



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

ANALÝZA NASTAVENÍ FORMÁTU JPEG 2000

ANALYSIS OF JPEG 2000 SETTINGS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MATĚJ MITAŠ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. DAVID BAŘINA, Ph.D.

BRNO 2019

Zadání bakalářské práce



21782

Student: **Mitaš Matěj**
Program: Informační technologie
Název: **Analýza nastavení formátu JPEG 2000**
Analysis of JPEG 2000 Settings
Kategorie: Zpracování obrazu

Zadání:

1. Seznamte se s formátem JPEG 2000. Zaměřte se na různá jeho nastavení a profily. Zabývejte se možnostmi progresivního přenosu, snížení výpočetních nároků a ochrany dat proti chybám přenosu.
2. Sestavte několik datových sad odlišného charakteru (např. běžné fotografie, skenované manuskripty).
3. Proveďte analýzu vhodného nastavení formátu pro jednotlivé datové sady. U ztrátové komprese zkoumejte závislost kvality a výpočetních nároků na datovém toku a jiných relevantních parametrech.
4. Vyhodnoťte výsledky. Doporučte nastavení vhodná pro danou datovou sadu.

Literatura:

- ISO/IEC 15444
- David S. Taubman, Michael W. Marcellin. *JPEG 2000: Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, 2001. ISBN-13: 978-0792375197

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2 dokumentované v technické zprávě. Rozpracovaný bod 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: 5. listopadu 2018

Abstrakt

Tato práce se zabývá analýzou nastavení formátu JPEG 2000. Cílem je vyšetřit vhodné hodnoty nastavení pro různorodá obrazová data vzhledem ke kompresnímu výkonu, časové a paměťové náročnosti za užití knihoven Kakadu a OpenJPEG. Za tímto účelem byl navržen a implementován testovací nástroj v jazyce Python. Výsledky této práce umožňují používat formát JPEG 2000 efektivněji a s hlubším pochopením jednotlivých nastavení.

Abstract

This thesis analyses settings of JPEG 2000 format. The aim is to find appropriate values for each setting. Main criteria are compression performance, elapsed time and used memory. There are two libraries used that are implementing JPEG 2000 format: Kakadu and OpenJPEG. Testing was secured by framework implemented in Python. Results of this thesis enable users to use JPEG 2000 more effectively and with deeper understanding of inner workings.

Klíčová slova

JPEG 2000, JPEG, Kakadu, OpenJPEG, komprese obrazových dat, ztrátová komprese, bezztrátová komprese

Keywords

JPEG 2000, JPEG, Kakadu, OpenJPEG, image compression, lossy compression, lossless compression

Citace

MITAŠ, Matěj. *Analýza nastavení formátu JPEG 2000*. Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. David Bařina, Ph.D.

Analýza nastavení formátu JPEG 2000

Prohlášení

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

.....

Matěj Mitaš
13. května 2019

Poděkování

Děkuji vedoucímu práce Ing. Davidu Bařinovi Ph.D. za odbornou pomoc a rady při zpracování této bakalářské práce.

Obsah

1	Úvod	2
2	JPEG 2000	3
2.1	Porovnání s předchůdcem	3
2.2	Řetězec zpracování	4
2.3	Shrnutí	9
3	Metodika testování	10
3.1	Kvalitativní metrika	10
3.2	Další měřené veličiny	11
3.3	Testovací data	12
3.4	Implementace	14
4	Analýza nastavení formátu	17
4.1	Porovnání formátů a testovací kritérium	19
4.2	Vyšetřování parametrů	22
4.3	Shrnutí testování	41
5	Závěr	42
	Literatura	43
A	Kompresní parametry	45
B	Dekompresní parametry	46

Kapitola 1

Úvod

Optimalizace velikosti obrazových dat pomocí komprese je velkým tématem počítačového světa. Dokáže ušetřit přenosovou kapacitu, stejně tak jako umožnit uživateli rychlejší a plynulejší práci. Práce s daty v nekomprimované podobě je výpočetně i paměťově velmi náročná. Jeden snímek videa s vysokým rozlišením (1920x1080 px) dosahuje v nekomprimované podobě v barevném prostoru RGB s 24 b na pixel velikosti 6.2 MB. Při běžné snímkovací frekvenci (24 snímků za sekundu) se jedná o značný objem dat. Navíc lze využít nedokonalostí lidského oka vzhledem k nepřesnostem obrazové komprese na rozdíl např. od textové. Drobná odchylka v barvě řádově desítek pixelů při současných rozlišeních (miliony pixelů) hraje naprosto minimální roli oproti chybně interpretovanému znaku v textu. Tato skutečnost poskytuje značné možnosti, zároveň je obtížné zvolit přesnou metriku pro porovnávání kvality zdrojových a zpracovaných dat.

Dlouhá léta byl - a stále je - standard komprese obrazových dat formát JPEG. Nároky uživatelů - např. průhlednost nebo bezztrátová komprese - které nebyl formát JPEG schopen uspokojit inspirovaly tvůrce k vytvoření jeho nástupce. Výsledkem jejich snahy byl JPEG 2000. Formát, jenž v mnohém staví na základech svého předchůdce. Navíc přidává nové možnosti mezi které patří více kvalit v jednom souboru nebo pořadí přenosu. Praxe ovšem ukazuje, že ani po skoro dvou dekadách od svého vzniku se nejedná o široce používaný formát. Dokonce jej některé běžné zdroje obsahu - fotoaparáty nebo scannery - nepodporují. JPEG 2000 je tak pokročilý, až je pro běžné uživatele příliš obtížné pochopit jeho možnosti. Existující různé práce obsahující doporučení nastavení [5] [13] [4], ale zpravidla jsou zmíněny vhodné hodnoty nastavení v nekompletní formě nebo bez bližšího vysvětlení. Hlavním smyslem této bakalářské práce je proto vytvořit ucelenou řadu doporučení pro práci s formátem JPEG 2000 nad rozličnými typy obrazových dat. Nejprve je čtenář seznámen se standardem a základním principem funkčnosti. Po přečtení této kapitoly by měl mít čtenář jasno, proč používat daný formát, kde leží těžiště výhod a kde naopak zvolit jiný technický prostředek. V následující kapitole je popsána metodika měření a vyhodnocování. Následně jsou ilustrovány výběry vstupních obrazových dat. Autorskou část práce tvoří následující dvě kapitoly. První z nich se zabývá návrhem a implementací testovací knihovny. Kapitola je zakončena informacemi o použitých nástrojích, technologiích a postupech, které umožnily postavit základy pro analýzu a vyhodnocení dat. Pro čtenáře zdaleka nejzajímavější bude druhá z nich. Jsou zde vyšetřeny nejdůležitější možnosti obou knihoven při ztrátové i bezztrátové kompresi. Z výsledků zde získaných je vytvořena ucelená sada doporučení pro kompresi dat formátem JPEG 2000.

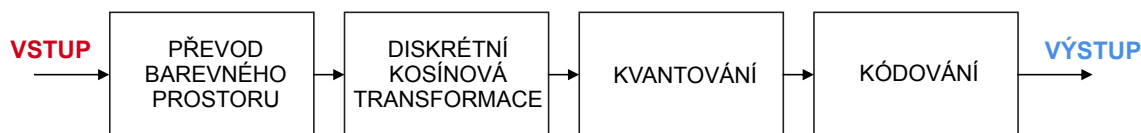
Kapitola 2

JPEG 2000

Jedná o formát [6] - definován normou ISO/IEC 15444 - jenž byl navržen *Joint Photographic Experts Group* v roce 2000 jako nástupce formátu JPEG [16]. Norma je rozdělena na více částí, které definují jednotlivé nosné myšlenky a přístupy formátu JPEG 2000. Těchto částí je mnoho, avšak pro účely tohoto textu postačí první dvě. První část (*Core coding system*) definuje celý rámec fungování, popisuje hlavní myšlenky a fundamentální principy. Druhá část (*Extensions*) se zabývá rozšířeními mezi které patří markery, nový formát výstupního souboru .jpx, zpracování videa a pokročilé zpracováváním exotických vstupní dat a formátů.

2.1 Porovnání s předchůdcem

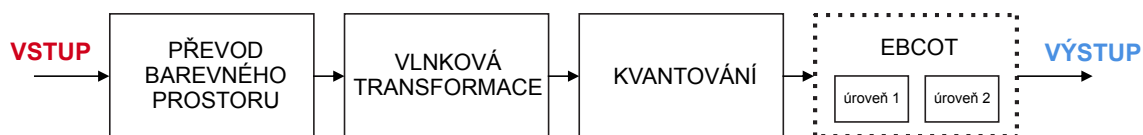
JPEG 2000 nabízí mnohé výhody oproti svému předchůdci. Pro uživatele je velice citelnou změnou podpora alfa kanálu (průhlednosti) a kompletní implementace bezztrátové komprese, kvůli které byl často používán formátu PNG. Uživatel zároveň dostává do rukou mnohem flexibilnější nástroj z hlediska nastavení jednotlivých možností komprese a dekomprese. Pokročilejší kódovací algoritmus zajišťuje lepší kvalitu při zachování stejné velikosti datového toku. S tímto fenoménem jdou ruku v ruce markery, které umožňují dekompresním nástrojům rozbalit pouze zadanou kvalitu, či dokonce region bez nutnosti výpočetně náročné dekomprese celého obrazu. I přes značnou podobnost základní funkcionality není zpětně kompatibilní. Zároveň je výpočetně i paměťově náročnější. Jelikož v mnohém staví na své předchůdci je nejprve vhodné si osvětlit základní principy fungování formátu JPEG.



Obrázek 2.1: Zjednodušený řetězec komprese JPEG.

První krok komprese zdrojových dat je transformace barevného prostoru z RGB do YCbCr. Výsledkem jsou tři složky obrazu: jas a barvonosné komponenty (modrá a červená). Barvonosné komponenty jsou poté podvzorkovány. Následně je obraz v přetransformovaném barevném prostoru rozdělen na bloky velikosti 8×8 px z důvodu úspory výpočetních zdrojů. Nad touto datovou strukturou je provedena Diskrétní kosínová transformace (DCT).

Jedná se o matematický aparát pro vyjádření libovolného signálu signály známými a dobře výpočetně zpracovatelnými signály. Výsledkem tohoto kroku je matice koeficientů o rozměru 64×64 . Následuje proces kvantování. Jedná o jedinou - pokud je zaručena dostatečná přesnost DCT a použit bezztrátový převod barevného prostoru - část komprese, která je ztrátová. Jak již bylo zmíněno v úvodu, při tomto kroku využívány jsou vlastnosti lidského oka. Je citlivé na malé změny jasu na velkých plochách, nicméně ne moc dobré v určování konkrétní síly jasu hran (ostré přechody v jasové složce). Tyto přechody jsou reprezentovány jako vysoké frekvence, které lze vhodně omezit či dokonce vynechat. V praxi je tento proces proveden dělením matice koeficientů maticí kvantizační a zaokrouhlením nově vzniklých koeficientů. Kvantizační matice ovlivňuje kvalitu komprese. Posledním krokem je zakódování koeficientů do datového toku. Z matice jsou vybrány metodou zig-zag, následně je tato posloupnost zakódována pomocí *Run-Length* (RLE) kódování, které dobře funguje pro shluky stejných znaků. V tomto případě dlouhé posloupnosti nulových koeficientů vysokých frekvencí. Kód je poté znovu zakódován pomocí Huffmanova kódování a následně složen jako výstupní zkomprimovaný soubor. Dekomprese je z velké části inverzní proces, pouze je třeba doplnit matice bloků 8×8 , jelikož vysokofrekvenční koeficienty byly při kompresi zahazeny. Předchozí popis jasně demonstruje značnou neflexibilitu formátu. V praxi jediné, co uživatel může ovlivnit je kvantizační matice (tabulka) pro nastavení kvality.



Obrázek 2.2: Zjednodušený řetězec komprese JPEG 2000.

2.2 Řetězec zpracování

JPEG 2000 se snaží v mnoha ohledech tuto neflexibilitu odstranit a nabídnout uživateli větší míru kontroly nad procesem. Základní stavební bloky komprese jsou podobné jako u předchůdce, jejich přesné fungování je v mnohém odlišné.

Transformace barevného prostoru a podvzorkování

Ačkoliv většina implementací umožňuje explicitně vynutit vynechání transformace barevného prostoru, pouze málokdy je k tomu dobrý důvod. Hlavním smyslem je odstranit nekonzistence v prostoru RGB. Velmi často dochází ke změnám jednotlivých složek pixelů v nejbližším okolí. Ačkoliv se může jednat o změny velmi malé, při nevhodně zvolené velikosti zpracovávaného bloku dochází ke vzniku blokových artefaktů.

Standardně používaným barevným prostorem je YCbCr, kde Y značí luminanci (jas), Cb/Cr chrominanci (barvonosné složky - červená, resp. modrá). Všechny důležité informace o obrazu jsou reprezentovány ve složce Y, což reflektuje citlivostí profil lidského oka [15], odstraňuje chyby barevné nekonzistence a zjednodušuje následné zpracování. Nakonec je důležité zmínit, že se nejedná o absolutní barevný prostor. Výsledný datový tok může být např. závislý na barevném profilu. Barvonosné složky lze podvzorkovat (např. 4:2:2, kdy dojde ke dvojnásobnému podvzorkování u barvonosných složek, ale luminance zůstane nedotčena).

Transformaci lze provést ve dvou módech. Buď ztrátově (*Irreversible Component Transform* - ICT) v aritmetice s plovoucí desetinnou čárkou.

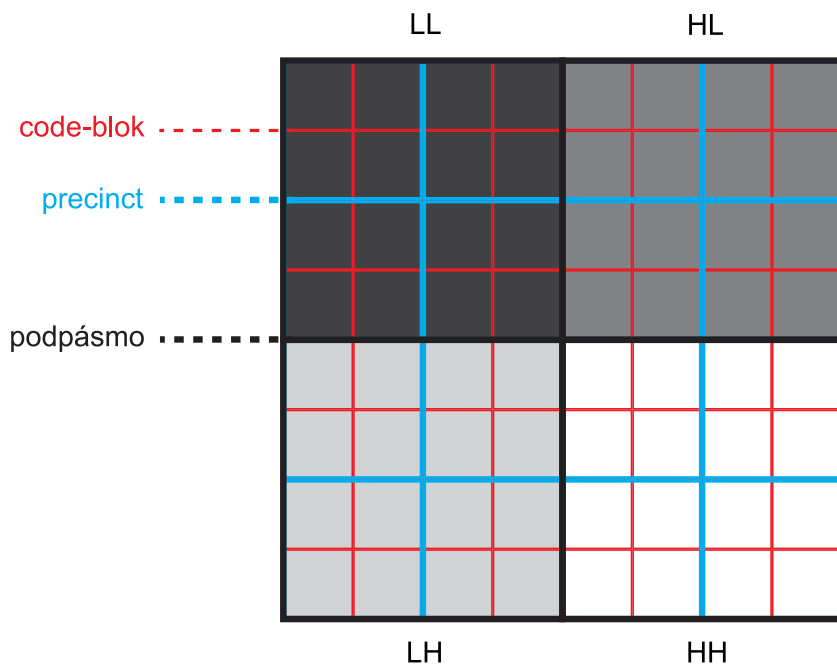
$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.5 \\ 0.85 & -0.41869 & -0.08131 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (2.1)$$

A nebo bezztrátově (*Reversible Component Transform* - RCT) v celých číslech.

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} \lfloor \frac{R+2G+B}{4} \rfloor \\ \lfloor B - G \rfloor \\ \lfloor R - G \rfloor \end{pmatrix} \quad (2.2)$$

Rozdělení obrazu

Zde probíhá první velký odklon od JPEG. U něj jsou normou definované bloky 8×8 px, které jsou samostatně zpracovávány. Jeho nástupce nabízí mnohem jemnější dělení a kontrolu nad procesem. V průběhu celé práce je možno se setkat s trojicí pojmů označující dělení obrazu. Jsou to dlaždice (*tiles*), precincty (*precincts*) a kódové bloky (*code-blocks*). Kromě dělby obrazu v této úrovni zpracování existují ještě tzv. regiony zájmu, které určitou část kódují prioritně. Tento proces probíhá po provedení vlnkové transformace.



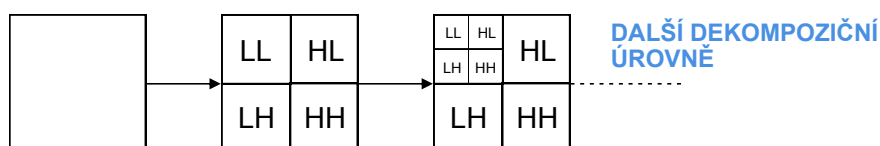
Obrázek 2.3: Nástroje pro dělení obrazu.

