dva prvky odpovídají počtu pohledů v horizontálním, resp. vertikálním směru. Poslední prvek je určený pro počet komprimovaných obrázků. Tento prvek je například využitelný pro kompresi plenoptického obrázku ve dvou dimenzích $D = 2$. V takovém případě by první dva prvky odpovídaly opět šířce a výšce, poslední prvek by sloužil ke specifikaci počtu pohledů v obrázku. Pokud by byl plenoptický obrázek komprimován jako jeden velký 2D obrázek složený z jednotlivých pohledů, mohlo by dojít ke vzniku artefaktů na hranách pohledů z důvodu vysokých frekvencí. Parametr udávající počet obrázků slouží ke korektnímu oddělení jednotlivých pohledů.

Obecně pro každý komprimovaný obrázek musí vždy platit, že součin všech položek pole `img_dims` odpovídá počtu pixelů komprimovaného obrázku. Tyto první dvě položky struktury `LfifEncoder` musí být specifikovány klientem ještě před inicializací kodéru.

```
template<size_t BS, size_t D>
void initEncoder(LfifEncoder<BS, D> &enc);
```

Funkce `initEncoder()` provede inicializaci vnitřních struktur kodéru. V této fázi je mimo jiné vypočítáno rozdělení bitů v symbolu Huffmanova kódování. Tato funkce také inicializuje kodér pro kompresi nového obrázku, pokud byla struktura již použita.

```
template<size_t BS, size_t D>
int constructQuantizationTables(LfifEncoder<BS, D> &enc,
                                const std::string &table_type, float quality);
```

Funkce `constructQuantizationTables()` provede inicializaci kvantovacích matic obrázku. Parametr `table_type` určuje typ kvantizační matice, která bude vytvořena. Implementované typy kvantizačních matic jsou specifikovány v tabulce 4.2. Parametr `quality` je hodnota z intervalu $\langle 1{,}0; 100{,}0\rangle$. Podle této hodnoty jsou kvantovací matice škálovány na požadovanou kvalitu. Tato funkce vrací nenulovou hodnotu v případě, že byl funkci předán neznámý argument. Jinak funkce vrací nulu.

Tabulka 4.2: Typy kvantizačních matic podporované aktuální implementací.

Typ	Popis
`UNIFORM`	Uniformní kvantizace.
`DCTDIAG`	Rozšíření pomocí diskrétní kosinové transformace a diagonál.
`DCTCOPY`	Rozšíření pomocí diskrétní kosinové transformace a kopírování matice.
`FILLDIAG`	Rozšíření pomocí doplnění nejbližších hodnot a diagonál.
`FILLCOPY`	Rozšíření pomocí doplnění nejbližších hodnot a kopírování matice.
`DEFAULT`	Zvolí nejvhodnější typ kvantizace.

```
template<size_t BS, size_t D, typename F>
void referenceScan(LfifEncoder<BS, D> &enc, F &&input);
```

Funkce `referenceScan()` provede první průchod obrázkem a získá průměrné rozložení koeficientů v blocích. Tento průchod je nutný pouze v případě, že má klient v plánu použít optimalizovanou linearizační matici. Parametr `input` představuje funkční objekt, pomocí kterého je přistupováno k obrázku.

```
template<size_t BS, size_t D>
int constructTraversalTables(LfifEncoder<BS, D> &enc,
                             const std::string &table_type);
```

Funkce `constructTraversalTables()` zkonstruuje linearizační matici na základě parametru `table_type`. Implementované typy linearizačních matic jsou specifikovány v tabulce 4.3. Tato funkce vrací nenulovou hodnotu v případě, že byl funkci předán neznámý argument. Jinak funkce vrací nulu.

Tabulka 4.3: Typy linearizačních matic podporované aktuální implementací.

Typ	Popis
`REFERENCE`	Linearizace podle průměrného rozložení koeficentů.
`ZIGZAG`	Linearizace metodou cik-cak.
`QTABLE`	Linearizace podle kvantizační matice.
`RADIUS`	Linearizace podle rostoucí Eukleidovské vzdálenosti od DC koeficientu.
`DIAGONALS`	Linearizace po diagonálách vzdalujících se od DC koeficientu.
`DEFAULT`	Zvolí nejvhodnější typ linearizace.

```
template<size_t BS, size_t D, typename F>
void huffmanScan(LfifEncoder<BS, D> &enc, F &&input);
```

Funkce `huffmanScan()` provede průchod obrázkem, při kterém je získána četnost symbolů Huffmanova kódování. Parametr `input` představuje funkční objekt, pomocí kterého je přistupováno k obrázku.

```
template<size_t BS, size_t D>
void constructHuffmanTables(LfifEncoder<BS, D> &enc);
```

Funkce `constructHuffmanTables()` zkonstruuje Huffmanovy tabulky na základě četností získaných při průchodu `huffmanScan()`.

```
template<size_t BS, size_t D>
void writeHeader(LfifEncoder<BS, D> &enc, std::ostream &output);
```

Funkce `writeHeader()` do výstupního streamu `output` zapíše hlavičku komprimovaného obrázku. Tato hlavička obsahuje magic number, velikost bloku, rozměry obrázku, kvantovací matice, linearizační matice a Huffmanovu tabulku.

```
template<size_t BS, size_t D, typename F>
void outputScan(LfifEncoder<BS, D> &enc, F &&input, std::ostream &output);
```

Funkce `outputScan()` provede poslední průchod obrázkem. Parametr `input` představuje funkční objekt, pomocí kterého je přistupováno k obrázku. Tato funkce zapisuje zakódovaný obrázek do streamu `output`.

