



**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

**LUCIE VRTĚLOVÁ**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. DAVID BAŘINA, Ph.D.**

BRNO 2017

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav počítačové grafiky a multimédií

Akademický rok 2017/2018

**Zadání bakalářské práce**

Řešitel: **Vrtělová Lucie**

Obor: Informační technologie

Téma: **Analýza nastavení formátu JPEG 2000**

**Analysis of JPEG 2000 Settings**

Kategorie: Zpracování obrazu

Pokyny:

1. Seznamte se s formátem JPEG 2000. Zaměřte se na různá jeho nastavení a profily.
2. Sestavte několik datových sad odlišného charakteru (např. manuskripty, družicové snímky, běžné fotografie).
3. Proveďte analýzu vhodného nastavení formátu pro jednotlivé datové sady. U ztrátové komprese zkoumejte závislost kvality na datovém toku.
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

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Bařina David, Ing., Ph.D.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2017

Datum odevzdání: 16. května 2018

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav počítačové grafiky a multimédií  
602 00 Brno, Božetěchova 2



---

doc. Dr. Ing. Jan Černocký  
vedoucí ústavu

## Abstrakt

Tato práce řeší analýzu jednotlivých nastavení formátu JPEG 2000. Cílem této práce prozkoumat a porovnat jednotlivá nastavení a implementaci v různých knihovnách. Analýzu jsem provedla na dvou knihovnách implementujících JPEG 2000 - Kakadu a OpenJPEG. Data jsem zpracovávala pomocí frameworku napsaného v jazyce Python3. Vytvořené řešení poskytuje ucelený pohled na použití jednotlivých nastavení JPEG 2000 nad různými daty. Výsledky této práce umožňují rychlejší použití JPEG 2000 na konkrétních datech.

## Abstract

This thesis is focused on analysis of JPEG 2000 settings. The aim of this thesis is to explore and compare each settings and different implementation. The analysis was done with two libraries, which implements JPEG 2000 - Kakadu and OpenJPEG. Data was processed by framework written in Python3. Created solution provides comprehensive view on using each settings of JPEG 2000 on specific data. Results of this thesis allow faster using of JPEG 2000 on specific data.

## Klíčová slova

JPEG, JPEG 2000, komprese dat, bezstrátová komprese, ztrátová komprese, Kakadu, OpenJPEG, Python

## Keywords

JPEG, JPEG 2000, data compression, lossless compression, lossy compression, Kakadu, OpenJPEG, Python

## Citace

VRTĚLOVÁ, Lucie. *Analýza nastavení formátu JPEG 2000*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Bařina David.

# 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.

.....  
Lucie Vrtělová  
17. května 2018

## Poděkování

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

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Kompresie obrazu</b>	<b>5</b>
2.1	Barevné modely . . . . .	5
2.2	Současný přístup ke kompresi obrazu . . . . .	6
2.3	Přínos formátu JPEG 2000 . . . . .	7
<b>3</b>	<b>JPEG 2000</b>	<b>9</b>
3.1	Popis funkcionality kodeku JPEG 2000 . . . . .	9
3.1.1	Diskrétní vlnková transformace . . . . .	10
3.1.2	Kvantizace . . . . .	11
3.1.3	EBCOT . . . . .	11
3.2	Srovnání JPEG a JPEG 2000 . . . . .	12
<b>4</b>	<b>Implementace</b>	<b>13</b>
4.1	Práce s knihovnou OpenJPEG . . . . .	13
4.2	Práce s knihovnou Kakadu . . . . .	13
4.3	Popis frameworku . . . . .	14
4.4	Analýza dat . . . . .	14
4.4.1	Použité srovnávací metody . . . . .	14
4.4.2	Vykreslení grafů . . . . .	15
<b>5</b>	<b>Analýza nastavení JPEG 2000 pro bezztrátovou kompresi</b>	<b>16</b>
5.1	Dataset č.1 - 6 Mpx fotografie . . . . .	17
5.1.1	Knihovna OpenJPEG . . . . .	17
5.1.2	Knihovna Kakadu . . . . .	22
5.1.3	Shrnutí . . . . .	26
5.2	Dataset č.2 - Skenované časopisy . . . . .	26
5.2.1	Knihovna OpenJPEG . . . . .	26
5.2.2	Knihovna Kakadu . . . . .	31
5.2.3	Shrnutí . . . . .	35
5.3	Dataset č.3 - Skenované mapy . . . . .	35
5.3.1	Knihovna OpenJPEG . . . . .	35
5.3.2	Knihovna Kakadu . . . . .	40
5.3.3	Shrnutí . . . . .	45
5.4	Dataset č.4 - Skenované knihy . . . . .	45
5.4.1	Knihovna OpenJPEG . . . . .	45
5.4.2	Knihovna Kakadu . . . . .	50