Nejprve je si nutno vysvětlit nastavené rozměry. Dlaždice z originálního obrazu vyřízne čtvercový blok o velikosti 512×512 px. Ačkoliv jsou v tomto případě voleny přesně čtvercové

rozměry dlaždice, je možno zvolit vzájemně nezávislé rozměry s libovolným poměrem stran za splnění podmínky nezápornosti obou rozměrů. Tato vlastnost umožňuje efektivně pokrýt dlaždicemi obrazové data atypických rozměrů. Standard také definuje možnosti posunu dlaždic a obrazu vůči výchozí bodu (levý horní roh). Dělení obrazu na dlaždice lze vypnout. Poté je obraz brán jako jedna dlaždice. Obraz je v tomto případě dekomponován pouze jednou. Výsledkem jsou čtyři podpásma o velikosti 256x256 px. Precinct v tomto případě dosahuje velikosti 128x128 px. Rozměr precinctu je přesně definován jako mocnina dvou a může se lišit pro každou dlaždici. Rozměry proto mohou být ušity na míru konkrétní části obrazu rozděleného dlaždicemi. Tento způsob dělení nemá přímý vliv na kvalitu, ale má vliv na velikost datového toku, jelikož každý precinct slouží jako základ pro pakety výsledného datového toku. Každý paket disponuje určitou režii - hlavně při použitých markerech na označení konce a začátku - je nutno velikost vhodně zvolit. Poslední dělicí komponentou je code-blok o velikosti 64x64 px. Code-bloky slouží jako vstupy kóderu.

Vlnková transformace

Srdcem komprese je Diskrétní vlnková transformace (DWT) [7]. Opět se jedná o tvorbu koeficientů na základě vstupních dat [8]. Hlavní rozdíl oproti *Discrete Fourier Transform* (DFT), potažmo DCT, je použití speciálního signálu (tzv. *wavelet* - vlnka). Ta umožňuje mnohem lépe popsat určitý jev v signálu. Harmonické průběhy mají problém s reprezentací náhlých změn průběhu signálu (hrany v obraze), jelikož nemají jasně vymezený čas (oscilují od $-\infty$ do ∞). Naopak vlnka u DWT má jasně vymezený čas a nulovou střední hodnotu. V případě JPEG 2000 je užitá vlnka *Cohen–Daubechies–Feauveau* (CDF) ve dvou variantách. Ztrátovou kompresi zajišťuje CDF 9/7 operující v plovoucí aritmetice vylučující úplnou zpětnou reprezentaci z důvodu zaokrouhlovacích chyb v případě dekomprese a bezztrátovou CDF 5/3 pracující na celých číslech. Některé implementace umožňují změnit volbu vlnky nezávisle na typu komprese; tato možnost bude experimentálně vyšetřena.



Obrázek 2.4: Naznačení vícenásobného rozkladu DWT.

Obraz je transformací rozložen na čtyři podpásma, která jsou dále zpracovávána. S každou další úrovní rozkladu je dekomponována nízkofrekvenční složka LL z důvodu citlivost lidského oka na nízké frekvence. Počet úrovní je jeden z nejdůležitějších parametrů ovlivňující kompresní výkon, jelikož je zapotřebí najít vhodný počet úrovní, kdy počet podpásem dosáhne dostatečného počtu pro kvalitní reprezentaci obrazu, avšak ne příliš velký, aby způsoboval značné zvýšení výpočetních nároků a zhoršení kompresního výkonu.

Kvantování

Kvantování funguje podobně jako u JPEG [12]. Odstraní nebo omezí složky málo důležité a nepotřebné (zpravidla vysoké frekvence). Na rozdíl od JPEG není použito implicitně vždy,

ale pouze při výběru ztrátové komprese. Běžně se používá skalární kvantizace, které funguje na principu odseknutí méně významných bitů koeficientů od určité bitové pozice.

Kódování

Tvoří poslední díl skládačky. Slouží k zakódování koeficientů DWT do příslušných datových struktur. Pracuje s code-bloky, které jsou kódovány nezávisle. Implementačně i principiálně se jedná o mnoho složitější funkční blok než u JPEG. Proto je rozdělen na dvě části.

EBCOT

Embedded Block Coding with Optimised Truncation (EBCOT) [14] popisuje pokročilý algoritmus na správu bitů v toku. Obecný princip je následující:

- Rozdělení code-bloků do n bitových rovin, které jsou seřazeny od nejvýše významného po nejméně významný bit.
- Každá rovina je zakódována třemi průchody, kde každý bit může být vybrán jenom jednou. Pokud je rovina složena pouze z nulových koeficientů je přeskočena, ale je započtena do celkového počtu rovin. Pro každý bit v code-bloku je vytvořena dvojice proměnných. Binární stav významnosti, nastaven při prvním výskytu hodnoty v koeficientu a vektor stavů významnosti nejbližšího okolí; zde osmi-okolí. Kódování probíhá v těchto úrovních:
 - *Significance Propagation Pass* - Bit je zakódován, pokud je nevýznamný, ale alespoň jeden z jeho sousedů významný je.
 - *Magnitude Refinement Pass* - Jsou zakódovány bity, které již významné byly, stejně tak jak ty označené v předchozím průchodu.
 - *Clean-up Pass* - Bity, které nebyly zakódovány žádnou z předchozích fází. Tato fáze je zároveň první při začátku průchodů novým code-blokem.
- Dvojice informací o významnosti - zde nazýváno symbol pro ukazatel významnosti a kontext pro informace o okolí z předchozího bodu - je zaslána do aritmetického kodéru.

Algoritmus zároveň definuje průchod code-bloků. Každý je rozdělen na vodorovné pruhy o výšce 4 b. Pruhy jsou zpracovávány zhora dolů. Pohyb v pruhu bitů je svislý, zleva doprava. Každý sloupec v pruhu je zpracováván svisle zhora dolů. Aritmetický kodér (dále MQ kodér), který je do značné míry zodpovědný za výsledný datový tok, umožňuje velkou míru nastavení. Z důvodu komplexních algoritmů existují formátem definované předvolby [6].

Generování výsledného toku

MQ kodér tvoří na svém výstupu posloupnost, která se skládá zpět na precincty (nadmnožina code-bloků) formující pakety po přidání hlavičky. Pakety všech podpásem jsou zkomponovány do vrstev. Pořadí paketů ve vrstvě není definováno. Obsah vrstev lze ovlivnit např. pomocí regionů zájmu. Velká výhoda formátu je možnost dynamicky dekodovat obraz po vrstvách - velmi vhodné např. pro pomalé přenosy dat, kdy je kritické uživateli zobrazit obsah i za cenu horší kvality - stejně tak jako možnost vytvářet nové vrstvy kvality z zakódovaného toku pomocí již vytvořených vrstev, kdy není třeba překódovat celý obrázek.

Typ	Vlastnost
BYPASS	Používá pouze aproximace průchodů, může být značně rychlejší za cenu nepatrně horší komprese.
CASUAL	Rozdělení zpracované části bitové roviny do pruhu o velikost 4 a zmenšení vlivu budoucích pruhu na současné kódování.
RESET	Vektor významnosti okolí je spočten pouze jednou.
RESTART	MQ kodér je při každém kódovacím průběhu vyčištěn.
ERTERM	Zajišťuje přesně definované chování symbolů a kontextu.
SEGMARK	Na konec každé bitové roviny je přidána čtyřbitová kombinace.

Tabulka 2.1: Možnosti nastavení kodéru.

Formát výstupního souboru

Vygenerovaný výsledný tok se musí někde uložit. K tomu účelu slouží kontejner `.jp2` definovaný v první části standardu. Další části definují jiné možné formáty (`.jpx` pro rozšířený standard a `.jpm` pro video), nicméně pro účely toho textu postačí `.jp2`. Soubor `.jp2` je rozdělen na boxy [10]. Existují povinné a volitelné boxy. Standard definuje čtyři povinné boxy. První z nich je *JPEG 2000 Signature*, který identifikuje soubor jako formát JPEG 2000. Následuje *File Type* a *JP2 Header*, který obsahuje informace o metadatech v souboru. Opět je specifikováno, která jsou povinná. Základní informace o obrazu jako jsou jeho rozměry, počet komponent nebo typ komprese obsahuje *Image Header*. Hned za ním následují informace o bitové hloubce každé komponenty v boxu *Bits per component*. Poslední povinné metadata jsou informace o barevném prostoru v boxu *Color Specification*. Poté následuje box *Contiguous Code-Stream* obsahující datový tok.

JPEG 2000 používá širokou plejádu markerů pro vyznačení určitých jevů v datovém toku. Dělí se do určitých skupin podle účelů použití (organizační, vyznačující, paketové, informační). V této práci budou vyšetřeny pouze paketové (konkrétně SOP a EPH označující začátek a konec paketu), z důvodu nekompletní implementace obou knihoven. Organizační a informační nejsou příliš zajímavé z pohledu testování (markery první skupiny označují začátek a konec toku dat a druhá skupina slouží k uložení komentářů). Poslední skupinou jsou vyznačující markery, mezi něž patří TLM (označení konkrétní dlaždice pro rychlý náhodný přístup v datovém toku), PLT (uchovávání délky paketů taktéž zlepšující náhodný přístup) a PPM (umožňuje přesunout hlavičky paketu na jinou lokaci, než na které se nachází tělo paketu).

2.3 Shrnutí

V této kapitole byla názorně demonstrována většina klíčových myšlenek formátu JPEG 2000. Byl porovnán se svým předchůdcem. Každá část kompresního řetězce byla popsána a u každé zmíněno, které parametry ji ovlivňují. Je důležité podotknout, že nebyla obsažena celá norma - a to ani není cílem této práce. Navíc některé parametry sice byly zmíněny, avšak ne všechny budou vyšetřovány, jelikož je nelze zobecnit na celý dataset (např. regiony zájmu) nebo vyšetřit s zvolenými knihovnamy (rychlý náhodný přístup). Mezi vyšetřované parametry se řadí: převod barevného prostoru, velikost dlaždice a její posun, regiony zájmu, použitá vlnka vlnkové transformace, dekompoziční úrovně, velikost precinctů a code-bloků, pořadí přenosu, vrstvy kvality a módy kodéru pro snížení výpočetní náročnosti a zabezpečení. Nyní si již stačí definovat testovací rámec - to bude úkolem následující kapitoly - a lze započít samotné vyšetřování.

Kapitola 3

Metodika testování

Každé testování by mělo mít alespoň dvě části: testovací data a zopakovatelný předpis testu. Předpis testu jasně definuje jaká data zpracovává, jaké parametry vyšetřuje a jaké parametry jsou konstantní po celou dobu běhu testu. Zároveň je nutno zajistit dostatečný počet opakování individuálních testů pro odstranění šumu (např. přístup k I/O, momentální vytížení pracovní stanice jinou činností) a získání relevantní statistické hodnoty. Z tohoto důvodu byl každý test proveden minimálně desetkrát, pokud není řečeno jinak. Z pohledu uživatele je při kompresi nejdůležitějším parametrem velikost výstupního souboru. Ta ovlivňuje objemy přenesených dat v čase a je důležité ji při zachování určité úrovně kvality co nejlépe využít. Metrik pro kvalitu komprese obrazových dat využívá více (např. SSIM apod.), nicméně pro tuto práci je naprosto dostačující metrika PSNR.

3.1 Kvalitativní metrika

Způsobu určování kvality může být více. První skupinou jsou metody zkoumající velikost zkomprimovaného souboru. Nejznámějším je kompresní poměr. Jedná se o prostý podíl velikosti původního a zkomprimovaného souboru. Následuje počet bitů na pixel (dále používaná jako zkratka **bpp**) neboli bitová hloubka. Obě tyto metriky neberou v potaz objektivní obrazovou kvalitu. Je možno obecně určit hranice, kdy je daný obraz ještě kvalitativně dobrý a kdy už ne, ale nereflktují např. na kompresní artefakty nebo chyby způsobené nevhodnou volbou barevného prostoru. Proto je v této práci - vždy, když to bude relevantní - použita jako metrika PSNR. Jedná se o logaritmičticky vyjádřený - z důvodu velkých rozsahů - poměr maximální možné energie signálu a střední kvadratické chyby (MSE). Maximální možná energie signálu je definována jako $(2^n - 1)$, kde n udává počet bitů na kanál. Všechny vstupní obrazová data disponují 8 b na kanál, proto je každý pixel v RGB zakódován jako 24 b hodnota.

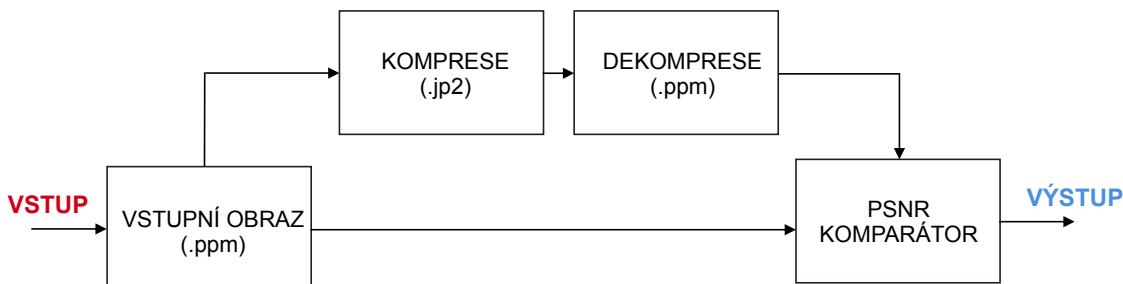
Výpočet MSE je definován:

$$\text{MSE} = \frac{1}{m \cdot n} \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} [I(i, j) - K(i, j)]^2 \quad (3.1)$$

Výpočet PSNR je definován jako:

$$\text{PSNR} = 10 \cdot \log \cdot \left(\frac{(2^n - 1)^2}{\text{MSE}} \right) \quad (3.2)$$

Výsledkem výpočtu je hodnota vyjádřená v decibelech. Pro dva stejné obrázky vychází hodnota ∞ , pro odlišené - rozměrově nebo obsahově - 0. Kvalitní komprese se pro 8 b/kanálové obrázky pohybuje mezi 30 dB až 40 dB. V tomto případě, kdy jsou obrazová data 100 MPx a větší, i relativně velká komprese v mnoha případech nezpůsobí značnou míru zkreslení kvůli velkému objemu dat. Proto je možno zvýšit kompresi až k hranici 25 dB při zachování objektivní kvality. Kvalitativní požadavky vyjádřené PSNR se mohou mezi měřeními měnit.



Obrázek 3.1: Blokové schéma komparace.

Implementace komparace dle PSNR je realizována následujícím způsobem. Vstupní data uložená jako .ppm/.pgm jsou zkomprimována do souboru .jp2, následně zpět dekomprimována do souboru .ppm/.pgm a porovnána s originálem.

3.2 Další měřené veličiny

V této práci kvalita není jedinou sledovanou veličinou.

Časová náročnost

Přímý vliv na efektivitu uživatelské práce. Zpravidla není komprimován pouze jeden obrázek, ale často je nutno někde uložit sadu fotografií (např. z dovolené) nebo poslat větší množství scanovaných dat (technické výkresy). Tato veličina je přímo úměrná rychlosti pracovní stanice. Čas byl měřen utilitou *usr/bin/time* a výstupem byl parametr *real*. Tento parametr byl zvolen kvůli možnému běhu knihoven v režimu zpracování více vláken. Parametr *user* poskytuje pouze spotřebovaný čas jako ekvivalent jednoho vlákna.

Paměťová náročnost

Nastaveními lze do určité míry ovlivnit i paměťovou náročnost. Tento výrok platí hlavně pro knihovnu OpenJPEG. Například volba velikosti dlaždice změní velikost zpracovávaného počtu dat. V případě dat o velikosti v řádech GB může nastat problém s nedostatkem paměti. Využití paměti bylo měřeno *usr/bin/time -l* a výstupem byl parametr *maximum resident set size*, který zaznamenává špičku paměti během celého běhu procesu. Jedná se tedy o nejhorší možný případ se kterým je nutno počítat.

3.3 Testovací data

Dle zadání byly vytvořeny čtyři (dataset „Bitonální“ se ještě dělí na barevné a černobílé obrázky) datasety reprezentující možnosti a rozdíly mezi nastaveními.

Typ	Velikost	Počet kanálů	Bitů na kanál	Počet
Fotografie	>100 MPx	3	8	110
Scany	>80MPx	3	8	120
Mapy	>75MPx	3	8	80
Bitonální RGB24	>100MPx	3	8	110
Bitonální Gray8	>100MPx	1	8	20

Tabulka 3.1: Základní informace o datasetech.

Pro statisticky významné měření obsahuje každý z datasetů alespoň 80 souborů. Jsou uloženy v formátu PPM/PGM ze specifikace Netpbm. Jedná se o formát obsahující surová data vhodně uložená k dalšímu zpracování.

Dataset „Fotografie“

Tento dataset slouží jako reference pro mnoho typů fotografií. Jedná se o soubor mnoha typů barevných fotografií (portréty, vesmírné, detaily, krajiny).

Dataset „Scany“

JPEG 2000 nabízí značné výhody při kompresi scanovaných dokumentů ať to jsou vrstvy kvality (dokáže nahradit mnoho adresářů s různými kvalitami jedním souborem) nebo speciální kompresní profily.

Dataset „Mapy“

Zpravidla se také jedná o scanované dokumenty, nicméně předchozí dataset jsou zpravidla obrazová data obsahující množinu textových znaků. Je možno nastavit velké úrovně komprese při zachování hodnoty informace (například pro OCR zpracování). Naopak u mapových podkladů - kdy množina tvarů není konečná, jako u textu, tzv. grafémy - je nutno zachovat preciznost obrazu. Jedná se o velice podrobné mapy s vysokým rozlišením, kde i malá míra zkreslení může mít velký dopad.