Dekodér

Aplikační rozhraní dekodéru poskytuje hlavičkový soubor `lfif_decoder.h`.

```
template<size_t BS, size_t D>
struct LfifDecoder {
  uint8_t color_depth;
  uint64_t img_dims[D+1];

  ...
};
```

Struktura `LfifDecoder` slouží pro uchování stavu dekomprese. Položky `color_depth` a `img_dims` jsou pomocí funkce `readHeader()` přečteny ze vstupního streamu a slouží klientovi k získání informací o komprimovaném obrázku před samotnou dekompresí. Klient tak má například možnost předem alokovat potřebné paměťové prostředky.

```
template<size_t BS, size_t D>
int readHeader(LfifDecoder<BS, D> &dec, std::istream &input);
```

Funkce `readHeader()` přečte ze streamu `input` hlavičku komprimovaného souboru. Tato hlavička obsahuje informaci zejména o barevné hloubce, rozměrech obrázku, dále obsahuje kvantizační a linearizačníí matice a informace pro Huffmanovo dekódování. Tato funkce vrací nenulovou hodnotu v případě, že hlavička obsahuje neočekávané symboly, jinak funkce vrací nulu.

```
template<size_t BS, size_t D>
void initDecoder(LfifDecoder<BS, D> &dec);
```

Funkce `initDecoder()` inicializuje dekodér na základě informací získaných z hlavičky komprimovaného souboru. V případě znovupoužití stejné struktury je dekodér reinicializován.

```
template<size_t BS, size_t D, typename F>
void decodeScan(LfifDecoder<BS, D> &dec, std::istream &input, F &&output);
```

Funkce `decodeScan()` dekóduje komprimovaný obrázek ze streamu `input` a pixely obrázku vrací pomocí funkčního objektu `output`.

Kapitola 5

Experimentální vyhodnocení

Součástí této práce je srovnání kompresních metod na plenoptických datech. V této kapitole jsou zdokumentovány experimenty provedené na obrázcích ze tří veřejně dostupných datasetů. Experimenty se věnují srovnání vlivu velikosti bloku, kvantizačních matic a linearizačních matic na kompresní výkon implementovaného řešení. Dále jsou provedeny experimenty srovnávající výkon implementovaného kodeku se třemi existujícími metodami pro kompresi obrazu a jednou metodou pro kompresi videa.

Sekce 5.1 popisuje dataset, na kterém bylo vyhodnocení prováděno. V sekci 5.2 jsou popsány kompresní metody, se kterými bylo experimentováno. Sekce 5.3 se věnuje metodice měření, zejména popisuje použitou metriku PSNR. Poslední sekce 5.4 se věnuje samotným experimentům.

5.1 Dataset

Kodeky jsou srovnány na třech typech plenoptických obrázků ze tří veřejně dostupných internetových archivů. Náhled testovaného datasetu je prezentován na obrázku 5.1. Podrobnosti o jednotlivých obrázcích jsou dostupné v tabulce 5.1.

První čtyři obrázky pochází z archivu "The (New) Stanford Light Field Archive"[1]. Tyto obrázky byly zachyceny portálovým zařízením, které pohybuje s fotoaparátem, v mřížce 17×17 pohledů. Disparita mezi jednotlivými pohledy se pohybuje v řádu desítek pixelů. Dalších pět obrázků pochází z archivu, který hostuje Heidelberg Collaboratory for Image Processing (HCI) "4D Light Field Dataset"[2] [6]. Obrázky z tohoto archivu jsou digitálně generované z polygonového modelu v mřížce 9×9 pohledů s disparitou v řádu jednotek pixelů. Posledních pět obrázků pochází z archivu "EPFL Light-Field Image Dataset"[3] [8]. Tyto obrázky byly zachyceny fotoaparátem Lytro Illum B01 (10-bit). Pro účely testování byly obrázky konvertovány do mřížky 15×15 pohledů s barevnou hloubkou 8 bitů na kanál. Disparita mezi pohledy v těchto obrázcích se pohybuje na subpixelové úrovni.

5.2 Kompresní metody

Referenční implementace navržené metody LFIF 4D, která byla popsána v této práci, byla srovnána s její 3D a 2D alternativou. Tyto alternativy ke kompresi využívají trojrozměrné,

[1] Stanford dataset – https://lightfield.stanford.edu/
[2] HCI dataset – http://hci-lightfield.iwr.uni-heidelberg.de/
[3] EPFL dataset – http://mmspg.epfl.ch/EPFL-light-field-image-dataset/

Tabulka 5.1: Popis testovaného datasetu.

Název	**Archiv**	**Rozlišení (px)**	**Velikost**	**Zachycení**
Amethyst	Stanford	$768 \times 1024 \times 17 \times 17$	681 MB	Lego gantry
Chess	Stanford	$1400 \times 800 \times 17 \times 17$	971 MB	Lego gantry
Stanford Bunny	Stanford	$1024 \times 1024 \times 17 \times 17$	910 MB	Lego gantry
Lego Truck	Stanford	$1280 \times 960 \times 17 \times 17$	1 044 MB	Lego gantry
Antinous	HCI	$512 \times 512 \times 9 \times 9$	64 MB	Syntetické
Dino	HCI	$512 \times 512 \times 9 \times 9$	64 MB	Syntetické
Medieval 2	HCI	$512 \times 512 \times 9 \times 9$	64 MB	Syntetické
Cotton	HCI	$512 \times 512 \times 9 \times 9$	64 MB	Syntetické
Tomb	HCI	$512 \times 512 \times 9 \times 9$	64 MB	Syntetické
Bikes	EPFL	$625 \times 434 \times 15 \times 15$	183 MB	Lytro Illum
Flowers	EPFL	$625 \times 434 \times 15 \times 15$	183 MB	Lytro Illum
Friends 1	EPFL	$625 \times 434 \times 15 \times 15$	183 MB	Lytro Illum
Lakehouse	EPFL	$625 \times 434 \times 15 \times 15$	183 MB	Lytro Illum
Stone Pillars	EPFL	$625 \times 434 \times 15 \times 15$	183 MB	Lytro Illum

Obrázek 5.1: Náhled testovaného datasetu.

resp. dvourozměrné bloky a tím dekorelují obrázek pouze v jednom směru, resp. vůbec. Pro účely vyhodnocení byl pro 3D alternativu zvolen směr vertikální. Kvantizační a linearizační matice pro 3D a 2D metodu odpovídají trojrozměrné, resp. dvourozměrné verzi matic využitých při 4D metodě v daném experimentu.