5.4.3	Shrnutí	55
<b>6</b>	<b>Analýza nastavení JPEG 2000 pro ztrátovou kompresi</b>	<b>56</b>
6.1	Dataset č.1 - 6 Mpx fotografie	57
6.1.1	Knihovna OpenJPEG	57
6.1.2	Knihovna Kakadu	62
6.1.3	Shrnutí	67
6.2	Dataset č.2 - Skenované časopisy	67
6.2.1	Knihovna OpenJPEG	67
6.2.2	Knihovna Kakadu	72
6.2.3	Shrnutí	77
6.3	Dataset č.3 - Skenované mapy	77
6.3.1	Knihovna OpenJPEG	77
6.3.2	Knihovna Kakadu	82
6.3.3	Shrnutí	87
6.4	Dataset č.4 - Skenované knihy	87
6.4.1	Knihovna OpenJPEG	87
6.4.2	Knihovna Kakadu	92
6.4.3	Shrnutí	97
<b>7</b>	<b>Závěr</b>	<b>98</b>
	<b>Literatura</b>	<b>99</b>
	<b>Přílohy</b>	<b>101</b>
<b>A</b>	<b>Manuál</b>	<b>102</b>

# Kapitola 1

## Úvod

Kompresa dat je zpracování počítačových dat, jejímž úkolem je zmenšit velikost datových souborů, což je výhodné pro jejich archivaci nebo pro přenos přes síť s omezenou rychlostí (snížení doby nutné pro přenos). Kompresa může být nutná při omezené datové propustnosti, např. mobilní telefon komprimuje hovor pro přenos GSM sítí [2].

Jak bylo řečeno výše, hlavní motivací pro zavedení komprese je snížení zdrojů potřebných pro uložení a přenos dat. Výpočetní zdroje jsou spotřebovávány v kompresním režimu a obvykle ve zpětném režim (dekompresi). Návrh komprese dat zahrnuje kompromisy mezi různými faktory, včetně stupně komprese, množství zavedeného zkreslení (při použití ztrátové komprese dat) a výpočetních prostředků potřebných pro kompresi a dekompresi dat. V současné době je ztrátová komprese nejčastěji prováděna pomocí formátu JPEG. Tento formát pracuje na principu diskrétní kosinové transformace. Formát JPEG je velice rozšířený, což jeho potenciálnímu nástupci - formátu JPEG 2000, nijak neulehčuje jeho možné nahrazení. JPEG 2000 navíc není zpětně kompatibilní. Ačkoli se může zdát zavedení nového formátu zbytečně pracné, má jisté vlastnosti, díky kterým by bylo vhodné jej prozkoumat. JPEG 2000 plně podporuje jak ztrátovou, tak bezztrátovou kompresi. Během jednoho datového přenosu, nabízí vyšší kompresní poměr pro ztrátovou kompresi nebo také zavádí tzv. oblasti zájmu (oblasti, kde jsou data komprimována prioritně vůči ostatním méně důležitým částem).

Cílem této práce je prozkoumat různá nastavení formátu JPEG 2000 pro různé datové sady a vytvořit tak pro ně doporučení vhodných nastavení. Důležitou roli také hraje výběr vhodných datasetů, které budou reprezentovat možnosti použití formátu JPEG 2000. Formát jednotlivých obrazů v jednotlivých datasetech by měl být ideálně nekomprimovaný (např. TIFF). V každém datasetu by mělo být větší množství vzorků (cca 100), aby se analýza mohla považovat za směrodatnou. Framework pro analýzu nastavení by měl být variabilní a snadno se přizpůsobovat změnám v nastavování jednotlivých parametrů.