Dataset „Bitonální“

Bitonální obrazová data jsou ta, která obsahují pouze dva tóny (dvě barvy). Jedná o experimentální měření, protože tento typ dat není běžně používán. Jsou ovšem velmi zajímavé z hlediska nastavení kompresního řetězce, jelikož se jedná o relativně exotický obrazový materiál, u kterého lze po vhodném nastavení očekávat zlepšení. Část datasetu slouží k testování vstupní dat v úrovních šedi (Gray8).



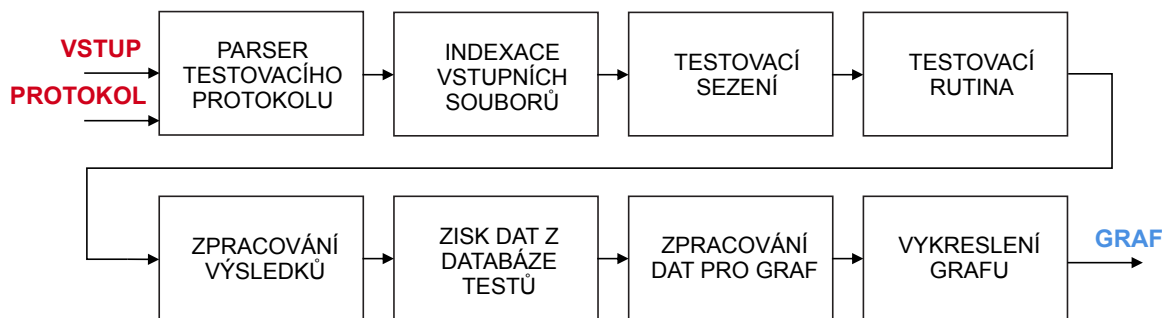
Obrázek 3.2: Náhled zpracovávaných datasetů.

3.4 Implementace

Implementace testovacího nástroje proběhla v jazyce Python verze 3 s využitím knihovny numpy pro matematické výpočty a nerelační databázi MongoDB pro průběžné ukládání výsledků. Byl kladen velký důraz na rozšiřitelnost a uživatelský komfort. Průběžné ukládání informací o testech slouží pro účely zpřesňování výsledků a vykreslování grafů. Jako implementace formátu JPEG 2000 byly použity dvě knihovny - Kakadu [1] a OpenJPEG [2]. Každá z knihoven přistupuje k dané problematice jinak, hlavně kvůli tržnímu postavení, kdy Kakadu plní roli komerčního nástroje pro velké instituce a OpenJPEG je nejpoužívanější implementace s volně dostupným zdrojovým kódem. Z hlediska pokrytí standardu má jasně navrch Kakadu, které umožňuje pokročilejší práci s parametry nastavení. Nutno dodat, že ani jedna z knihoven plně nepokrývá ani první část standardu, na druhou stranu Kakadu obsahuje naprostou většinu podstatných věcí (např. chybí markery PPM/PLT). Vzhledem k použití obou knihoven při testování odpadá možnost vyšetřovat určité parametry simultánně - hlavně u OpenJPEG - který některé nastavení neimplementuje nebo ignoruje.

Testovací nástroj

Testovací nástroj je spouštěn z příkazové řádky v třech variantách. Před každým testem - a po určité uživatelem nastavitelné době - je nejdříve nutno projít vybranou složku a zjistit informace o všech souborech. Mezi nejdůležitější informace patří rozlišení obrazu, počet kanálů a bitová hloubka. Tyto informace umožňují efektivně počítat kompresní poměr a pomáhají s nastavením kompresního řetězce. Po úspěšném získání jsou výsledky testování uloženy do databáze. Nástroj funguje pouze nad obrazovým materiálem ve formátu .ppm pro třísložkové a .pgm pro jednosložkové obrázky. Nejdůležitější variantou spuštění je testování. Jak již bylo zmíněno v předešlé sekci, každé testovací sezení disponuje svým předpisem. Všechny jsou uloženy ve formátu .json pro snadnější strojové zpracování. V jednom souboru se může nacházet více sezení.



Obrázek 3.3: Blokové schéma testovacího nástroje.

```

{
  "type" : "approximation",
  "files" : {}
  "dir" : "path",
  "drivers" : ["drivers"],
  "routines" : {
    "compress": {
      "criterion" : {
        "type" : "psnr",
        "value" : [opt1...]
      },
      "testing_param" : {
        "flag" : "tiles",
        "opts" : [opt1...]
      },
      "params" : {}
    },
    "decompress": {
      "testing_param" : {},
      "params" : {}
    }
  }
}

```

Základní struktura předpisu testovacího sezení je relativně jednoduchá. Nejprve je nutno zvolit druh. Ten určuje počet opakování testu pro zajištění statistické věrohodnosti vyšetřených výsledků, hlavně časových a paměťových nároků. Existují dva druhy testování: aproximační a koncové. Pokud by jediné kritérium vyšetřování komprese byla bitová hloubka, stačilo by koncové testování s dostatečným počtem opakování. Když je jako kritérium zvoleno PSNR, nelze dopředu jednoznačně určit jakému PSNR odpovídá konkrétní bitová hloubka. Lze přibližně určit interval, kde se nachází, avšak pro dosažení přesnosti aproximace (zvolena na dvě desetinná místa) je nutno každý soubor - a každou jeho variantu - několikrát prohnat celým řetězcem zpracování. Tato problematika klade značné nároky na rychlost určení odhadu a zvolenou metodu aproximace. Byl kladen velký důraz na kvalitu implementace, jelikož zpracovávané soubory velmi často dosahují značných velikostí, které znesnadňují efektivní algoritmizaci problému. Proběhly pokusy se strojovým učením, které byly ovšem záhy opuštěny z důvodu značného zesložnění programu s nevalnými výsledky v časové doméně. Možnost zjednodušení se také jevila při zmenšení souborů (řádově jednotky megapixelů) a jejich následné vyšetření pomocí triviálních algoritmů jako je např. bisekce intervalu. Tento přístup dosahoval velmi dobrých výkonů, avšak křivka kvality není přesně škálovatelná - hlavně u atypický dat. Určení intervalu obsahujícího řešení bylo velmi rychlé. Kromě zmenšení obrazu, jelikož se jedná o výpočetně náročnou operaci, kterou lze provést ale před testováním, nicméně úvodní náklady na testování neúměrně vzrostou. S vymezeným intervalem stačilo pouze najít koeficient zvětšení obrazu a kvality, což bylo realizováno výpočtem rozdílů kvalit na obou koncích intervalu původního i zmenšeného souboru. Při testování toho přístupu se zvolenou přesností na jedno desetinné místo algoritmus vykazoval mnohem lepší výsledky než referenční řešení (bisekce intervalu). Jakmile byla zvolena přesnost vyšších řádů výhoda počátečního určení intervalu a následně lepší aproximace byla smazána, jelikož dramaticky vzrostl počet kompresí původního souboru na úkor zmenšeného. Když se připočte nutnost ošetření mnoha extrémních stavů byl tento přístup také opuštěn. Výsledkem aproximační fáze je v obou případech bitová hloubka, která u PSNR odpovídá zvolenému kritériu s určenou přesností. Následuje koncové testování, které si z databáze načte aproximační výsledky a na jejich základě testuje již vyšetřené

kritérium s určitým počtem opakování pro zajištění statistické přesnosti a odstranění šumu. Tyto výsledky slouží jako datový základ ke grafům.

Po výběru druhu je třeba zajistit práci se soubory. Soubory je lze filtrovat dle jména, omezit jejich počet nebo některé explicitně z testování vyjmout. Jelikož byl testovací nástroj zamýšlen jako plně rozšiřitelný pro další knihovny, je uživateli nabídnuta volba použitých knihoven. Následuje nastavení komprese a dekomprese. U obou jsou parametry děleny na dvě množiny. První z nich jsou pevné, které se nemění během celého trvání testování. Druhá obsahuje vyšetřovaný parametr a hodnoty jež nabývá. Počet záznamů v databázi odpovídá počtu vyšetřovaných hodnot. Tímto fáze testování končí, následuje zpracování dat a vykreslení grafu. Opět se jedná o netriviální úkon, jelikož existují dvě kritéria, které vyžadují odlišené zpracování a sekundární data k vykreslení (časová a paměťová náročnost). Vykreslovací nástroj umožňuje kombinovat více testovacích sezení, dokonce i s rozdílnými vyšetřovanými parametry při zaručení vzájemné kompatibility pevných parametrů. V neposlední řadě nabízí uživateli volbu rozsahu, typu a popisu grafů.

Kapitola 4

Analýza nastavení formátu

V této kapitole jsou rozebrány možnosti komprese (ať již ztrátové či bezztrátové) a dekomprese vzhledem k možnostem nastavení parametrů knihoven Kakadu a OpenJPEG. Technická stránka vyšetřování je osvětlena v předchozí kapitole. Byla zadána množina zkoumaných parametrů k vyšetření. Nejprve jsou demonstrovány používané profily a pro každý z pěti datasetů je zvolen jeden jako referenční. Každý parametr má svůj obor hodnot. Konkrétní nastavení parametru dle vybraného referenčního profilu je okomentováno, pokud je jím definován. Testováním je ověřena jeho vhodnost, případně slouží jako základ pro doporučení jiné volby. Pokud to knihovna dovolí, budou parametry vyšetřovány u obou zástupců, stejně tak jako časová a paměťová náročnost. Aby nebyl čtenář zatěžován množstvím nadbytečných grafů, které ilustrují stejný jev, ne vždy budou demonstrovány všechny hodnoty. Stane se tak pouze v případě, když mezi nimi bude dostatečný rozdíl či bude potřeba poukázat na konkrétní jev u dané knihovny, datasetu nebo profilu. V předchozích kapitole bylo vysvětleno v čem je aproximace k určitému PSNR vhodná, nicméně když je odhlédnuto od implementační a časové náročnosti nachází se zde ještě jeden problém. Parametr je vyšetřen pouze v jednom bodě kvalitativní stupnice - resp. ve dvou, pro ztrátovou a bezztrátovou - ale obě knihovny škálují s změnami kvality symetricky, proto tento přístup v naprosté většině případů dostačuje. Tam, kde bude záhodno ukázat průběh jevu na větším intervalu bude místo PSNR jako kritérium využita bitová hloubka. Některé testy se liší typem grafů. Cílem je ukázat spojitost veličiny, kdy např. převod barevného prostoru lze zapnout či vypnout, ale velikost dlaždice může být prakticky jakákoliv bez ohledu na poměr a velikost (samozřejmě respektující velikost vstupního obrazu). *Při zvoleném kritériu PSNR platí u kompresního výkonu, že čím menší bitová hloubka, tím lépe, protože kvalita je neměnná a mění se pouze velikost souboru.* Nakonec jsou zhodnoceny všechny dílčí výsledky měření, je poukázáno na zajímavé jevy a vydána sada doporučení pro práci s daným datasetem a knihovnou. Posloupnost vyšetřovaných parametrů není náhodná. Každý následující test využívá doporučené nastavení z předchozího testu.

Existující profily JPEG 2000

Ačkoliv se JPEG 2000 na spotřebitelském trhu neuchytil, tak jak si autoři představovali, knihovny a archivační instituce po celém světě jej hojně používají. Většina zde zmíněných profilů pochází z těchto zdrojů. Existují v relativně velké množství, nicméně drtivá většina z nich je velmi obecně zaměřena (fotografie či manuskripty) pro ztrátovou kompresi. Jak se v průběhu vyšetřování ukáže, mnohokrát se vyplatí pro určitý dataset věnovat úsilí při sestavování specifického profilu.

Parametr	Fotografie	Mapy	Scany	Bitonální
kompresní poměr	6:1	8:1 až 10:1	neuveдено	10:1
typ komprese	ztrátová	ztrátová	ztrátová	ztrátová
dlaždice	1024x1024	1024x1024	NE	NE
pořadí přenosu	RLCP	RPCL	RPCL	RPCL
dekompoziční úroveň	5	5 až 6	6	7
vrstvy kvality	8	12	ANO	7
regiony zájmu	NE	NE	NE	NE
code-blok	64x64	64x64	64x64	64x64
módy kodéru	NE	BYPASS	NE	NE

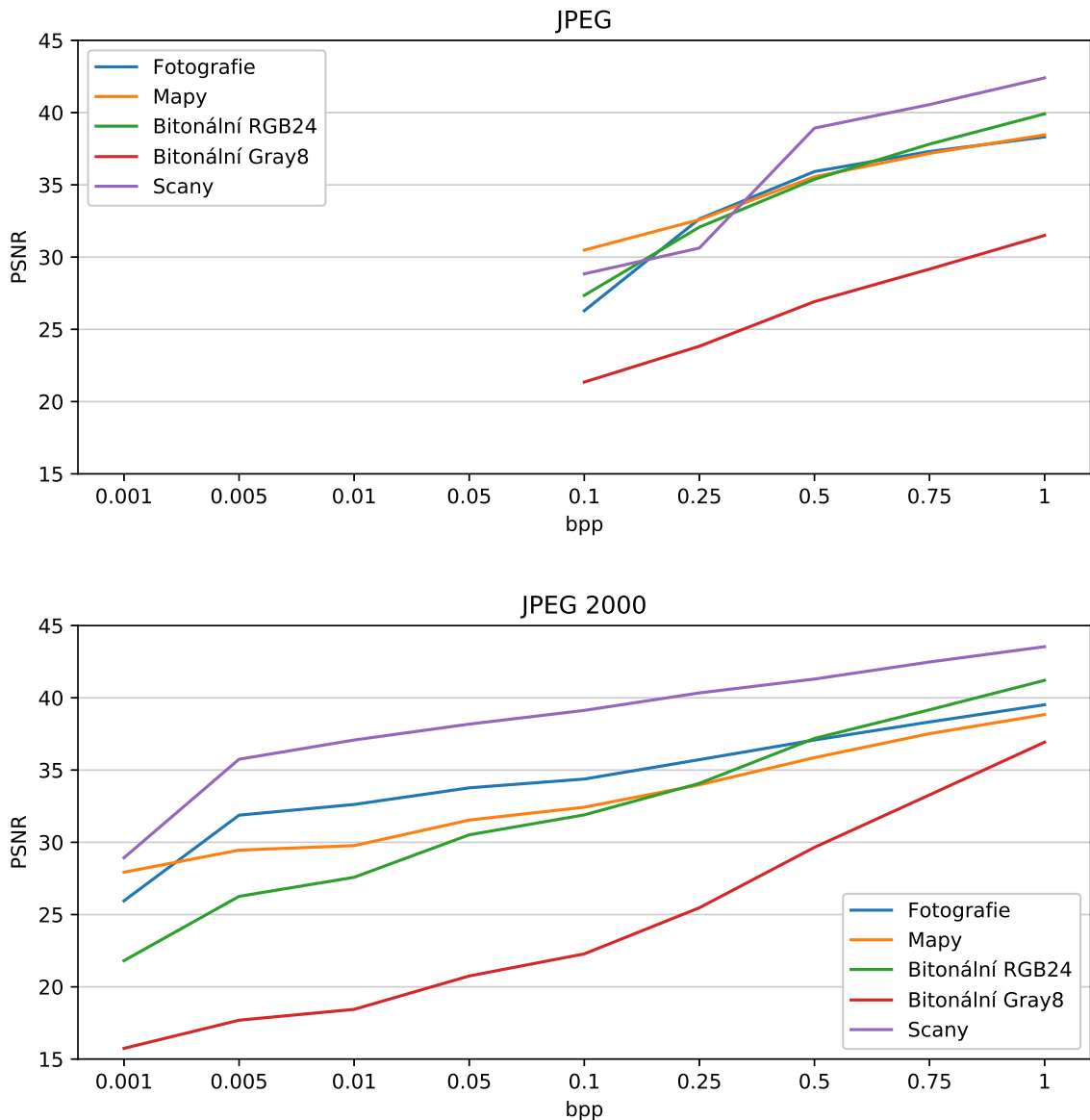
Tabulka 4.1: Profily ztrátové komprese. Jako zdroje těchto profilů slouží podklady Národní digitální knihovny [13], British Library [4] a Bibliotheca Alexandrina [3].

Parametr	Fotografie a Bitonální	Mapy a Scany
kompresní poměr	2:1	3:1
typ komprese	bezeztrátová	bezeztrátová
dlaždice	4096x4096	NE
pořadí přenosu	RPCL	RPCL
dekompoziční úroveň	5 až 6	6
vrstvy kvality	1	6
regiony zájmu	NE	NE
code-blok	64x64	32x32
módy kodéru	NE	neuveдено

Tabulka 4.2: Profily bezeztrátové komprese. Jako zdroje těchto profilů slouží podklady Národní digitální knihovny [13] a Library of Congress [11].