Dále je provedeno srovnání s metodami pro kompresi obrazu JPEG, JPEG 2000, HEVC intra a metodou pro kompresi videa HEVC. Metoda JPEG byla testována pomocí knihovny mozjpeg[4]. Knihovna byla konfigurována pro kompresi v barevném prostoru YCbCr bez podvzorkování barevné složky. Komprese a dekomprese metodou JPEG 2000 byla provedena pomocí knihovny OpenJPEG[5]. Knihovna byla konfigurována tak, aby používala ztrátovou kompresi pomocí vlnek DWT 9-7 a ztrátový převod mezi barevnými modely, jinak bylo nastavení ponecháno na implicitních hodnotách. Metoda HEVC, společně s její intra variantou, byla testována pomocí knihovny x265[6], ke které bylo přistupováno pomocí rozhraní FFmpeg[7]. Knihovna byla přizpůsobena pro metriku PSNR. Komprese probíhala v režimu placebo.

5.3 Metrika PSNR

Pro měření výkonu ztrátové komprese existuje celá řada metrik a postupů. Nejjednodušší metriky porovnávají obrázky na úrovni vzorků a pracují s rozdílem. Složitější metriky berou v potaz strukturu obrázku jako takového a zohledňují i lidské vidění. V této práci je srovnání provedeno pomocí špičkového poměru signálu k šumu (PSNR), který pracuje na úrovni vzorků a lidské vidění nezohledňuje. Tato metrika byla zvolena zejména kvůli tomu, že u plenoptických dat jiné metriky nepřináší žádnou novou informaci. Křivky často odpovídají metrice PSNR a liší se pouze jednotkou.

PSNR lze vypočítat ze střední kvadratické chyby (MSE)

$$\begin{aligned} \mathrm{MSE} &= \frac{1}{K}\sum_{k=0}^{K-1}(M_k - N_k)^2 \\ \mathrm{PSNR} &= 10 \times \log_{10}\left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}}\right) \end{aligned} \tag{5.1}$$

Vzorec porovnává dva stejně velké obrázky M a N, kde oba obsahují K vzorků. Hodnota MAX odpovídá maximální hodnotě vzorku v porovnávaných obrázcích. Pro osmibitová data je tato hodnota 255.

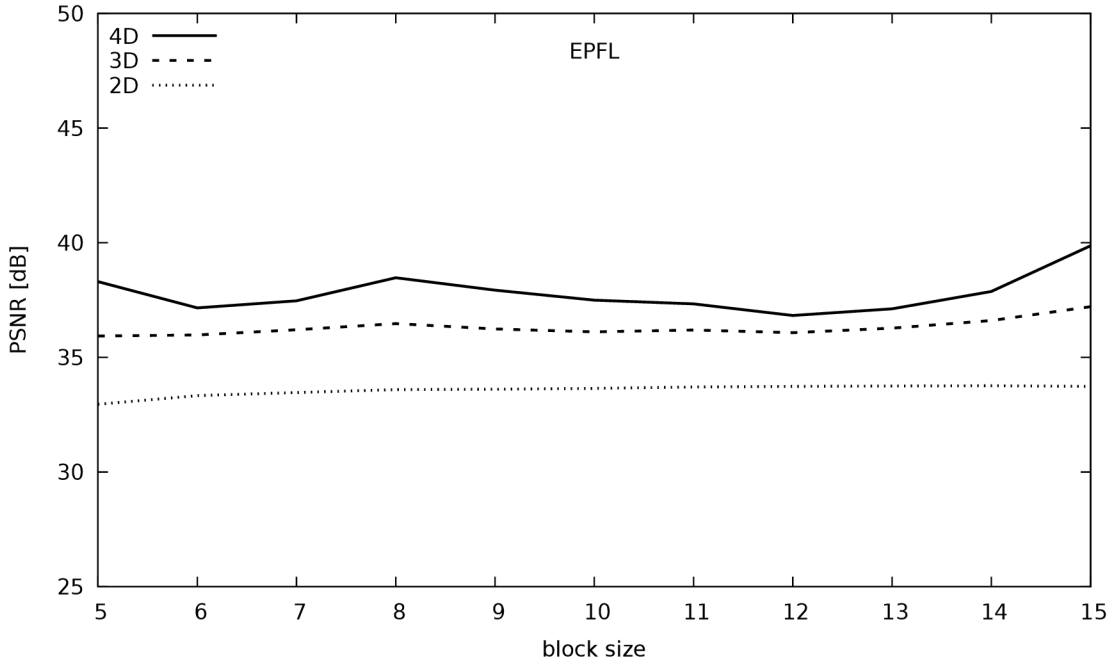
PSNR byla vypočtena nad původní a dekódovanou sadou jednotlivých pohledů na scénu v barevném prostoru RGB. U plenoptických dat existuje alternativní přístup, který porovnává kvalitu 2D obrázků vyrenderovaných z původního a komprimovaného 4D light field obrázku s různými úhly pohledu, rovinami ostření a clonou. Tento přístup je více tolerantní vůči ztrátám komprese, obrázky s hrubší kompresí v tomto případě dosahují vyšších hodnot PSNR.