V první kapitole je shrnut současný přístup a nastíněn přínos formátu JPEG 2000 na poli komprese obrazu a jsou zde naznačeny principy formátu JPEG. Ne všechny definice a principy jsou v tomto textu uvedeny, dodatečné informace ke kompresi dat lze nalézt v odkazované literatuře [8], [13],[4]. Druhá kapitola je věnována formátu JPEG 2000, jeho podrobnému popisu a srovnání s formátem JPEG. Třetí kapitola popisuje implementaci frameworku pro analýzu jednotlivých nastavení formátu JPEG 2000, zároveň popisuje práci s vybranými knihovnamy, které tento formát plně podporují. V kapitolách čtyři a pět je zpracována analýza nastavení pro odlišné typy dat, za použití různých knihoven implementujících kodek JPEG 2000. Kapitola čtyři se zabývá bezztrátovou kompresí vhodnou zejména k archivaci co nejmenšího množství dat, při vysoké kvalitě (např. skenované knih).

Kapitola pět se pak zabývá ztrátovou kompresí, která je vhodná k přenosu po síti nebo archivaci dat, při kterých nevadí větší míra komprese (např. fotografie).



## Kapitola 2

# Kompresie obrazu

Kompresie obrazu je typ komprese dat aplikovaných na digitální obraz, za účelem redukce místa nebo přenosu. Algoritmy většinou čerpají výhody vizuálního vnímání, například té, že lidské oko nejvíce vnímá změnu jasu, dále pak statických vlastností obrazu k poskytnutí kvalitnějších výsledků v porovnání s generickými kompresními metodami [3].

### 2.1 Barevné modely

#### RGB

RGB je zkratka z anglických slov *red*, *green* a *blue* – česky červená, zelená a modrá. Tyto barevné složky jsou u zmíněného modelu zastoupeny rovnocenně. Jedná se o aditivní barevný model - míchání na principu barevných světelných kuželů, které mícháním získávají vyšší intenzitu než barvy míchané. Kombinací tří základních barev vznikne barva bílá. V počítačové grafice se lze nejčastěji setkat s modelem RGB24, který používá 24 bitů na jeden pixel, tedy 8 bitů na jednu barevnou složku.

#### Y'CbCr

Kvůli časté změně hodnot R, G, B složek mezi sousedními pixely se barevný model RGB ke kompresi obrazu nehodí.  $Y'CbCr$  model je založený na rozdělení složek na jasovou složku a barvosné složky - konkrétně modrou  $C_b$  (*blue*) a červenou  $C_r$  (*red*). Toto rozdělení je vhodné zejména proto, že lidské oko je více citlivé na jas a je možné se u barvosných složek dopustit při kompresi větší nepřesnosti než u jasu. Jelikož se jedná o formu kódování  $R'G'B'$  informace, je možné tento model převést na  $R'G'B'$  model.

Pro převod z 8bitového  $R'G'B'$  modelu na 8bitový  $Y'CbCr$  se využívá těchto rovnic 2.1, 2.2, 2.3, uvedených ve standardu JFIF:

$$Y = 0,299R + 0,587G + 0,114B \quad (2.1)$$

$$C_b = -0,1687 - 0,3313G + 0,5B + 128 \quad (2.2)$$

$$C_r = 0,5R - 0,4187G - 0,0813B + 128 \quad (2.3)$$

Převod zpět pak podle 2.4, 2.5, 2.6:

$$R = Y + 1,402(C_r - 128) \quad (2.4)$$

$$G = Y - 0,34414(C_b - 128) - 0,71414(C_r - 128) \quad (2.5)$$

$$B = Y + 1,772(C_b - 128) \quad (2.6)$$

Tento převod se nazývá ICT (Irreversible Color Transform). Je implementován v desetinných číslech a dochází tak k zaokrouhlovacím chybám. Pro bezztrátovou kompresi se používá převod RCT (Reversible Color Transform), ke kterému jsou definovány následující rovnice 2.7, 2.8, 2.9:

$$Y_r = \lfloor \frac{R + 2G + B}{4} \rfloor \quad (2.7)$$

$$C_b = B - G \quad (2.8)$$

$$C_r = R - G \quad (2.9)$$

Zpět pak podle 2.10, 2.11, 2.12:

$$G = Y - \lfloor \frac{1}{4}(C_b + C_r) \rfloor \quad (2.10)$$

$$R = C_r + G \quad (2.11)$$

$$B = C_b + G \quad (2.12)$$

## 2.2 Současný přístup ke kompresi obrazu

V současné době se nejčastěji pro ztrátovou kompresi využívá formát JPEG. Tento standard je z roku 1992 a mylně se označuje zkratkou JPEG, což je název výboru (Joint Photographic Experts Group), který stojí za standardem ISO/IEC 10918-1 nebo také doporučením CCITT/ITU-T T.81. Skutečným názvem formátu je JFIF, což znamená JPEG File Interchange Format. Přesto, že se primárně používá pro ztrátovou kompresi obrazu, definuje také režim pro bezztrátovou kompresi, která se ale běžně nepoužívá.

Standard JPEG definuje 4 kompresní postupy. Základní postup umožňuje komprimovat 8bitová vstupní data, pracuje s Huffmanovým kódováním. Data jsou komprimována pouze sekvenčně. Rozšířený postup umožňuje použít progresivní zpracování. Dalším postupem je postup bezztrátový, který se jak bylo řečeno výše, příliš nepoužívá. Posledním, taktéž nepoužívaným kompresním postupem, je postup hierarchický.

## Zkrácený popis funkcionality JPEG kodeku

Kodek JPEG využívá barevný model  $YC_bC_r$ , jako první krok tedy dojde k převodu pomocí rovnic 2.1, 2.2, 2.3.

Dále se barvonosné složky  $C_b$  a  $C_r$  podvzorkují, typicky na polovinu. Jednotlivé složky jsou následně zpracovávány odděleně. Jelikož by zpracování obrázku jako celku vyžadovalo vysoký výpočetní výkon, je každá složka rozdělena na bloky o velikosti 8x8 pixelů (vzorků). Rozdělení na bloky může u obrázku s vysokým stupněm komprese vést ke vzniku tzv. blokového efektu.

Pro každý blok se následně spočítá dvourozměrná diskretní kosinová transformace (DCT), která na rozdíl od diskretní Fourierovy transformace (DFT), produkuje pouze reálné koeficienty. Výsledkem je matice o 64 koeficientech. Po tomto kroku je postup zcela invertibilní. Nad bloky se provede kvantování koeficientů, ke kterému je potřebná tzv. kvantizační matice. Kvantizační matice je stejná pro celý obraz. Hodnoty v matici jsou takové, aby po vydělení původní hodnoty hodnotou z matice a zaokrouhlení vyšlo co nejmenší číslo, nejlépe nula. Nejvyšší hodnoty jsou v matici koncentrovány zejména u vysokých frekvencí, kde je možné dopustit se větší ztráty dat. Při dekompresi je tento postup opačný, tj. koeficienty jsou těmito hodnotami násobeny. Pro jasovou složku a chromatické složky se používá odlišná tabulka, jelikož lidské oko je více citlivé na jas. Kvantizační tabulka udává míru komprese.

Výsledná data jsou linearizována metodou zig-zag, kdy jsou nejdříve procházeny koeficienty s nižšími frekvencemi, jelikož je u nich i po kvantizaci vyšší pravděpodobnost, že zůstanou nenulové. Následně se data komprimují bezztrátově pomocí Run-length Encoding (RLE), vyjma koeficientu DC (Direct Current), který je kódován diferencially proti koeficientu DC z předchozího bloku.

Posledním bodem je komprimace pomocí Huffmanova nebo aritmetického kódování. V praxi se většinou využívá Huffmanova kódování. Koeficienty DC a AC (Alternating Current) se kódují pomocí odlišných tabulek Huffmanových kódů. I když je sestavení tabulek záležitostí kodéru, často se ke snížení výpočetní náročnosti komprese používají předpočítané tabulky. Základní formát kodeku JPEG je JIF (JPEG Interchange Format) využívající uspořádání bajtů big endian. Pro uložení lze využít různých jiných formátů, například TIFF. Nejčastěji se však používá formát JFIF, který přidává hlavičku a náhledy k formátu JIF.