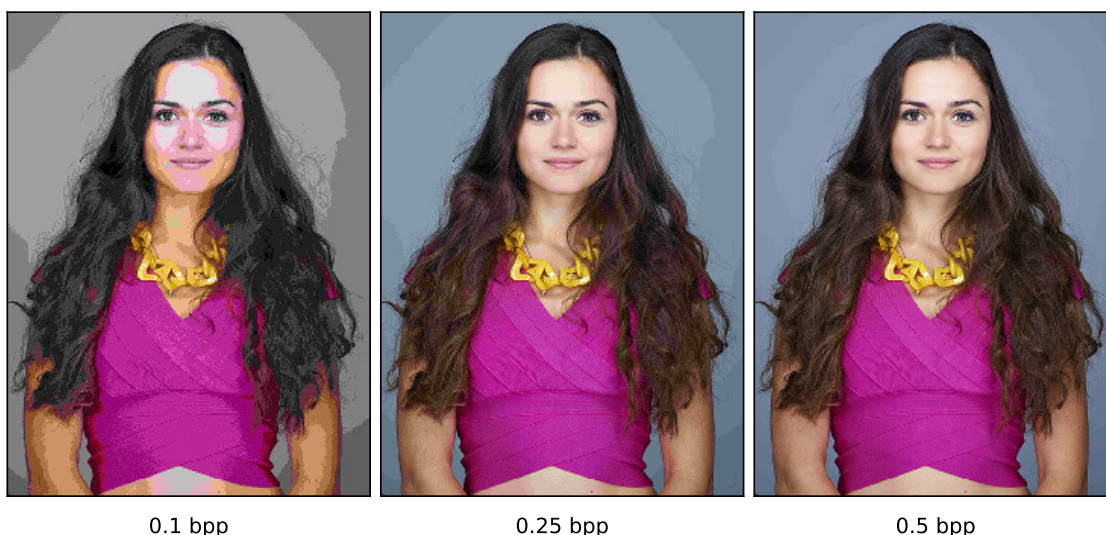
4.1 Porovnání formátů a testovací kritérium

Před samotným testováním je vhodné demonstrovat, kde se nachází obor hodnot sloužících jako kvalitativní nastavení komprese, jaké výkony komprese lze očekávat a jak moc se liší kompresní výkon mezi JPEG 2000 a jeho předchůdcem. Pro testování se zvoleným kritériem PSNR ztrátové komprese je nutno určit hodnotu, která bude reprezentativní napříč dataseť. Bitová hloubka vykazuje nejzajímavější hodnoty z pohledku komprese kolem kvality **35 dB**. Tato hodnota bude použita jako univerzální kritérium PSNR při testování ztrátové komprese.



Obrázek 4.1: Porovnání kompresních výkonů. Pro kompresi dle formátu JPEG byl použit nástroj Imagemagick s kompresní utilitou *convert*.

Při prvním pohledu do grafů je očividná jedna věc. JPEG nedisponuje možností zvolit menší hodnotu bitové hloubky než 0.1 bpp. Zároveň nedisponuje příliš flexibilním způsobem nastavování kvality. Kvalitu lze ovlivnit pouze volbou nastavení kvantizační matice, v intervalu $\langle 1, 100 \rangle$ s krokem v celých číslech. Nelze přesně zadat zvolenou kvalitu, je nutno ji dopočítat aproximačním způsobem. V tomto testu byla dosažena přesnost aproximace na dvě desetinná místa, která pro ilustraci dostačuje. Z mnoha důvodů - DCT transformace, kódování - se nelze ani zdaleka přiblížit efektivitě komprese JPEG 2000. Průměrný rozdíl v kvalitě komprese při stejném bpp dosahuje **12.43%**. Při menších bpp je rozdíl ještě markantnější, kdy v intervalu $\langle 0.1, 0.5 \rangle$ bpp je rozdíl nezanedbatelných **26.74%**, což je zprůměrovaná hodnota pro všechny datasety. Největší rozdíl v tomto intervalu vykazuje dataset „Scany“ s více než o třetinu lepším kompresním výkonem (**35.81%**). Vzhledem k neschopnosti JPEG se neprojeví plné možnosti jeho nástupce. JPEG 2000 exceluje při velmi malých bitových hloubkách. Např. u datasetu „Fotografie“ odpovídá stejné PSNR jako u JPEG osmdesátinásobně menší bitové hloubce. Spolu s pořadím přenosu lze docílit při opravdu velkých kompresních poměrech alespoň náhledové kvality použitelné např. na webu.



Obrázek 4.2: Průběh kvality JPEG.

Kompresní artefakty se v malé míře objevují již při 0.5 bpp. Obraz je však stále dostatečně kvalitní i v plném rozlišení. Poloviční bpp vykazuje známky začínající posterizace, která je způsobena malým tonálním rozsahem. Minimální možná kvalita vykazuje značnou posterizaci. Takto zkomprimovaný obraz může sloužit maximálně jako malý náhled.



Obrázek 4.3: Průběh kvality JPEG 2000.

Tam, kde JPEG končí s nejmenší možnou nastavitelnou bitovou hloubkou JPEG 2000 začíná. 0.1 bpp znamená velmi kvalitní obraz, který lze použít i při plném rozlišení. Desetinásobně zmenšená bitová hloubka stále věrohodně reprezentuje barevné spektrum bez známek kompresních artefaktů. Rozumné minimum u JPEG 2000 se pohybuje hranici 0.001 bpp, kde je už silně znát velký kompresní poměr, obraz trpí značnou posterizací a je - i bez použití dlaždicového mechanismu - znatelné blokové dělení.

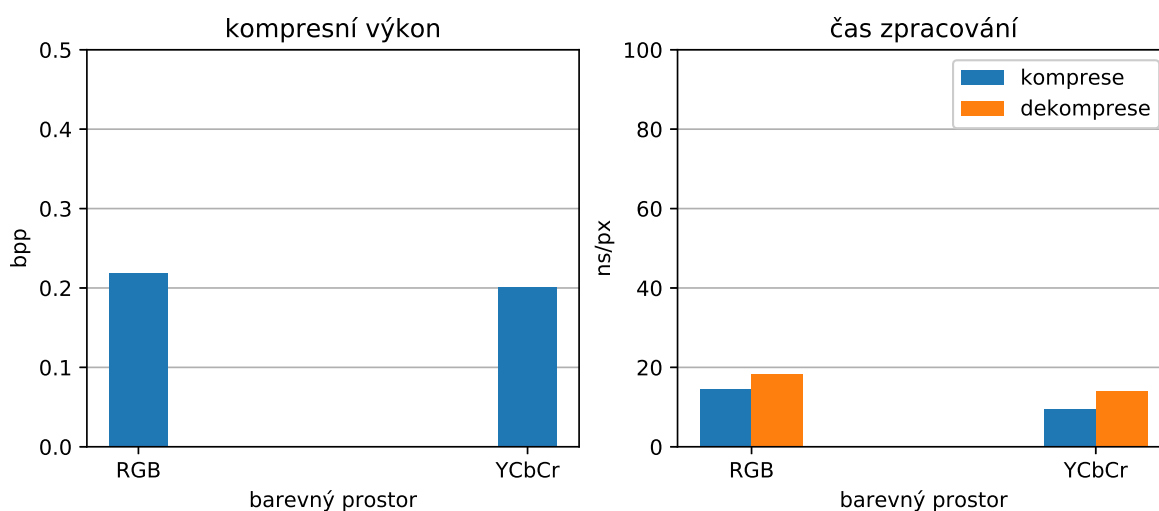
Časové a paměťové nároky jsou často udávány jako jedna z velkých nevýhod JPEG 2000. U první jmenované se neprokázal velký rozdíl, průměrně **9.71%**. Paměťové nároky ovšem mohou hrát roli při výběru formátu, průměrná hodnota maximální využití operační paměti při zpracování vzrostla u JPEG 2000 o **27.44%**. Tento fakt ovlivňuje mnoho mechanismů, které budou probrány v dalších sekcích (např. velikost dlaždice). Celkově lze JPEG 2000 doporučit. Dosahuje násobně lepších kompresních výkonů - hlavně u velkých kompresních poměrů - při celkovém patnáctiprocentním zvýšení režie.

4.2 Vyšetřování parametrů

Tato sekce se zaměřuje na testování parametrů pro ztrátovou i bezztrátovou kompresi. Měřítka grafů jsou vzájemně porovnatelná dle typů komprese.

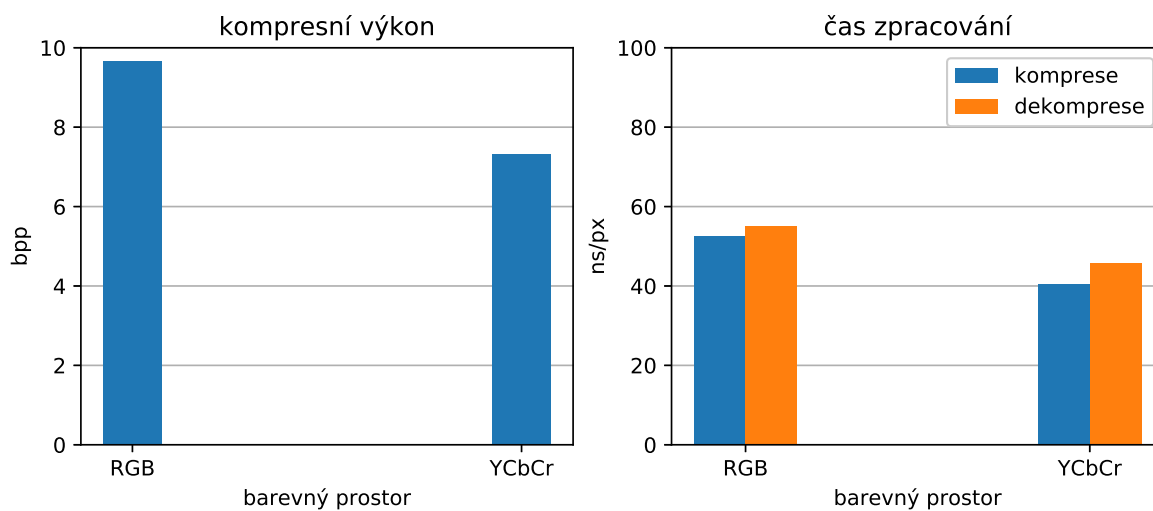
Převod barevného prostoru

Jedná se o první krok, který je nutno provést při zpracování vstupních dat. Používají se dvě cesty, ICT a RCT pro ztrátovou, resp. bezztrátovou kompresi. Všechny datasety jsou uloženy v RGB prostoru - kromě „Bitonální Gray8“, který v tomto testu nemá smysl vyšetřovat, jelikož není reprezentovaný ve třech, ale pouze v jedné složce - komponentě, podle terminologie JPEG 2000. Profily se tímto parametrem nezabývají, nelze využít referenčního nastavení. Podvzorkování je v při tomto testu vypnuto.



Obrázek 4.4: Převod barevného prostoru - ztrátově.

Nepoužití převodu barevného prostoru napříč datasety zhorší kompresní výkon o **9.78%** a zpomalí zpracování souboru o **26.77%**. Pro ztrátovou kompresi lze jednoznačně doporučit převod barevného prostoru.

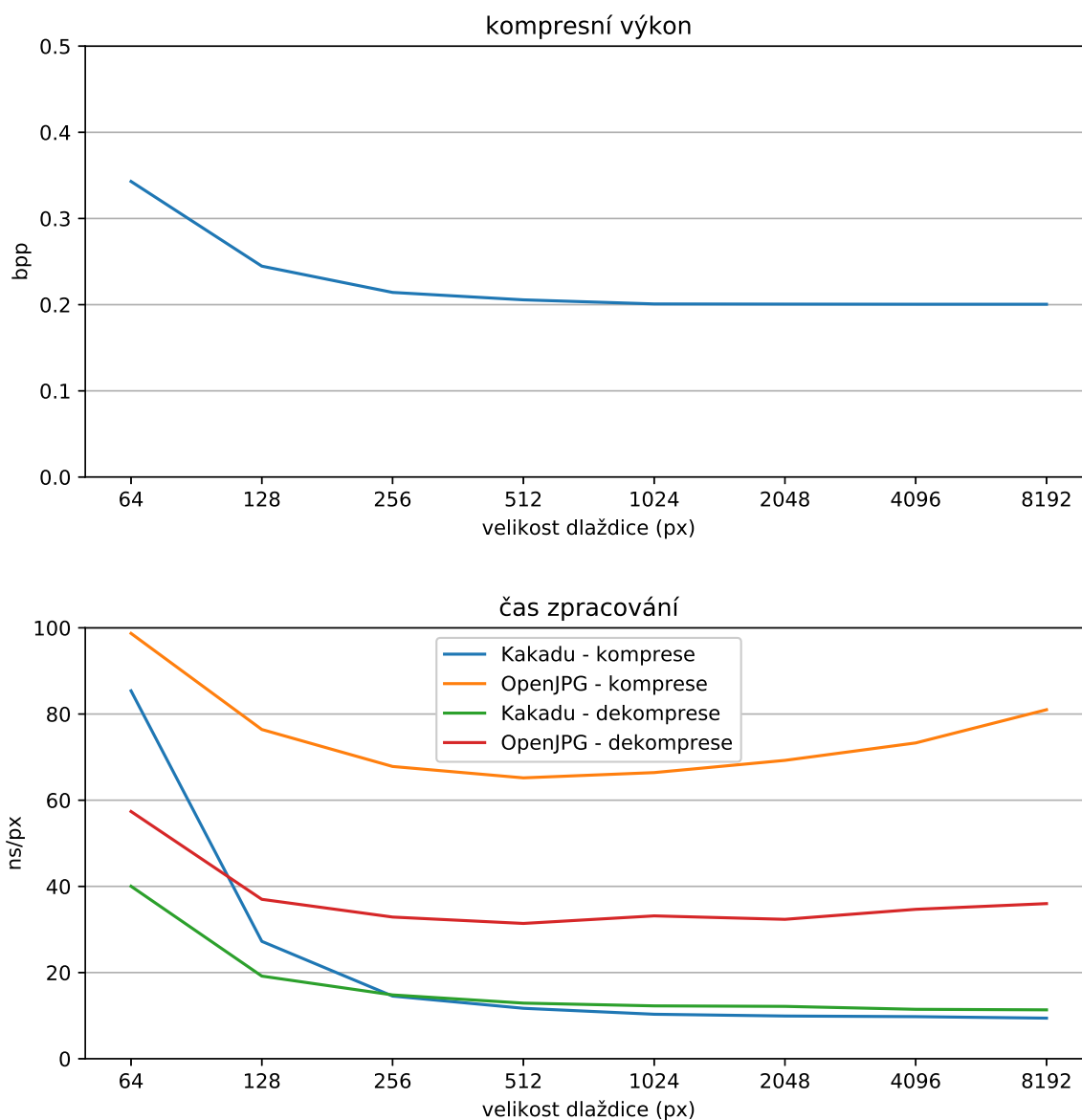


Obrázek 4.5: Převod barevného prostoru - bezztrátově.

Nepoužití převodu barevného prostoru napříč datasey zhorší kompresní výkon o **31.74%** a zpomalí zpracování souboru o **14.98%**. Pro bezztrátovou kompresi lze jednoznačně doporučit převod barevného prostoru. Jasně se ukázalo, že nepoužití převodu vede u obou typů kompresí ke zhoršení kompresního výkonu.

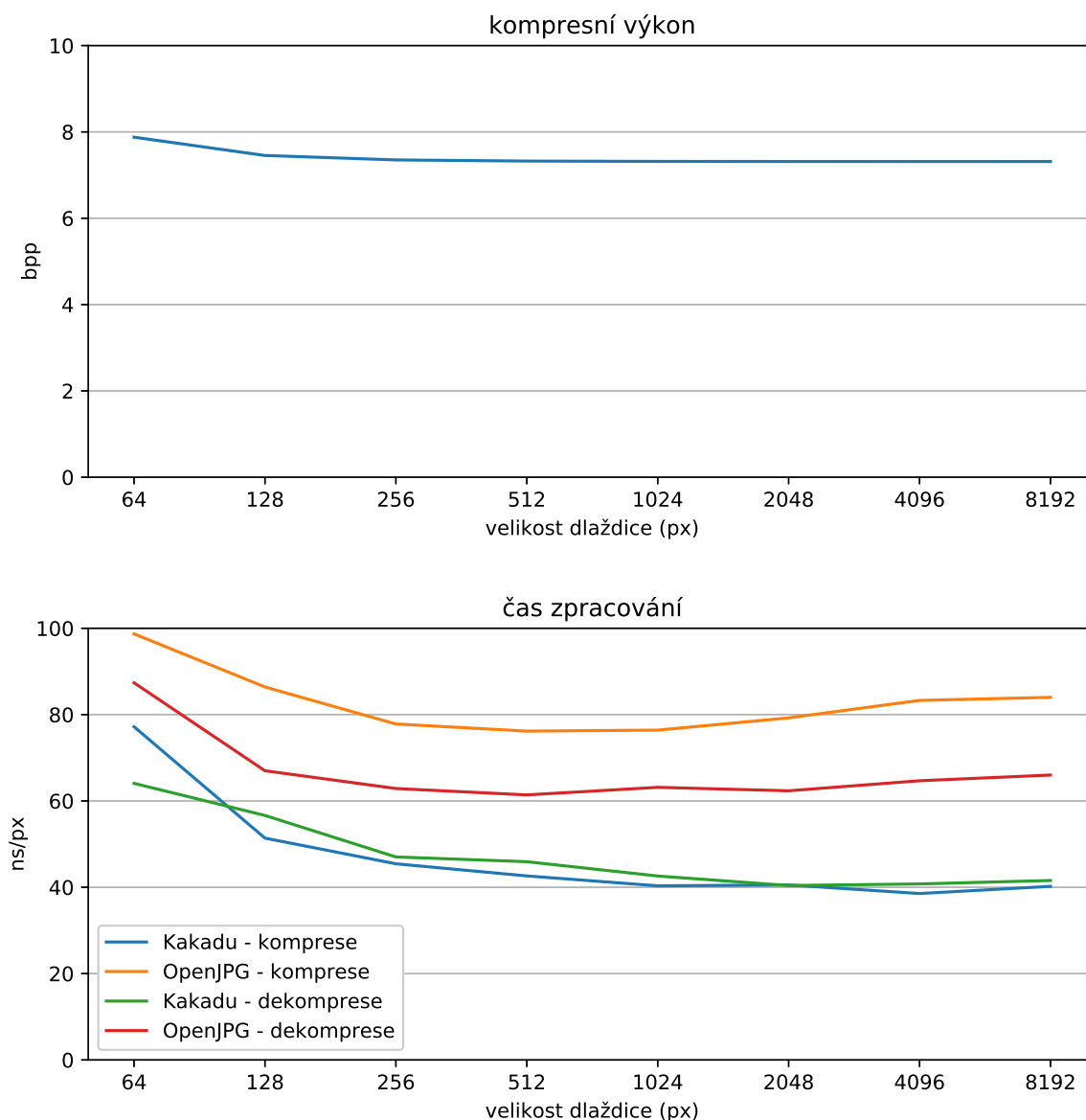
Dlaždice

Jeden ze základních parametrů nastavení umožňující plné využití formátu JPEG 2000. Obraz je rozdělen na části, které jsou samostatně zpracovávány. Tento krok dovoluje aplikovat ostatní parametry nikoliv na celý obraz, avšak pouze na dlaždici. V terminologii JPEG 2000 nazýváno *Tile-parts*. Typickým příkladem jsou tzv. *compound images*, které obsahují více typů obrazových dat v jednom obraze. Strojové vyšetřování tohoto nastavení je obtížné, jelikož se musí jednat o obraz, který obsahuje přesně definované části s rozdílným obsahem. Každá dlaždice má určitou režii - násobená volbou precinctu, při testování nepoužity - a při použití velmi malých dlaždic (128x128 px, 64x64 px) vznikají blokové kompresní artefakty. Velikosti dlaždic jsou při tomto testu čtvercové, ale lze zvolit libovolný poměr stran.