5.4 Experimenty a vyhodnocení

Vyhodnocení bylo provedeno v pěti experimentech. První čtyři experimenty srovnávají různá nastavení implementovaného kodeku. Poslední experiment srovnává nejlepší naměřenou konfiguraci kodeku s již existujícími metodami. Výsledky jednotlivých obrázků byly proloženy spline křivkou a nad hodnotami PSNR byl proveden průměr.

[4] Knihovna mozjpeg – https://github.com/mozilla/mozjpeg
[5] Knihovna OpenJPEG – http://www.openjpeg.org/
[6] Knihovna x265 – http://x265.org/
[7] FFmpeg – https://ffmpeg.org/

Srovnání velikosti bloku

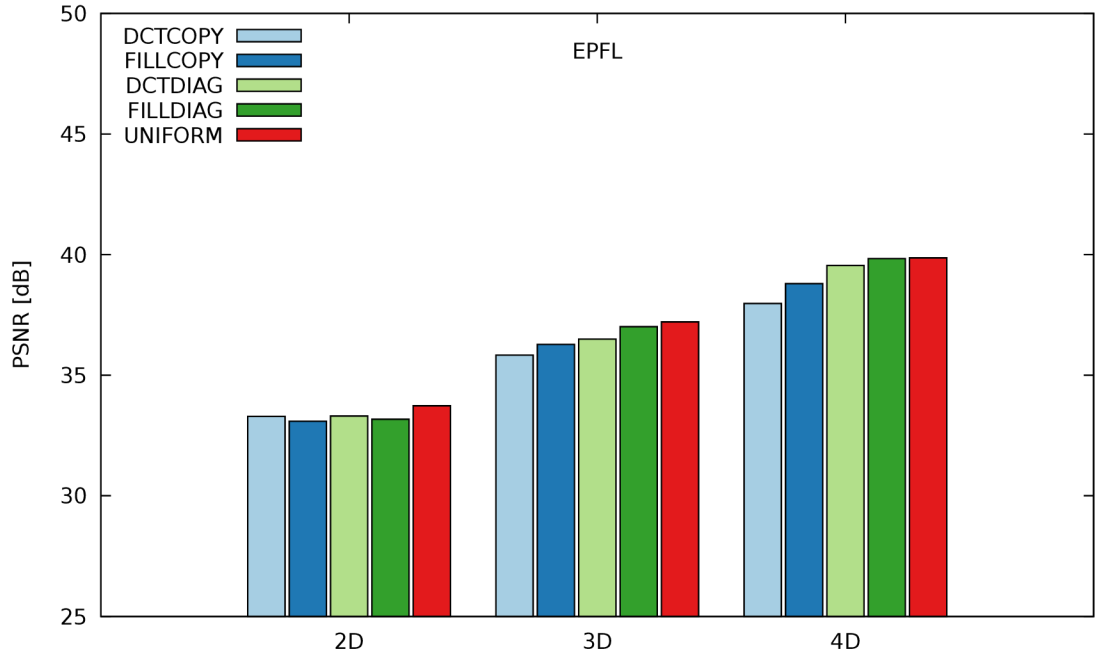


Obrázek 5.2: Srovnání velikosti bloku na datasetu EPFL (15×15 pohledů).

První experiment se zabývá měřením vlivu velikosti bloku na kompresní výkon. Podnětem k těmto experimentům byla myšlenka, že při kompresi obrázku, jehož počet pohledů není beze zbytku dělitelný velikostí bloku, může po rozdělení na bloky obsahovat více vzorků doplněných než původních. Tento problém si lze představit analogicky v metodě JPEG, kde se při kompresi obrázku o velikosti 4×4 pixelů doplní vzorky do bloku o velikosti 8×8. Četnost doplněných vzorků poté bude trojnásobná oproti četnosti vzorků původních. Jelikož je metodou JPEG málokdy komprimován obrázek o velikosti 4×4 pixely, lze tento problém zanedbávat. Ve čtyřrozměrném prostoru je ovšem problém umocněn dalšími dvěma rozměry a má proto signifikantní vliv na kompresní výkon.

Z grafu 5.2 plyne, že nejlepšího kompresního výkonu navržené 4D metody je dosaženo v případě, že velikost bloku odpovídá počtu pohledů. Metoda se chová stejně i!pro obrázky z ostatních datasetů (17×17 pro EPFL dataset, 9×9 pro Stanford dataset). V případě, že velikost bloku přesně odpovídá počtu pohledů, je minimalizován počet doplněných vzorků. K tomu se navíc v rámci jednoho bloku nejvíce projevuje korelace mezi pohledy. Dobrých výsledků je také dosaženo v případě, že n-násobek velikosti bloku odpovídá počtu pohledů, nebo jej přesahuje o jedničku. Mimo očekávání jsou přijatelné výsledky dosaženy i při kompresi s velikostí bloku, jejíž n-násobek je o jedničku menší, než počet pohledů. To sice vede k velkému množství doplněných vzorků, ovšem vzhledem k povaze doplňování těchto vzorků budou krajní bloky po 4D-DCT obsahovat nenulové frekvence pouze ve dvou dimenzích.