## 2.3 Přínos formátu JPEG 2000

Cílem JPEG 2000 bylo přinést doplňující funkce ke staršímu JPEGu. Byl vytvořen stejným výborem v roce 2000 a je standardizován mezinárodním standardem ISO/IEC 15444-1 nebo také doporučením ITU-T T.800. Přináší řadu vylepšení:

- Podpora progresivního dekódování, které umožňuje pozorovateli vidět nižší verzi kvality obrazu, mezitím co se celý soubor přenáší. Kvalita obrazu se progresivně zlepšuje s přibývajícím daty.
- JPEG 2000 podporuje během jednoho datového přenosu ztrátovou i bezztrátovou kompresi.
- Nabízí zachování transparentnosti obrazu. Uplatnění této vlastnosti je především ve webových službách a tisku.

- Zahrnuje flexibilní formát souboru, který umožňuje zachování informací o použitém barevném prostoru, metadat a interaktivity v síťových aplikacích.
- Nabízí vyšší kompresní poměr pro ztrátovou kompresi.
- Podobně jako TIFF formát JPEG 2000:
  - Umožňuje členění neomezeného množství soukromých nebo speciálních informací do metadat v souboru.
  - Je navržen tak, aby byl snadno rozšiřitelný a vyvíjel se elegantně s nově přicházejícími potřebami.
  - Obsahuje řadu kompresních schémat a umožňuje tak vývojářům nasazení kompresního systému upravit přesně podle potřeby pro konkrétní situaci.
- Nabízí rozsáhlejší práci s obrazem (větší než 64 tisíc na 64 tisíc pixelů) bez použití dlaždic (angl. Tiling).
- Je schopný vypořádat se jak s přírodními, tak s počítačem generovanými obrazy.
- Nabízí jedno-dekompresní architekturu
- Poskytuje efektivnější kódování (nižší bitový tok), při zvýšení kvality. Efektivně až pod 0,25 bitů na pixel, pro vysoké rozlišení obrazů ve stupních šedi.
- Díky škálovatelnosti bitového toku mohou být data enkódována jen v takové míře, která je potřeba pro uspokojení požadavku, což může být například různá velikost obrazu.
- Oproti formátu formátu TIFF podporuje JPEG 2000 bezeztrátovou kompresi, což znamená že obraz může být komprimován tak, aby zabíral co nejmenší místo v paměti, ale zároveň neztratil nic na kvalitě.
- Oblast zájmu (regions of interest, ROI), která je komprimována prioritně vůči ostatním méně důležitým částem, např. pozadí. Při dekompresi tedy pozorovatel vidí oblast zájmu dříve než pozadí, což může být užitečné zejména při načítání velkých obrazů po síti.

## Kapitola 3

# JPEG 2000

JPEG 2000 je enkódovací systém pro kódování obrazu skládající se z několika částí. Vývoj byl zahájen v roce 1997 a první část publikována v roce 2000. Proces standardizace byl řízen skupinou JTC (Joint Technical Committee). JPEG 2000 používá kompresní techniky založené na diskrétní vlnkové transformaci a umožňuje uživatelům implementovat bezztrátovou kompresi. Cílem JPEG 2000 bylo vytvoření obrazového kódovacího systému pro různé typy statických obrazů s rozličnými charakteristikami. Povoluje také různé obrazové modely jako je přenos v reálném čase, archiv knihovny obrazů, limitovaný buffer a architekturu klient/server v rámci jednotného systému. Je používán ke kompresi obrazových dat, které obvykle pochází z digitálních fotoaparátů, skenerů, zdravotnických a satelitních systémů a programů pro úpravu fotografií.

### 3.1 Popis funkcionality kodeku JPEG 2000

Ke ztrátové kompresi využívá formát JPEG 2000 stejně jako JPEG barevný model  $YC_bC_r$ . K převodu lze využít rovnic uvedených výše (2.1, 2.2, 2.3). Pro bezztrátovou kompresi pak slouží transformační rovnice 2.7, 2.8, 2.9.