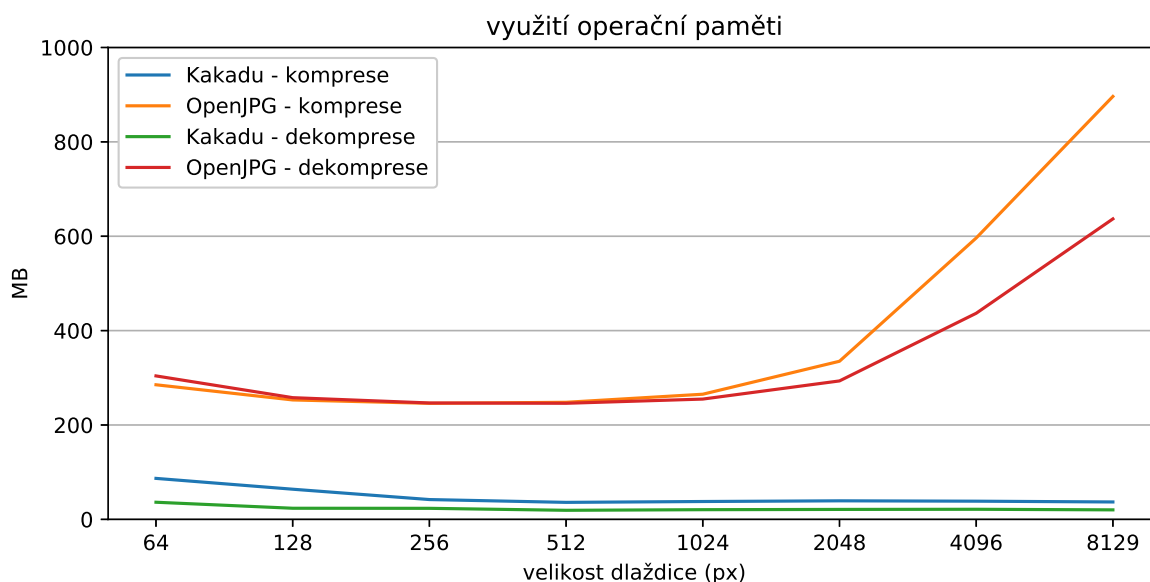
Obrázek 4.6: Velikost dlaždice - ztrátově.

U ztrátové komprese se doporučení profilů dělí na dva tábory. Pro „Fotografie“ a „Mapy“ dlaždice 1024 px a pro „Bitonální“ a „Scany“ vypnout systém dělení obrazu úplně. Kompresní výkon se ustaluje při 2048 px. Zvětšovat dlaždice za tuto úroveň má z hlediska velikost smysl pouze u obrazových dat, která obsahují alespoň desítky dlaždic při dané velikosti, což klade velké nároky na rozlišení. Datasety jsou navrženy na maximální měřitelnou velikost dlaždice 8192x8192 px. U takto velkých souborů režie dlaždice nehraje měřitelnou roli, proto lze pozorovat naprosto minimální rozdíl velikostí. Průměr rozdílu mezi velikostí dlaždice 2048x2048 px a maximální měřenou velikostí je u ztrátové komprese **0.27%**. Při volbě dlaždic o straně menší než 1000 px je ovšem rozdíl velikostí dramatický. Dlaždice o velikosti 64x64 px zhorší kompresní výkon o **69.33%** oproti referenční hodnotě. Z hlediska velikosti proto vyplývá, že pokud není rozumný důvod pro užití dlaždic je lepší je nepoužívat.



Obrázek 4.7: Velikost dlaždice - bezztrátově.

U bezztrátové komprese opět doporučují profily dvě nastavení. Pro „Fotografie“ a „Bitonální“ dlaždice 4096 px a pro „Mapy“ a „Scany“ vypnout systém dělení obrazu úplně. Závislost kompresního výkonu na dlaždici má zde podobný průběh jako u ztrátové komprese. Rozdíl mezi minimální (64x64 px) a doporučenou (4096x4096 px) je **9.44%**. Výpočetní náročnost je také podobná jako u ztrátové komprese. Kakadu se ustaluje pomaleji, OpenJPEG opět za hranici velikosti dlaždice 1024 px zvyšuje svoji časovou náročnost pro kompresi i dekompresi.



Obrázek 4.8: Vliv velikosti dlaždice na využití operační paměti.

Posledním pomocníkem, který pomůže rozhodnout o velikosti dlaždice je využití operační paměti. Ačkoliv se nejedná o tak uživatelsky postřehnutelný ukazatel jako je kompresní výkon nebo čas zpracování, je nutno jej také brát v potaz. Obě knihovny u nejmenších dlaždic (64 px a 128 px) využívají určité množství paměti, které přímo nesouvisí s velikostí dlaždice, dle mého odhadu se může jednat o alokaci paměti pro částečné simultánní zpracování. Následně se odehrává pokles. Kakadu se u hranice 512 px srovná a již další paměť nevyžaduje. OpenJPEG opět ukazuje své implantační slabosti. Od hranice 1024 px dramaticky stoupá využití paměti. Je pravděpodobné, že načítá celou dlaždici do paměti, kdežto Kakadu pravděpodobně pracuje se stejně velkými bloky nezávisle - od určité úrovně - na velikosti dlaždice. V extrémních případech OpenJPEG - nezobrazeno v grafu, obsahuje zprůměrovaná data všech datasetů - při kompresi s velikostí dlaždice 8912 px souborů s rozlišení 400 MPx a více při nezkomprimované velikosti 1-2 GB spotřeboval ve špičkách až **3.8 GB** operační paměti.

Kakadu lze doporučit při ztrátové kompresi velikost dlaždice **1024 px a výše** (zde se spíše odvíjí od velikosti samotného obrazu). Vypnutí systému dělení obrazu poskytuje zanedbatelné zrychlení v časové i velikostní doméně. To samé platí i pro bezztrátovou kompresi. U OpenJPEG lze vydat konkrétnější doporučení. Z hlediska kompresního výkonu se jeví opět 1024 px a více jako vhodná volba. Časová i paměťová náročnost o obou typech kompresí hovoří jasně v neprospěch zvyšování velikosti dlaždice za hranici **1024 px**. Vypnutí systému dělení u velkých obrazových dat způsobí to stejné, jako velká dlaždice, proto jej nelze u OpenJPEG doporučit pro žádný z datasetů.

Posun dlaždice a obrazu

Lze libovolně měnit začátek zpracovávaného obrazu i dlaždice. Vhodné je to například pro zajištění zpracování určitého oblasti v plné velikosti dlaždice. Ty mají jasně definovanou velikost, nicméně ty z kraje obrazu mají takovou velikost, jaká na ně při ořezu vyjde. Postupuje se od levého horního rohu. Tento parametr patří do specifických pro konkrétní obraz, proto byl vyšetřován pouze interně a jeho výsledky - zhoršení kvality při posunu dlaždice v jednotkách procent její velikosti - nelze reprezentativně vztáhnout na celé datasey.

Regiony zájmu

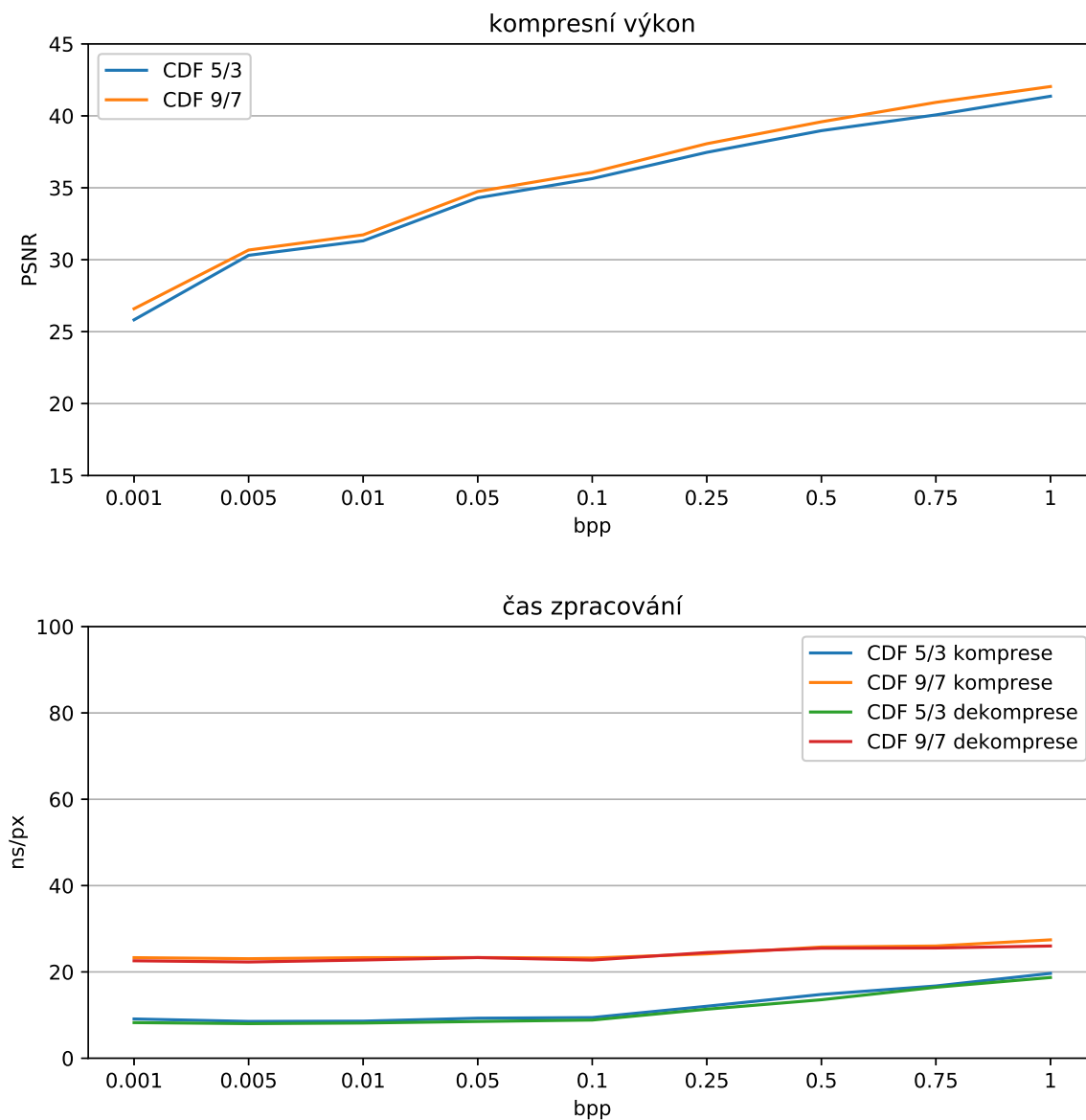
Další funkcionalita formátu JPEG 2000, která umožňuje zpracovat určitou část obrazu jinak než ostatní části [9]. Na rozdíl od dělení dlaždicemi jsou regiony zájmu méně flexibilnější nástrojem. Vznikají až v okamžiku provedení vlnkové transformace. Její koeficienty jsou v regionu zájmu „učiněny důležitějšími“ bitovým posunem vlevo, tak aby byly obsaženy ve vyšších bitových rovinách pro dřívější zakódování. Do výsledného souboru jsou poté uložena metadata umožňující zpětný posun koeficientů pro zpětné dekodování.



Obrázek 4.9: Regiony zájmu při postupném zvyšování bitové hloubky.

Vlnka

JPEG 2000 používá dvě hlavní vlnky pro vlnkovou transformaci. Obě jsou z rodiny biortogonálních spline vlnek. Většina knihoven volbu vlnky vůbec neumožňuje, implicitně používají CDF 5/3 pro bezztrátovou kompresi a CDF 9/7 pro ztrátovou. Kakadu umožňuje tuto volbu nechat na uživateli.

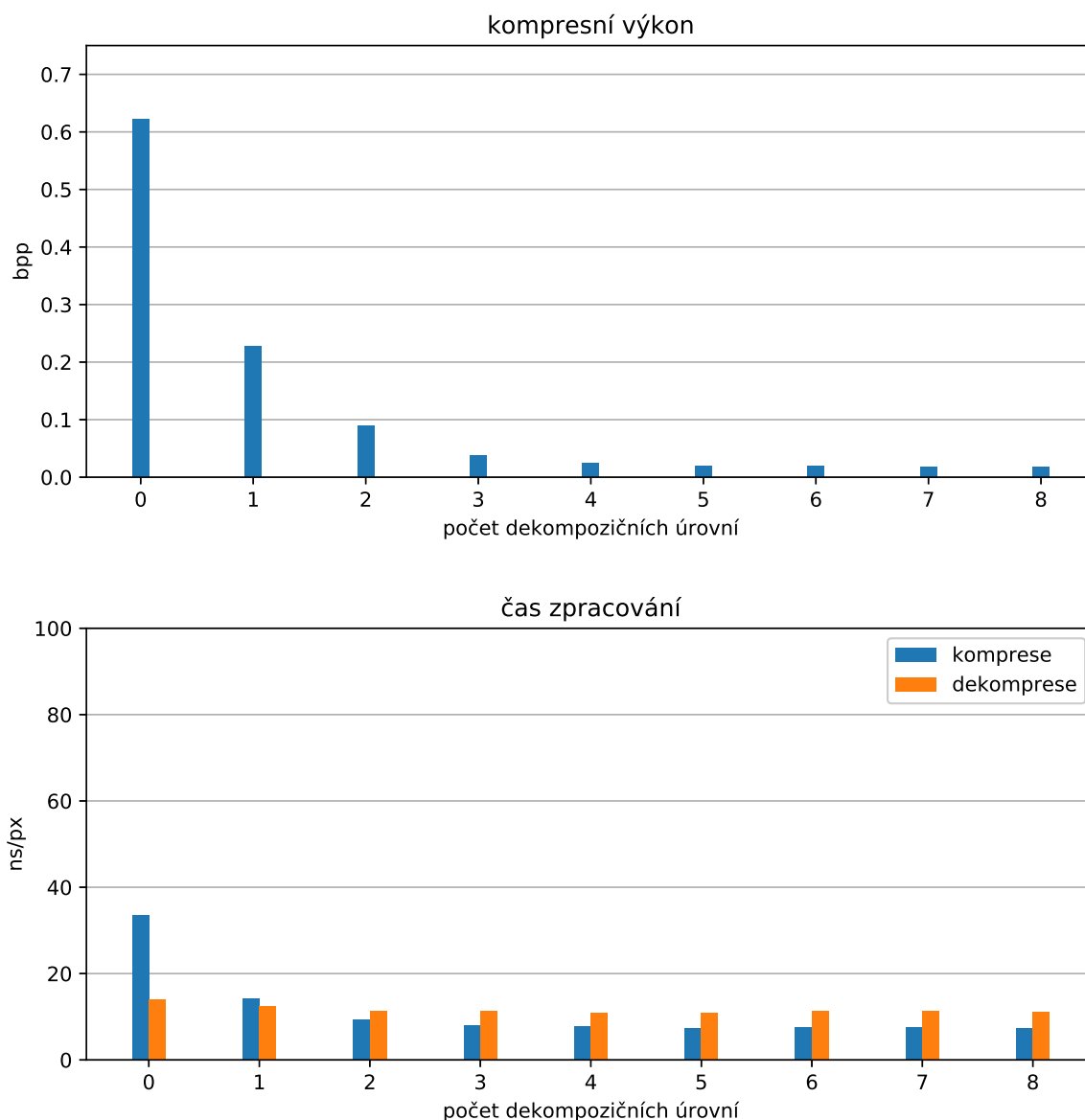


Obrázek 4.10: Vliv použité vlnky na transformaci při ztrátové kompresi.