Srovnání kvantizačních matic

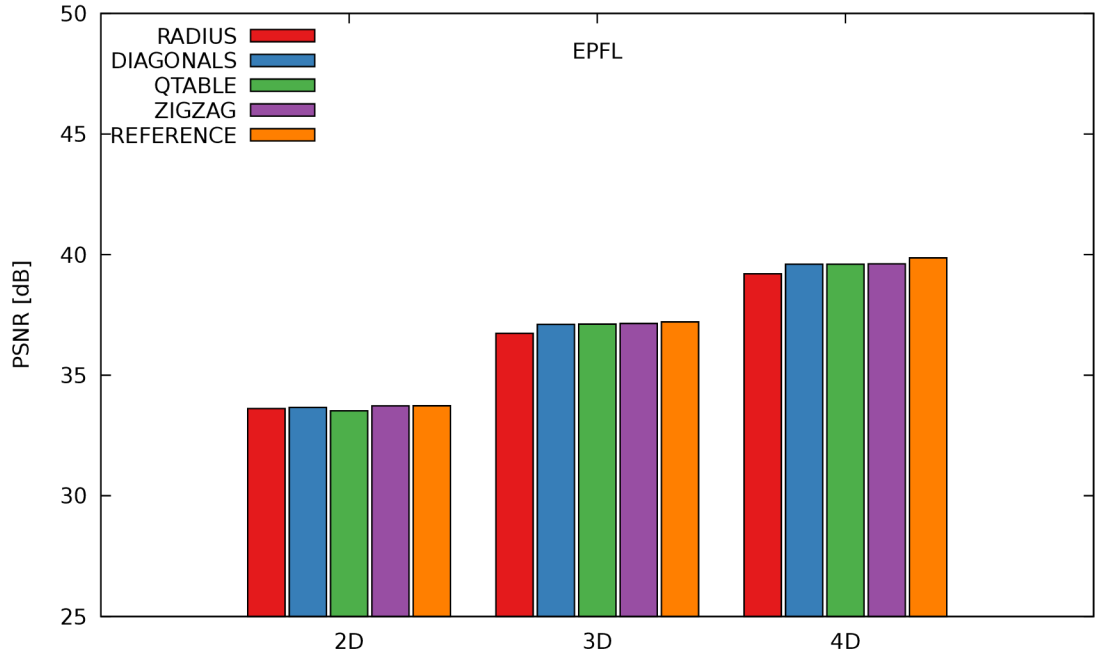


Obrázek 5.3: Srovnání kvantizačních matic na datasetu EPFL.

Druhý experiment srovnává navržené kvantizační matice. V této práci byly navrženy dva způsoby rozšíření matice na požadované rozměry a dva způsoby pro rozšíření matice do požadovaného počtu dimenzí. Metoda `FILL` rozšíří matici na požadované rozměry doplněním nejbližšími definovanými koeficienty. Metoda `DCT` provede na matici dopřednou DCT, matici rozšíří nulami a provede inverzní DCT. Metoda pro rozšíření na požadovaný počet dimenzí `COPY` naskládá určitý počet matic za sebe tak, aby vznikla matice požadovaných dimenzí. Metoda `DIAG` vypočte průměrné hodnoty diagonál a tyto hodnoty poté použije v příslušných diagonálách v matici požadovaných dimenzí. V tomto experimentu byly rozšiřovány matice, které používá knihovna mozjpeg. Vyhodnocení bylo provedeno s velikostmi bloků, které dosahovaly nejlepších výsledků pro daný dataset v předchozím experimentu. Jako reference v tomto experimentu slouží uniformní kvantizace, která všechny koeficienty kvantizuje stejnou hodnotou.

Z grafu 5.3 plyne, že nejlepších výsledků dosahuje uniformní kvantizace. Z metod rozšíření si vedou nejlépe matice, které jsou rozšířeny na požadované rozměry metodou `FILL` oproti metodě `DCT`. Zároveň jsou lepší matice, které jsou rozšířeny do čtyř dimenzí metodou `DIAG` oproti metodě `COPY`. Stejné chování se projevuje i na obrázcích z ostatních datasetů. V kompresi ve 2D se dimenzionální rozšíření původních matic neprovádí. Krok rozšíření `DIAG` je implementován tak, že pouze nahradí diagonály jejich průměrnými hodnotami. Krok `COPY` v tomto případě neprovede žádnou modifikaci původní matice. Z grafu plyne, že pro 2D metodu je vhodné škálovat matici metodou `DCT`.

Srovnání linearizačních matic

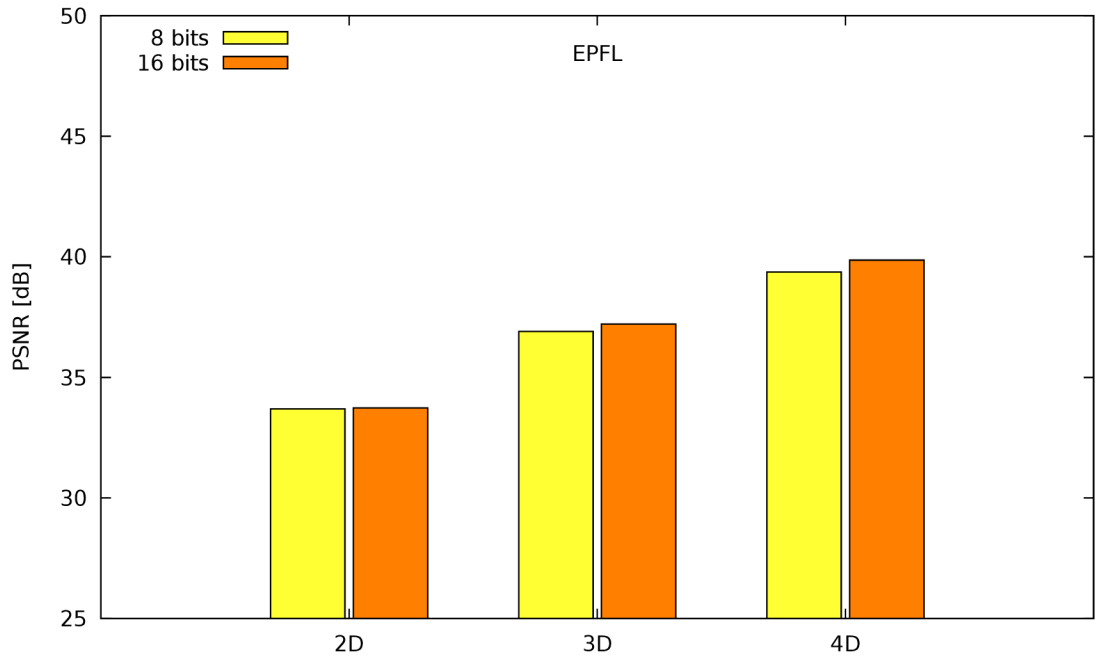


Obrázek 5.4: Srovnání linearizačních matic na datasetu EPFL.

Třetí experiment zkoumá vliv navržených linearizačních matic na kompresní výkon. Tento krok je bezeztrátový, nejsou proto očekávány žádné razantní skoky v kompresním výkonu. I přesto byly navrženy nové metody linearizace, se kterými bylo experimentováno s cílem najít lepší alternativu k cik-cak průchodu. Celkem bylo experimentováno se čtyřmi novými metodami. Metoda `QTABLE` provede průchod na základě kvantizační matice. Myšlenka je taková, že koeficienty, které jsou kvantizovány větší hodnotou budou menší a naopak. Metoda `RADIUS` provádí průchod na základě Eukleidovské vzdálenosti od DC koeficientu. Podobně metoda `DIAGONALS` prochází blok na základě manhattanské vzdálenosti od DC koeficientu. Metoda `REFERENCE` provádí průchod na základě průměrného rozložení koeficientů. Pro aplikaci této metody je potřeba provést výpočet průměrného rozložení koeficientů v bloku, což značně prodlužuje dobu komprese. Tyto metody jsou srovnány s referenčním průchodem `ZIGZAG`, který je rozšířením cik-cak průchodu do čtyř rozměrů.

Z grafů 5.4 plyne, že metoda `REFERENCE` dosahuje mírně lepších výsledků oproti metodě `ZIGZAG`. Zejména u nižších datových toků metoda `REFERENCE` vyniká v tom, že garantovaně seřadí koeficienty, které jsou ve všech blocích nulové, na konec linearizované struktury, což má pozitivní vliv na run-length kódování. Vliv na kompresní výkon je ovšem v porovnávání s cik-cak průchodem minimální, a to za cenu jednoho průchodu obrázkem navíc. Zisk z použití této metody je proto diskutabilní. Ostatní metody linearizace se stabilně ukázaly jako stejně či méně výkonné.

Srovnání velikostí symbolu Huffmanova kódování



Obrázek 5.5: Srovnání velikosti symbolu Huffmanova kódování na datasetu EPFL.

Čtvrtý experiment srovnává velikosti symbolu Huffmanova kódování. V předchozích kapitolách bylo vysvětleno, proč by symbol o velikosti 8 bitů, který je využíván v metodě JPEG, nemusel být dostačující. V této práci byl navržen symbol o velikosti 16 bitů, který by měl být dostatečně dimenzovaný i na 4D bloky s většími rozměry. Předpokládá se, že za cenu větší Huffmanovy tabulky v hlavičce výstupního streamu se tato změna projeví mírně lepším kompresním výkonem.

Z grafu 5.5 lze vyčíst, že při kompresi se symbolem o velikosti 16 bitů je dosaženo lepšího kompresního výkonu, než se symbolem o délce 8 bitů. Zlepšení je znatelnější při kompresi ve více dimenzích. Dá se předpokládat, že změna bude více znatelná také s rostoucí velikostí bloku. Obecně lze říct, že větší symbol těží zejména z velkých shluků nul mezi nenulovými koeficienty.

Srovnání s existujícími metodami

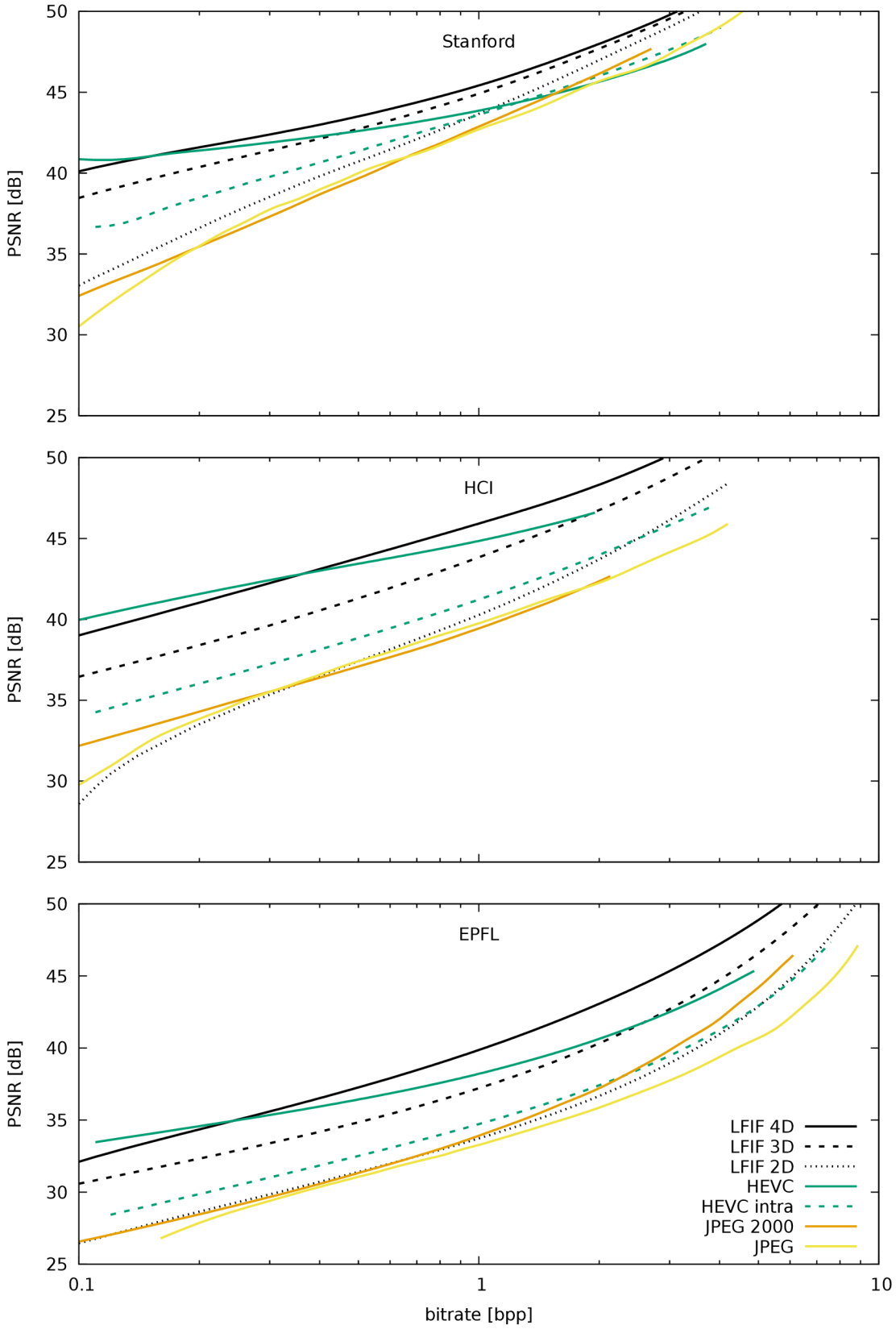
Pátý a poslední experiment srovnává navrženou metodu s existujícími metodami pro kompresi obrazu a videa. Z metod pro kompresi obrazu byly srovnány metody JPEG, JPEG 2000 a HEVC intra. Jako zástupce metod pro kompresi videa zde byla zvolena metoda HEVC, která je považována za jednu z nejvýkonnějších. Implementace metody navržené v této práci byla testována v její nejlepší konfiguraci na základě výsledků předchozích experimentů. Výsledky tohoto experimentu jsou demonstrovány na grafech 5.6.