Byl vyšetřen kompresní výkon ztrátové komprese vlnek běžně používaných pro oba typy kompresí. Kompresní výkon je očekávatelně lepší u CDF 9/7, která je určená pro tento typ zpracování, a to o **4.37%**. Mnohem zajímavější je rozdíl v čase komprese a dekomprese. U malých bitových hloubek je CDF 5/3 pro ztrátovou kompresi lepší, ale později se rozdíl smazává.

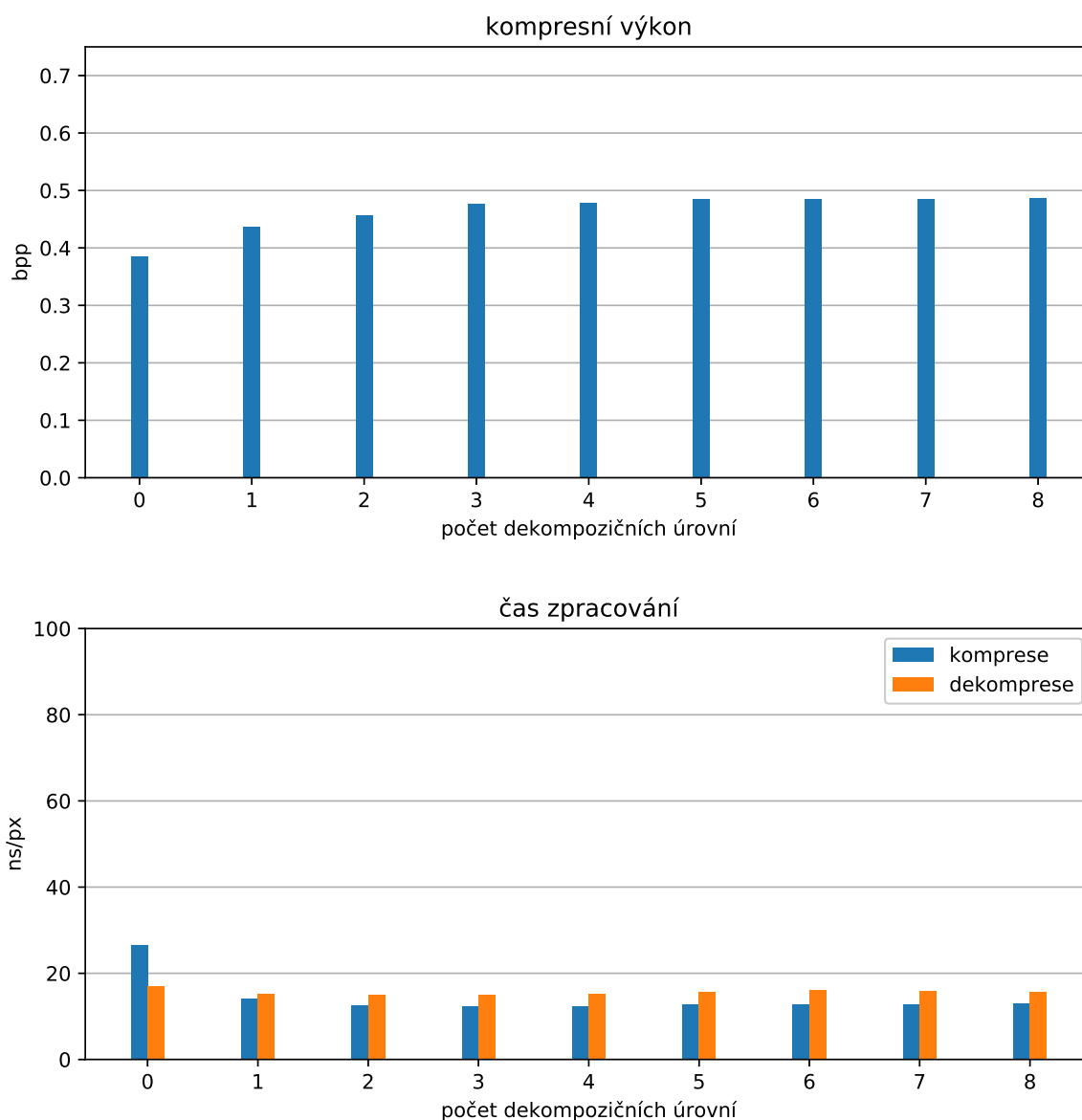
Dekompoziční úrovně

Další z parametrů, který má zásadní vliv na kompresní výkon. Určuje počet provedení dekompozice na pod pásma vstupního obrazu. Obor hodnot testování byl zvolen vzhledem k profilů - naprostá většina doporučuje hodnoty 5 až 7 - a smyslu vyšetřování, kdy je maximální počet dekompozičních úrovní určen velikostí obrazu dle mocniny s dvojkovým základem počtu úrovní násobeného velikostí code-bloku. Při šestiúrovňové dekompozici s velikostí code-bloku 64x64 je vyžadováno, aby měl obraz alespoň 4094x4096 px. Rozumné maximum vyšetřování je 8 dekompozičních úrovní. Obě knihovny se v kompresním výkonu shodují, stejně tak jako v průběhu časové náročnosti, s malými výjimkami, které jsou vysvětleny dále.



Obrázek 4.11: Dekompoziční úrovně datasetu „Fotky“.

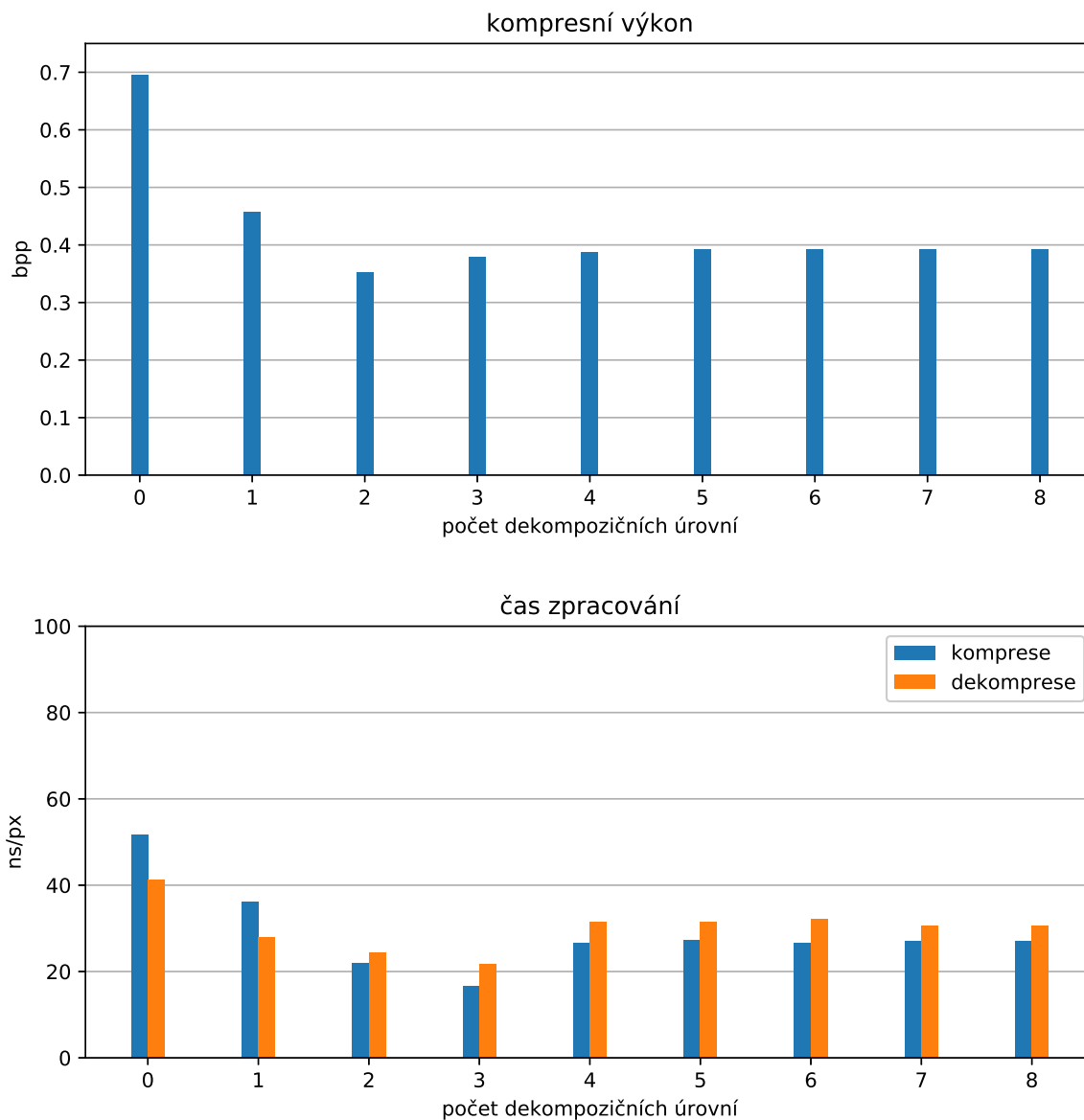
První částí je dataset „Fotografie“. Vlnková transformace je nutná pro tento typ obrazových dat. Profilem doporučených 5 úrovní zhruba odpovídá naměřeným výsledkům. Nejlepšího kompresního výkonu je dosaženo při 6 úrovních. Po překonání této hranice kompresní výkon mírně klesá, rozdíl mezi 6 a 8 úrovní je **3.4%**. Při pohledu do grafu časové náročnosti, lze předchozí verdikt potvrdit. Ačkoliv je graf sjednocen pro obě knihovny, OpenJPEG vykazuje u tohoto datasetu značné zhoršení časové náročnosti pouze při 8 úrovních. Důvodem je pravděpodobně kvalita implementace.



Obrázek 4.12: Dekompoziční úrovně datasetů „Bitonální RGB24“ a „Bitonální Gray8“.

„Bitonální“ obrazová data tvoří další část. Domněnka o relativně malém počtu úrovní se ukázala být správná, z testů nejlépe vyšla varianta neprovádět dekompozici. Profily se tentokrát příliš netrefily, ale to je logické vzhledem k exotičnosti dat. Po širším vyšetřování byla nalezena pravděpodobná příčina. Při zvolené dekompozici docházelo ke slévání jednotlivých

vých pixelů, kdy na malém prostoru může být ostrá změna jasu i po jednom pixelu. Časová náročnost mluví v neprospěch vynechání dekompozice, ale kompresní výkon je v tomto případě důležitější. Lze doporučit vynechání dekompozice.



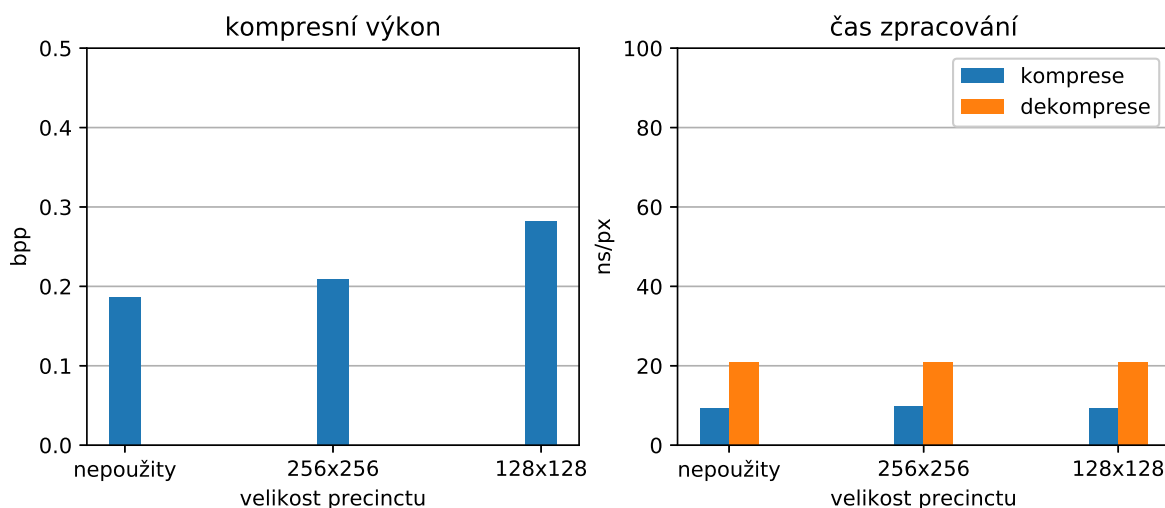
Obrázek 4.13: Dekompoziční úrovně datasetů „Mapy“ a „Scany“

Nakonec jsou vyšetřeny datasety „Mapy“ a „Scany“. Profily opět doporučují 5 až 6 úrovní. Skladba dat se oproti datasetu „Fotografie“ liší, avšak stále se jedná o třísloužkový obrazový materiál s dostatečnou diverzitou obsahu. Počet dekompozičních úrovní o hodnotě 2 se ukázal jako nejlepší. Je pravděpodobné, že tento typ obrazového materiálu sice má dostatečnou diverzitu, avšak je rozložena řídko po obraze. Text nebo rukopis na manuskriptu zabírá malou část obrazu, zbytek je papír, který tvoří relativně homogenní plochy, jejichž přílišná dekompozice pouze vytváří kompresní artefakty a nepřesnosti. Graf časové náročnosti signalizuje nejlepší výsledky u trojnásobné dekompozice (rozdíl **18.79%**), ale stejně

jako u předchozího vyšetřování má kompresní výkon přednost. Celkově lze doporučit nejvyšší počet dekompozičních úrovní u datasetu „Fotografie“ **6**, následované „Scany“ a „Mapy“ s **2** a „Bitonální“ **nerozkládat**.

Precincty

Naprostá většina profilů pracujících s první částí standardu JPEG 2000 buď nedefinuje jejich použití nebo - při použití alespoň třech dekompozičních úrovní DWT - volí velikosti 256×256 px pro první úroveň a 128×128 px pro zbylé. Vliv ztrátové a bezztrátové komprese nemá smysl vyšetřovat, jedná se pouze o organizaci výstupních dat.

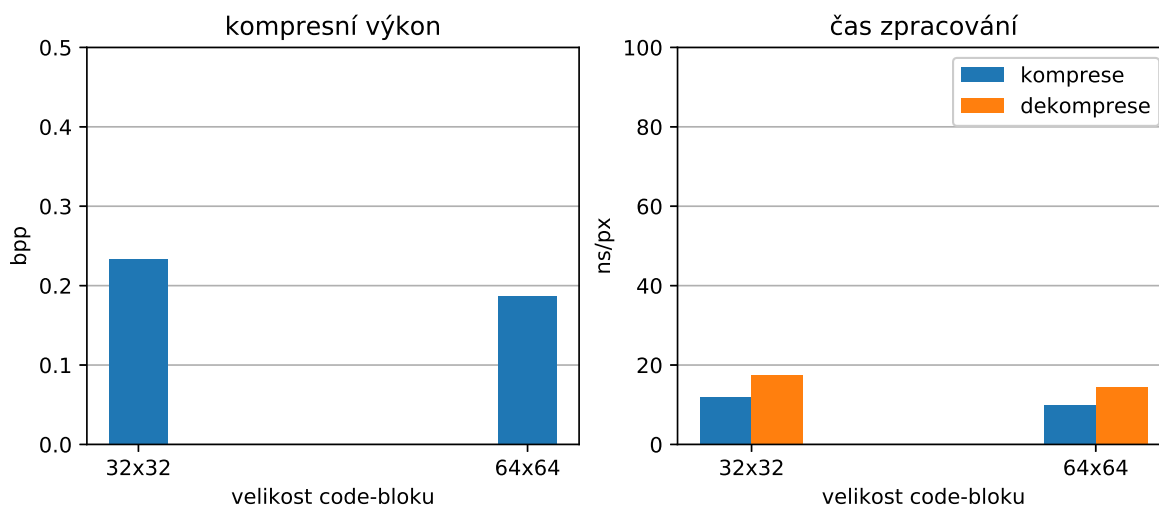


Obrázek 4.14: Vliv velikosti a použití precinctů.

Čím menší precinct, tím více paketů a tedy větší režie. Velikost 256×256 px zhorší kompresní výkon o **16.62%**, u dvojnásobně menších precinctů - a tedy dvojnásobně většího počtu paketů o **61.02%**. S počtem paketů roste poměr metadat na úkor užitečných dat.

Code-blok

Na rozdíl od precinctu se code-blok nemusí striktně držet čtvercových rozměrů, avšak je omezen explicitně daných počtem zpracovávaných hodnot (4096) a velikostí jedné strany (64 px). Drtivá většina dostupných profilů doporučuje maximální velikost 64x64, u bezztrátové komprese a datasetů profil pro „Mapy“ a „Scany“ doporučuje použití code-bloku 32x32 px.