Z grafů lze vyčíst, že kompresní výkon navržené metody se liší v závislosti na typu kódovaného obrázku a metoda v porovnání s ostatními metodami nabývá na výkonu se vzrůstajícím datovým tokem. Pro obrázky z datasetu univerzity Stanford překonává navržená metoda videometodu HEVC pro datové toky větší než jedna desetina bitu na pixel. To odpovídá kompresnímu poměru 1:240. Při tomto datovém toku dosahuje navržená metoda v průměru 41 dB PSNR. Trojrozměrná alternativa navržené metody LFIF 3D překonává videometodu HEVC kolem půl bitu na pixel, což odpovídá kompresnímu poměru 1:48. Videometoda HEVC je dále překonána všemi metodami pro kompresi obrazu. Z metod pro kompresi obrazu dosahuje nejlepších výsledků metoda HEVC intra. Ta je však překonána metodou LFIF 2D pro datové toky větší než jeden bit na pixel. Metoda JPEG se od metody JPEG 2000 pro tyto obrázky příliš neliší, se zvyšujícím se datovým tokem se tyto metody několikrát kříží.

Pro obrázky z datasetu HCI překonává navržená 4D metoda videometodu HEVC kolem čtyř desetin bitů na pixel. To odpovídá kompresnímu poměru 1:60 a při tomto datovém toku dosahuje metoda 43 dB PSNR. Z metod pro kompresi obrazu dosahuje nejlepších výsledků metoda HEVC intra. Ostatní dvourozměrné metody LFIF 2D, JPEG a JPEG 2000 dosahují pro tato data podobných výsledků, které se různí v závislosti na datovém toku.

Navržená 4D metoda dosahuje nejlepších výsledků i na obrázcích z datasetu EPFL, které jsou zachyceny fotoaparátem Lytro. Výjimkou jsou datové toky do čtvrtiny bitu na pixel, kde je navržená metoda opět překonána videometodou HEVC. Metoda HEVC je také překonána metodou LFIF 3D pro datové toky kolem dvou až tří bitů na pixel. Kromě metody HEVC intra dosahují 2D metody podobných hodnot. Rozdíly se projevují až se vzrůstajícím datovým tokem.

Z grafů plyne, že s výjimkou velmi nízkých datových toků navržená 4D metoda překonává videometodu HEVC. Navržená metoda těží zejména z větší velikosti bloku, která odpovídá počtu pohledů v obrázku. V případě menšího počtu pohledů je pro kompresi směrodatná disparita a povaha obrázku jako takového. Obrázky s velkým počtem odlesků a lomů světla jsou méně komprimovatelné v porovnání s obrázky, ve kterých převažuje zejména difuze. Ve všech grafech lze vidět, že navržená 4D metoda má stabilně lepší kompresní výkon v porovnání s 3D metodou a zároveň má 3D metoda stabilně lepší kompresní výkon v porovnání s 2D metodou. Zisk z vícedimenzionální komprese testovaných dat je proto nezpochybnitelný.



Obrázek 5.6: Srovnání kompresních metod.

Kapitola 6

Závěr

Cílem této práce bylo navrhnout, implementovat a vyhodnotit kompresní řešení pro plenoptické fotografie. Cíle byly v této práci splněny. První kapitola se věnuje teorii, která souvisí s plenoptickými fotografiemi. Jsou zde popsány formáty, ve kterých se tato data můžou vyskytovat. Druhá kapitola popisuje techniky ztrátové komprese obrazu a videa. Je zde podrobně popsána metoda JPEG, na které je založeno kompresní řešení pro plenoptické fotografie. Toto řešení je popsáno ve třetí kapitole. Řešení zrcadlí metodu JPEG do čtyř rozměrů a přináší nové myšlenky, které vedou k lepšímu kompresnímu výkonu. Řešení bylo implementováno jako programová knihovna, která je popsána v kapitole čtvrté. Navržená knihovna byla srovnána s existujícími kompresními metodami HEVC, HEVC intra, JPEG 2000 a JPEG v kapitole páté.