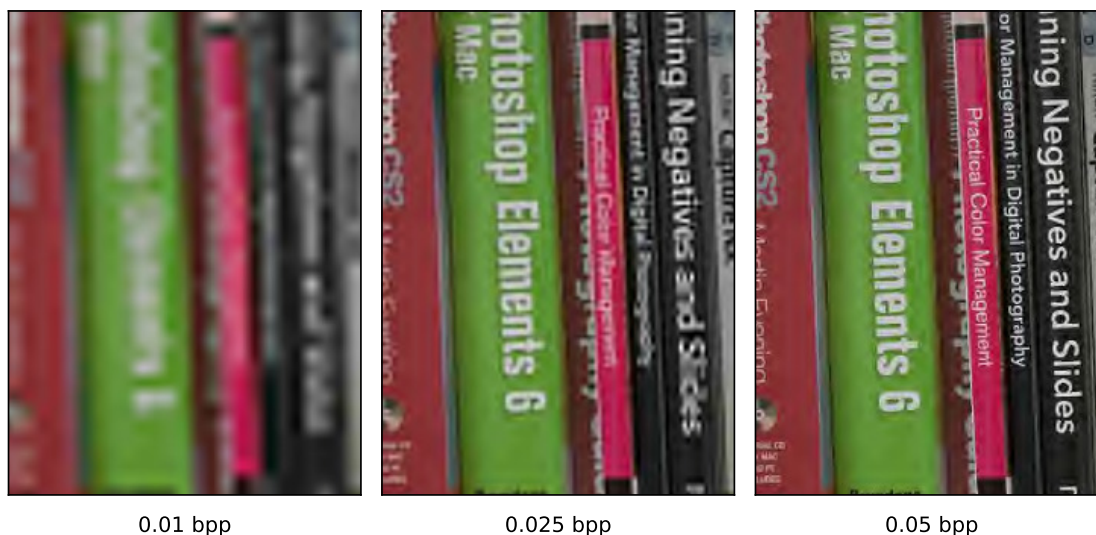


Obrázek 4.15: Vliv velikosti code-blok.

Při použití code-bloků velikosti 32x32 px se zhorší kompresní výkon o **27.44%** a časová náročnost vzroste o **19.71%**. Jiné velikosti než 64x64 px mají svůj význam při exotických velikostech precinctů nebo vstupních datech.

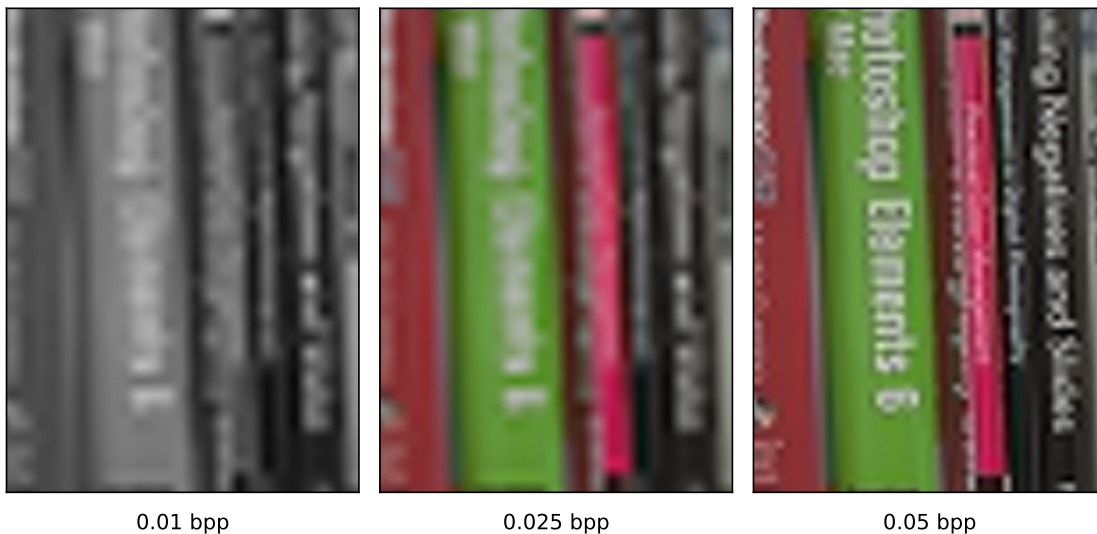
Pořadí přenosu

Další z parametrů, který přímo neovlivňuje kompresní výkon, ale určuje organizaci přenosu. Tato vlastnost je velmi žádána např. při přenosu po linkách s omezenou kapacitou. Uživatel je obslužen vizuálním vjemem, jehož kvalita se zvyšuje s rostoucím počtem obdržených dat. Tento parametr nebude exaktně vyšetřován, ale bude demonstrován jeho dopad v praxi. Definice kvality v tomto případě nahrazena čtyřmi měřítky, podle níž je kvalita škálována. První z nich je vrstva (*quality*). Pokud je přenášený obraz komprimován s vrstvami, jsou s postupným nárůstem přenesených dat protkávány a tím se zvyšuje kvalita. Další je rozlišení (*resolution*). Nejedná se o rozlišení ve smyslu dekompozičních vrstev, ale o rozlišení obrazu. Na začátek souboru je zakódován malý náhled. Rozlišení roste v násobcích dvou. Škálovat lze také pomocí pozice (*spatial location*). Pozicí jsou myšleny části obrazu, které se přenášejí za sebou shora dolů, vhodné například pro tiskárny a paměťově omezené aplikace. Poslední je komponenta (*component*), což jsou u vyšetřovaných datasetů složky barevného prostoru. Tento parametr je zadáván jako čtyřznaková kombinace. Např. LRCP, první písmeno znamená složku, která bude postupovat nejpomaleji a poslední nejrychleji.

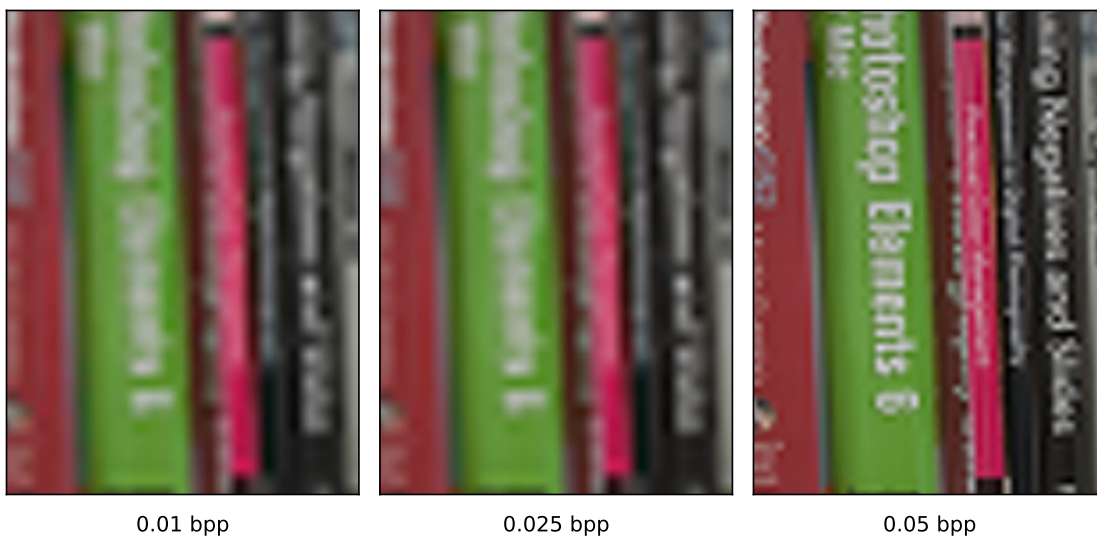


Obrázek 4.16: Pořadí přenosu *LRCP*.

U obou knihoven výchozí hodnota nastavení pro tento parametr. Nejrychleji roste pozice následována komponentou, rozlišením a vrstvou. Nastavení parametru zde vykazuje nejlepší výsledky. Celkem existuje pět voleb. Níže jsou rozebrány dvě nejčastější v profilech datasetů.



Obrázek 4.17: Pořadí přenosu *RPCL*.

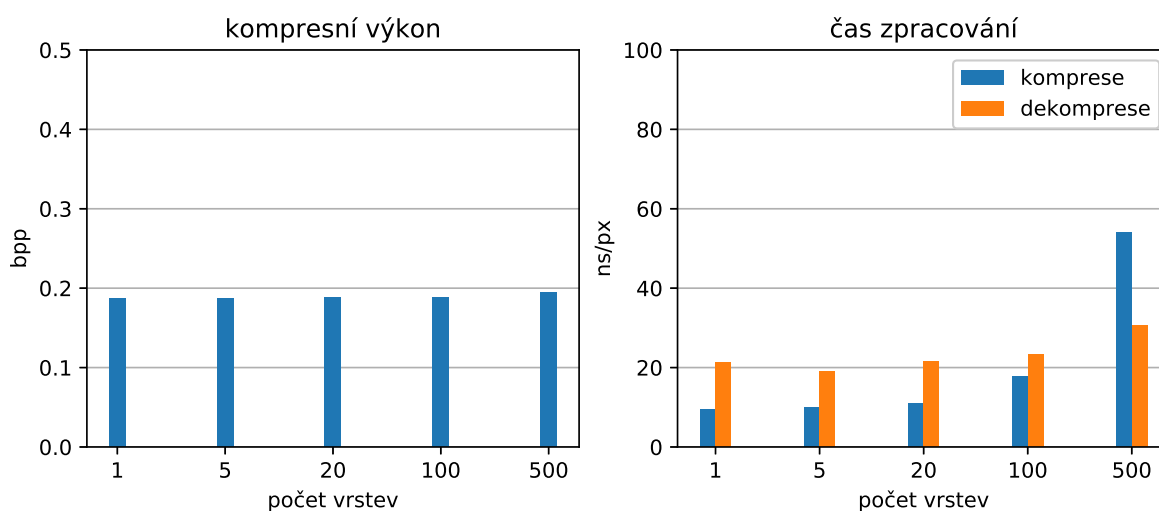


Obrázek 4.18: Pořadí přenosu *RLCP*.

Striktně dle výsledku vyšetřování komprese lze doporučit pořadí přenosu **LRCP**. Pořadí neovlivňuje kompresní výkon ani dobu nutnou pro kompresi, proto je toto doporučení volnější a záleží spíše na uživateli, jak hodlá zkomprimovaný soubor používat. Pouze dataset „Bitonální Gray8“ logicky vykazuje lepší výsledky při použití přenosu **CPRL**, jelikož komponenta je zde jenom jedna.

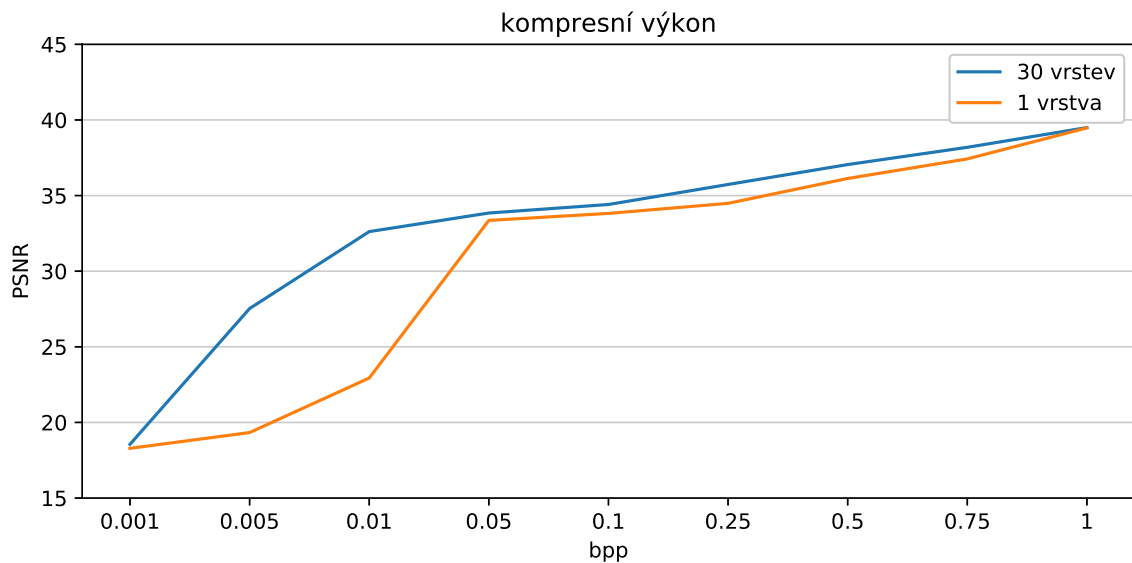
Vrstvy kvality

Slouží k uložení více kvalit obrazu do jednoho souboru. Jednotlivé bitové roviny code-bloků jsou v kódovacích průchodech algoritmu EBCOT prokládány pro určité kvality. Problém toho přístupu je opět - jako u precinctů - režie. Každý code-blok pro každou vrstvu kvality musí být jasně oddělen. V praxi se používají tři přístupy pro použití vrstev. První možnost je použít jenom jednu, což je případ profilů pro bezeztrátovou kompresi. Takto zpracované soubory většinou slouží jako archivní zdroje nebo zálohové soubory, ze kterých jsou tvořeny ztrátové kopie. U nich se využití vrstev vyplatí, stejně tak jako doporučují profily. Pokud jsou známy případy použití, lze ušít přímo na míru kvalitu vrstev (vyjmenovat konkrétní kvality). Tento přístup je neefektivnější. Poslední přístup je explicitně zadaný počet vrstev s logaritmickým rozložením, čímž je zajištěna škálovatelnost obrazu bez nutnosti znalosti přesných kvalit.



Obrázek 4.19: Vliv počtu vrstev na kompresní výkon a čas.

Paměťová režie se neukázala při velikosti code-bloku 64x64 px jako velký problém, i při stovkách vrstev kvality je nárůst do **5%**. Na co naopak počet vrstev má značný vliv je doba komprese. 20 vrstev představuje nárůst času nutného ke kompresi o **23.44%**, 100 vrstev více než dvojnásobek a 500 vrstev sedminásobek. Z výkonnostního hlediska lze doporučit maximálně **20 až 30** vrstev.

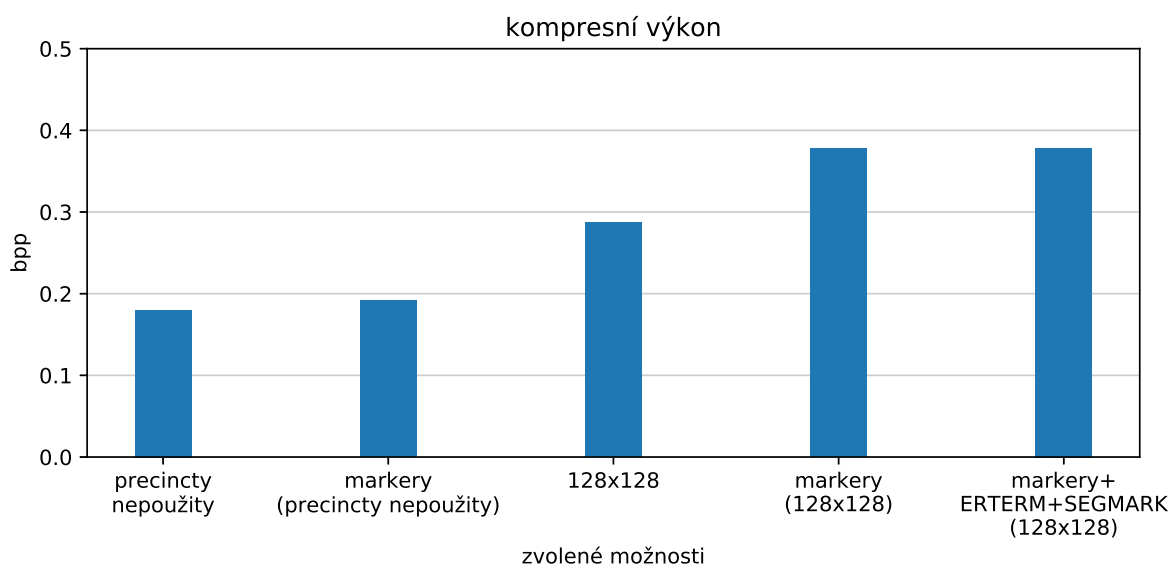


Obrázek 4.20: Průběh kvality pořadím přenosu LRCP s vrstvami a bez.

Z předchozího vyšetřování vyplynulo, o kolik při zvolené kvalitě dokáží metadata nafouknout datový tok. Tentokrát je vyšetřena kvalita v intervalu hodnot. Bylo použito 30 vrstev s logaritmickým rozsahem v intervalu $\langle 0.001, 1 \rangle$. Pro jednovrstvou kompresi bylo zvoleno 1 bpp. U malých bitových hloubek je rozdíl velmi znatelný, průměrně více než **85.23%**. S rostoucí bitovou hloubkou se rozdíl smazává. Profily pro ztrátovou kompresi používají maximálně 7 vrstev, kdy 5 vrstev tvoří kvalitní minimum pro škálovatelného obrazu. Při použití vrstev lze doporučit hodnotu **5 až 10** pro běžné případy.

Odolnost vůči chybám

Jako jakýkoliv jiný datový tok, jenž je přenášen po nespolehlivém přenosovém médiu - což je naprostá většina používaných koncovými zákazníky - je jej nutno zabezpečit proti chybám. JPEG 2000 nabízí v dalších částech standardu široké možnosti zabezpečení, nicméně cílem této není práce zevrubně zkoumat zabezpečení, proto jsou vyšetřeny jenom základní parametry. Nejjednodušším způsobem zabezpečení je označit paket. JPEG 2000 pro tuto činnost nabízí dva markery: SOP a EPH. První z nich značí začátek paketu (v datovém toku hodnota větší než FF8F H, což je maximální hodnota dvou datových bitů) a druhý konec. Dále lze instruovat kodér tak, aby používal módy určené pro zabezpečení. SEGMARK označí každou bitovou rovinu čtyřbitovou kombinací a ERTERM vynutí standardem definované ukončování zpracovávaných bloků. Výpočetní náročnost v tom případě nehraje roli, markery pouze zvětší velikost výsledného souboru a zvolením dva módy kodéru slouží pouze k označení významných míst v datovém toku.