Referenční knihovna byla implementována v jazyce C++. Tato knihovna je parametrizovatelná pro kompresi v požadovaném počtu dimenzí a s požadovanou velikostí bloku. Knihovnu tak lze instancovat jako modifikovanou 2D metodu JPEG, metodu pro kompresi 3D volumetrických dat či metodu pro kompresi 4D plenoptických dat. V práci byly představeny nové kvantizační a linearizační matice pro multidimenzionální data. Byla zde představena možnost komprese s různými velikostmi bloku a oproti metodě JPEG byla modifikována také část entropického kodéru.

S novými myšlenkami bylo experimentováno a metoda byla modifikována až do fáze, kdy překonává videometodu HEVC. Zejména dobře navržená metoda funguje pro obrázky s větším počtem pohledů, jako jsou například fotografie pořízené fotoaparátem Lytro. Z výsledků této práce plyne, že plenoptické fotografie je možné efektivně komprimovat metodou JPEG rozšířenou do čtyř rozměrů. Dá se předpokládat, že pokročilejší metody pro kompresi obrazu se budou chovat podobně a bude tak možné dosáhnout ještě lepších kompresních výkonů.

Tato práce mne uvedla jak do problematiky plenoptických obrázků, tak i do problematiky ztrátové komprese. V této práci jsem se seznámil se strukturou kompresních řetězců a měl jsem možnost takový řetězec navrhnout a implementovat pro méně obvyklý, ale velmi perspektivní formát dat. Tato práce a její výsledky pro mne byly natolik motivující, abych pokračoval v bádání nad touto problematikou i mimo rozsah bakalářské práce.

V budoucí práci bych chtěl provést podrobné vyhodnocení kompresního výkonu pro data o různých bitových hloubkách. Dále bych se rád věnoval optimalizacím kvantizačních matic v závislosti na povaze komprimovaného obrázku, zejména na disparitě mezi pohledy. Mimo to bych rád experimentoval s rozšířením metody JPEG 2000 do čtyř rozměrů s cílem překonat metodu navrženou v této práci. Podobně bych také rád vyhodnotil nejnovější

metody pro kompresi videa, příkladem je metoda Versatile Video Coding (VVC), která je v době psaní této práce ve vývoji a metodu HEVC údajně překonává.

Tato práce byla prezentována na studentské konferenci Excel@FIT 2019 [4], kde byla oceněna odbornou komisí za kvalitní zpracování užitečného tématu a řadu experimentálních výsledků. Výsledky této práce byly také publikovány ve sborníku konference WSCG 2019 [3]. Referenční knihovna LFIF[1] byla zveřejněna na GitHubu pod zjednodušenou BSD licencí.

[1]Knihovna LFIF – https://github.com/xdlaba02/light-field-image-format

Literatura

[1] Adelson, E. H.; Bergen, J. R.: The Plenoptic Function and the Elements of Early Vision. In *Computational Models of Visual Processing*, MIT Press, 1991, s. 3–20.

[2] Bařina, D.: *Ztrátová komprese obrazu*. Bakalářská práce, Vysoké učení technické v Brně, Fakulta informačních technologií, 2007.
URL http://www.fit.vutbr.cz/study/DP/BP.php?id=5779

[3] Barina, D.; Chlubna, T.; Solony, M.; aj.: Evaluation of 4D Light Field Compression Methods. In *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, 2019.

[4] Dlabaja, D.: 4D-DCT Based Light Field Image Compression. In *Excel@FIT*, 2019.

[5] Ebrahimi, T.; Foessel, S.; Pereira, F.; aj.: JPEG Pleno: Toward an Efficient Representation of Visual Reality. *MultiMedia, IEEE*, ročník 23, č. 4, 2016: s. 14–20, ISSN 1070-986X, doi:10.1109/MMUL.2016.64.

[6] Honauer, K.; Johannsen, O.; Kondermann, D.; aj.: A dataset and evaluation methodology for depth estimation on 4D light fields. In *Asian Conference on Computer Vision*, Springer, 2016, doi:10.1007/978-3-319-54187-7_2.

[7] Levoy, M.; Hanrahan, P.: Light field rendering. In *Proceedings of the 23rd annual conference on computer graphics and interactive techniques*, SIGGRAPH '96, ACM, 1996, ISBN 0897917464, s. 31–42, doi:10.1145/237170.237199.

[8] Rerabek, M.; Ebrahimi, T.: New Light Field Image Dataset. 2016.
URL http://mmspg.epfl.ch/EPFL-light-field-image-dataset

[9] Szeliski, R.; Gortler, S.; Grzeszczuk, R.; aj.: The Lumigraph. 1996, doi:10.1145/237170.237200.

[10] Urbánek, P.: *Srovnání videokodeků*. Bakalářská práce, Vysoké učení technické v Brně, Fakulta informačních technologií, 2011.
URL http://www.fit.vutbr.cz/study/DP/BP.php?id=12253

[11] Wilburn, B.; Joshi, N.; Vaish, V.; aj.: High performance imaging using large camera arrays. *ACM Transactions on Graphics (TOG)*, ročník 24, č. 3, 2005: s. 765–776, ISSN 1557-7368, doi:10.1145/1073204.1073259.

[12] Wong, T.-T.; Fu, C.-W.; Heng, P.-A.; aj.: The plenoptic illumination function. ročník 4, č. 3, 2002: s. 361–371, ISSN 1520-9210, doi:10.1109/TMM.2002.802835.