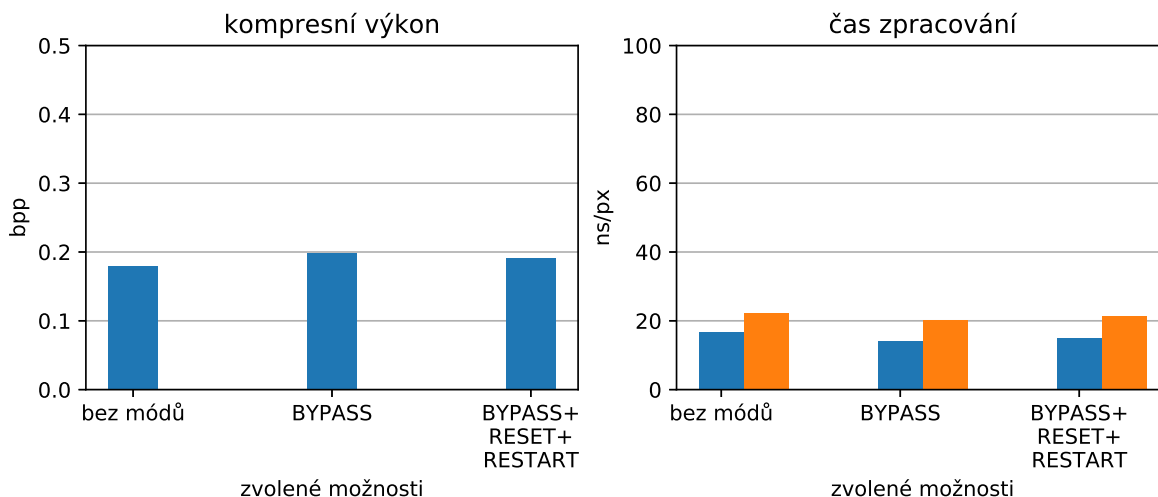


Obrázek 4.21: Vliv mechanismů zabezpečení na velikost souboru.

Nejprve byl zakódován soubor bez jakékoliv uživatelem vynutitelné ochrany proti chybám s rozdílnou velikostí precinctu ovlivňující generování paketů. Při označení paketů vzrostla velikost souboru o **6.48%**, respektive **17.41%** u velikosti precinctu 128x128 px. Použití módů kodéru nepřineslo měřitelný rozdíl. Profily zabezpečení nedefinují, proto lze doporučit užití módu ERTERM/SEGMARK u obou typů komprese. S markery je nutno být opatrnější. Zhorší kompresní výkon (u malých precinctů značně), proto je jejich užití doporučeno pouze v případě oprávněné obavy o chybný přenos.

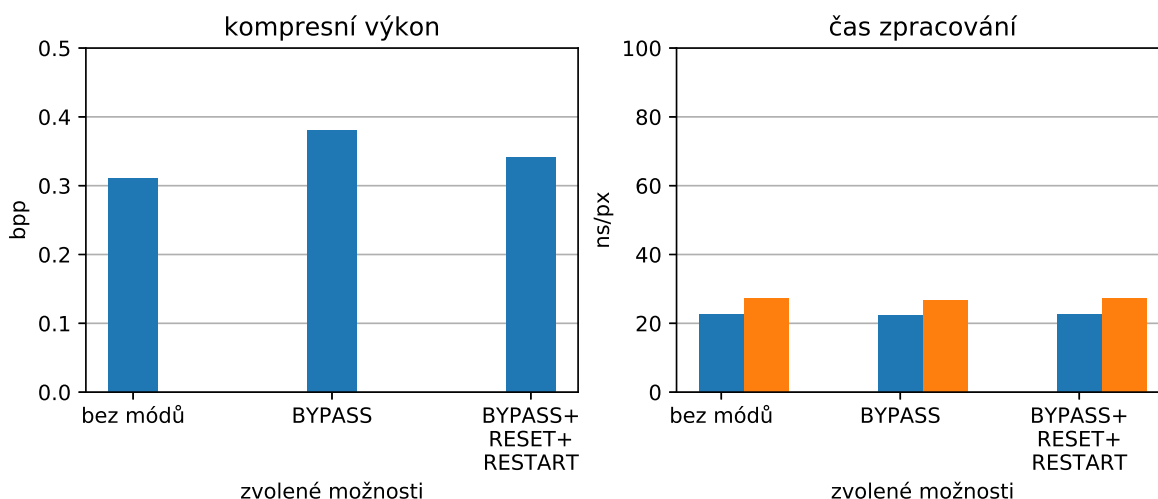
Redukce výpočetních nároků

Poslední částí testování je zjišťování vlivu módů kodéru na rychlost komprese a dekomprese. Prvním je BYPASS. Ten pracuje na principu přemostění MQ kodéru. Mělo by dojít ke snížení výpočetních nároků bez velké ztráty kvality. Další vyšetřované módy jsou RESET (vyčistí kontextové stavy po každém průchodu) a RESTART (celý kodér je restartován).



Obrázek 4.22: Vliv mechanismů redukce výpočetních nároků na kompresní řetězec.

Pro oba typy komprese a všechny datasets došlo ke zhoršení kompresního výkonu o **7.84%** pro mód BYPASS a **5.42%** pro módy BYPASS, RESET a RESTART. Redukce výpočetních nároků proběhla, avšak ne tak znatelně, o **8.95%** pro mód BYPASS a **3.82%** pro módy BYPASS, RESET a RESTART.



Obrázek 4.23: Vliv mechanismů redukce výpočetních nároků na kompresní řetězec „Bitonální“.

Po bližším zkoumání bylo odhaleno, že oba datasety „Bitonální“ vykazují známky výrazného zhoršení kompresního výkonu, konkrétně o **22.34%** při velmi malém zlepšení výpočetních nároků **0.88%**. Tento fenomén je pravděpodobně způsoben specifickým obsahem zmíněného datasetu. Celkově lze mód BYPASS doporučit pro oba typy kompresí a všechny datasety kromě „Bitonální“. Výhody (rychlost zpracování) převažují nevýhody (zhoršený kompresní výkon). Použití další dvou módů (RESET, RESTART) je více diskutabilní. Mírně lepší kompresní výkon, avšak výpočetním nárokům pomůžou minimálně.

4.3 Shrnutí testování

Byly vyšetřeny parametry pro ztrátovou i bezztrátovou kompresi pomocí knihoven Kakadu a OpenJPEG. Většina vyšetřování měla za cíl určit vhodnou hodnotu, či obor hodnot, která je vzhledem ke kompresnímu výkonu, časové a paměťové náročnosti nejlepší. U některých typů vyšetřování ovšem toto postrádalo smysl, a proto byla zvolena forma demonstrace funkcionality (např. ROI, nastavení kompresní vlnky). Nejprve byl vyšetřen rozdíl mezi formáty JPEG a JPEG 2000, kdy druhý jmenovaný jasně ukázal svoji technickou vyspělost, hlavně při velkých kompresních poměrech. Poté následovalo testování parametrů. Převod barevného prostoru ze vstupního RGB do YCbCr se ukázal jako nutný pro oba druhy komprese a všechny datasety, kromě „Bitonální Gray8“, kde kvůli počtu barevných komponent nebylo potřeba tento parametr vyšetřovat. Co ovšem měl smysl vyšetřovat u všech datasetů byla velikost dlaždice. Zde se jasně ukázaly nectnosti implementace OpenJPEG z oblasti časové a paměťové domény, které jasně definovaly, jakou volbu parametru použít. Kakadu naopak využívalo paměť i čas očekávaně. Posun dlaždice a obrazu se neukázaly jako vhodné k zevrubnému vyšetřování. U regionů zájmu byla demonstrována pouze jejich funkčnost, jelikož záleží na konkrétním vstupním obrazu, jak ovlivní kompresní výkon a časovou/paměťovou náročnost. Další z teoreticky vyšetřených parametrů byl typ vlnky při ztrátové kompresi. Tento test mohl být proveden pouze v Kakadu, jelikož OpenJPEG tuto volbu neumožňuje. Vlnka pro bezztrátovou kompresi dosahovala horších kompresních výkonů, ale menší časové náročnosti. Dekompoziční úrovně opět ukázaly rozdíly mezi datasety, kde „Fotografie“ vykazovaly nejlepší výsledky při použití šestinásobného rozkladu, ale „Bitonální“ naopak nevyžadují vůbec žádný rozklad. Následovala dvojice precincty a code-bloky, kde byly výsledky očekávatelné, ale nebylo na škodu ukázat, proč se dané hodnoty používají. Pořadí přenosu bylo opět spíše teoretické vyšetřování, kdy z části záleží na volbě uživatele a případu použití dat. Co lze ovšem skoro paušálně doporučit jsou vrstvy kvality. Režie je při jednotkách až desítkách vrstev únosná, pouze při bezztrátové kompresi archivních originálů má smysl tento mechanismus vypnout. Zlepšení kompresního výkonu při použití vrstev je velmi znatelné při postupné dekompresi velkých kompresních poměrů. Nakonec by vyšetřen kodér, a to ve dvou testech. První se zabýval zabezpečením. Zde se ukázalo, že použití módů pro zabezpečení je velmi levné z hlediska kompresního výkonu. To samé se ale nedá říci o markerech pro označení paketu. Ty přinášejí značnou režii, kterou je nutno akceptovat pro zabezpečení dat. Druhý se zabýval redukcí výpočetních nároků. Mód BYPASS ukázal svoji sílu při zmenšení časové náročnosti na úkor zhoršení kompresního výkonu. U datasetu „Bitonální“ produkoval velmi špatné výsledky, proto jej nelze doporučit plošně. Během testu byly vyšetřeny další módy pro redukcí výpočetních nároků (RESTART a RESET). Kakadu bylo při testování průměrně **2.4x** rychlejší než OpenJPEG. Paměťové nároky byly u Kakadu zhruba **poloviční**, pouze v extrémních případech - dlaždice - rostly exponenciálně. Některé parametry nešlo vyšetřit za využití obou typů knihoven z důvodu neúplného pokrytí standardu.

Kapitola 5

Závěr

Práce je složena ze tří částí. V kapitole nazvané *JPEG 2000* proběhlo seznámení se s formátem, kdy jeho předchůdce JPEG posloužil jako odrazový můstek. Byla vysvětlena většina klíčových principů, aby se čtenář dokázal orientovat v problematice při vyšetřování parametrů nastavení. Pro vyšetřování bylo sestaveno celkem 5 datasetů - „Fotografie“, „Scany“, „Mapy“ a „Bitonální“ ve dvou variantách (RGB24 a Gray8) - s odlišnými charakteristikami a rozdílnými případy použití.

Druhá část práce se v kapitole *Metodika testování* zabývá principy testování. Jsou vysvětleny volby a typy použitých kritérií (kompresní výkon, čas a paměť). Následuje podrobný náhled na návrh a implementaci testovacího nástroje.

Těžiště této práce se nachází v kapitole *Analýza nastavení formátu*. Nejprve jsou získány obecně používané profily pro kompresi ve formátu JPEG 2000. Vyšetřování je děleno na dva typy. První typ jsou parametry, které lze zobecnit na celý dataset a tudíž reprezentativně vyšetřit. Existují i parametry, které nelze paušálně vztáhnout na celý dataset. U nich bylo demonstrováno možné použití a vydáno doporučení pro uživatele. Každé vyšetřování je shrnuto a na jeho konci vydáno doporučení pro uživatele.

Literatura

- [1] Kakadu software. [Online; navštíveno 05.10.2018].
URL <http://kakadusoftware.com/documentation/>
- [2] OpenJPEG. [Online; navštíveno 05.10.2018].
URL <https://www.openjpeg.org>
- [3] Bibliotheca Alexandrina: Wellcome Arabic Manuscripts Cataloguing Project. [Online; navštíveno 22.02.2019].
URL <https://wellcomelibrary.org/content/documents/22082/JPEG2000-preservation-format.pdf>
- [4] British Library: British Library JPEG 2000 profile. [Online; navštíveno 22.02.2019].
URL <http://www.digitizationguidelines.gov/still-image/documents/Martin.pdf>
- [5] Brower, B.: JPEG 2000 Profile. [Online; navštíveno 5.10.2018].
URL <http://www.gwg.nga.mil/ntb/2002SICS/10CompProf.PDF>
- [6] David S. Taubman, M. W. M.: *Image Compression Fundamentals, Standards and Practice*. Springer Science+Business Media, LLC, 2002, ISBN 978-1-4613-5245-7.
- [7] Elena Anisimova, J. B.: Image processing using the wavelet transform. *Elektrorevue*, ročník 15, č. 4, 2013, ISSN 1213-1539.
- [8] Impoco, G.: JPEG2000 - A Short Tutorial. [Online; navštíveno 5.10.2018].
URL http://www.impoco.it/files/impoco_2004_tutorial_JPEG2000.pdf
- [9] Joel Askelöf, C. C., Mathias Larsson Carlander: Region of interest coding in JPEG 2000. *Signal Processing: Image Communication*, ročník 17, č. 1, 2002, doi:10.1016/S0923-5965(01)00026-1.
- [10] J.Scott Houchin, D. W. S.: File format technology in JPEG 2000 enables flexible use of still and motion sequences. *Signal Processing: Image Communication*, ročník 17, č. 1, 2002, doi:10.1016/S0923-5965(01)00023-6.
- [11] Library of Congress: JPEG 2000 Profile for the National Digital Newspaper Program. [Online; navštíveno 03.03.2019].
URL http://www.loc.gov/ndnp/guidelines/docs/NDNP_JP2HistNewsProfile.pdf
- [12] Michael W.Marcellin, A. B., Margaret A.Lepley: An overview of quantization in JPEG 2000. *Signal Processing: Image Communication*, ročník 17, č. 1, 2002, doi:10.1016/S0923-5965(01)00027-3.

- [13] Národní digitální knihovna: Standardy pro obrazová data. [Online; navštíveno 14.01.2019].
URL
<https://www.ndk.cz/standardy-digitalizace/standardy-pro-obrazova-data>
- [14] S-Medouakh, Z.-E. B.: Entropy Encoding EBCOT (Embedded Block Coding with Optimized Truncation) In JPEG2000. *IJCSI International Journal of Computer Science*, ročník 8, č. 4, 2011, ISSN 1694-0814.
- [15] Schubert, E. F.: *Human eye sensitivity and photometric quantities*. Cambridge University Press, 2006, ISBN 0521865387.
- [16] Wallace, G.: The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, ročník 38, č. 1, 1992, ISSN 1558-4127, doi:10.1109/30.125072.

Příloha A

Kompresní parametry

Kakadu	OpenJPEG	Význam
-fprec -frag -icrop -flush_period -com -no_weights -rgb_to_420, Cyclic -num_threads -roi -rate Cversible	-F -C -ROI -r -q -I	Dodání explicitních informací o obrázku (<i>OpenJPEG</i>). Práce s bitovou hloubkou u komponent vstupního souboru. Pouze část obrazu ke kompresi. Ořez obrazu. Standardně se čeká na kompresi všech dat (obrazu), vynucení iterativního zápisu souboru. Komentář přímo do datového toku. Nepoužívat implicitní váhy barevných RGB koeficientů. RGB do Luminance/Chrominance (druhé inline implicitně nastavené) Počet vláken u Kakadu. Regiony zájmu. Nastavení kompresní poměrů, (nelze kombinovat s -q). Kvality udávané v PSNR (nelze kombinovat s -r, pouze <i>OpenJPEG</i>). Ztrátovost komprese, ztrátová (<i>CDF 9-7</i>), bezztrátová (<i>CDF 5-3</i>).
Cuse_precincts Cprecincts Cblk Ctiles Ctile_origin Sorigin Clevels	-c -b -t -T -d -n	Použití bloků transformačních koeficientů u Kakadu. Čtvercové bloky transformačních koeficientů. Část pásma bloků transformačních koeficientů. Velikost dlaždice, po kterých je obraz zpracováván. Offset dlaždice. Offset obrázku. Počet dekompozičních úrovní (<i>OpenJPEG +1</i>).
CLayers Corder Porder Ssampling Cmodes	 -p -POC -s -M	Počet vrstev kvality (implitně specifikováno počtem kompresí/kvalit). Pořadí operací (L=vrstva; R=rozlišení; C=komponenta; P=pozice). Změna pořadí operací na úrovni dlaždic. Nastavení vzorkování (Kakadu umí pro každý datový tok, OpenJPEG paušálně). Nastavení módu EBCOT.
Cuse_sop Cuse_eph	-SOP -EPH	Zápis SOP (označení resynchronizace) před každý paket. Zápis EPH (end of packet header) před každou hlavičku paketu.

Příloha B

Dekompresní parametry

Kakadu	OpenJPEG	Význam
-num_threads	-threads	Počet vláken.
-rate		Dekomprese při nastavené kvalitě.
-region	-d	Explicitní výběr části obrazu k dekompresi.
-layers	-l	Počet vrstev k dekódování.
-reduce	-r	Počet rozlišení k zahození.
-no_alpha		Bez vrstev průhlednosti.