Pro snížení výpočetní náročnosti je stejně jako u JPEG možné rozdělit obraz na části, tentokrát se však nebudeme o blocích ale o tzv. dlaždicích (*tiles*). Velikost dlaždic je omezena na mocniny dvou a obvyklé rozměry se pohybují od 64 výše. Obdobně jako bloky i dlaždice jsou rozmístěny na pravidelné mřížce a nepřekrývají se. Rozdělením obrazu na dlaždice je snížena redundance na okrajích, díky kterému dojde k omezení výpočetních, především paměťových, nároků zařízení kódující obraz. Dlaždice jsou následovně zpracovávány nezávisle. Pro každou dlaždici je vypočtena diskrétní vlnková transformace (3.1.1) s rozkladem na podpásma. Používají se dva druhy filtrů – vlnek – podle toho, zda se jedná o ztrátovou nebo bezztrátovou kompresi. Používají se *Cohen-Daubechies-Feauveau* (CDF) vlnky ve dvou variantách. Pro bezztrátovou kompresi je určena vlnka CDF 5/3 (lze zapsat pomocí celých čísel) a pro ztrátovou kompresi vlnka CDF 9/7 (reprezentována v reálných číslech). Následně je na koeficientech DWT provedena kvantizace (3.1.2). Dojde k rozdělení podpásma na čtvercové nepřekrývající se části a určité části podpásma v dané úrovni a lokalitě jsou spojena do paketu. Po této fázi nastává fáze dělení podpásem na bloky, které jsou kódovány nezávisle. Proces kódování se provádí metodou EBCOT (3.1.3). Výstupem jsou dílčí toky, kde každý tok odpovídá určité bitové rovině. Spojením těchto dílčích toků pro určité bloky a následným spojením bloků ze všech tří podpásem vznikne paket. Paket je doplněn hlavičkou, která obsahuje informace potřebné pro dekódování. Skupina paketů na

jedné úrovni je označována jako vrstva. Jedna vrstva pak představuje kvalitativní zlepšení celého obrazu. Pakety jsou na sobě nezávislé, díky čemuž je možné pracovat s celým tokem podle potřeb konkrétního použití [14].

### 3.1.1 Diskrétní vlnková transformace

Diskrétní vlnková transformace (anglicky discrete wavelet transform, zkráceně DWT) je odvozená z vlnkové transformace pro diskrétní vlnky. Vlnka  $\psi$  je funkcí s konečnou energií z  $\psi \in L^2(\mathbb{R})$ , platí tedy 3.1 [14].

$$\int |\psi(t)|^2 dt < +\infty \quad (3.1)$$

Vlnková transformace je definována jako konvoluce mateřské vlnky a vstupního signálu. Je definována vztahem 3.2:

$$y[n] = \sum_{k=-\infty}^{\infty} x[n]h[n-k] \quad (3.2)$$

Provedení DWT signálu  $x$  se provádí pomocí dvou komplementárních filtrů. Filtry se označují jako dolní propust  $h_0$ , která vrací přibližné koeficienty signálu a horní propust  $h_1$ , která vrací podrobné koeficienty signálu. Tyto dva filtry spolu vzájemně souvisejí a jsou popsány následujícími rovnicemi 3.4, 3.3:

$$y_L[n] = \sum_{k=-\infty}^{\infty} x[k]h_0[2n-k] \quad (3.3)$$

$$y_H[n] = \sum_{k=-\infty}^{\infty} x[k]h_1[2n-k] \quad (3.4)$$

Pro ztrátovou kompresi se nejčastěji používá vlnka CDF 9/7 a pro bezztrátovou vlnka CDF 5/3 (využívá celočíselné aritmetiky). Výstupy filtrů jsou podvzorkovány dvěma. Tato filtrace je prováděna opakovaně pro další zvýšení frekvenčního rozlišení a tím se vytváří banka filtrů. Pro vstupní signál s počtem vzorků  $2^n$  je počet opakování až  $n$ . Vícezměrná transformace může být vypočítána jako série jednorozměrných transformací postupně v každém rozměru nebo může být pro maximalizaci efektu dekorelace a koncentrace signálu využita dvojrozměrná transformace. Vstupní 2d signál je tak dělen do čtyř podpásem a jsou proto zavedeny čtyři filtry (3.5), které jsou kombinací dvou původní filtrů horní a dolní propusti.

$$\begin{aligned} \phi(n_1, n_2) &= \phi(n_1)\phi(n_2) \\ \psi^H(n_1, n_2) &= \psi(n_1)\phi(n_2) \\ \psi^V(n_1, n_2) &= \phi(n_1)\psi(n_2) \\ \psi^D(n_1, n_2) &= \psi(n_1)\psi(n_2) \end{aligned} \quad (3.5)$$

Každý filtr reprezentuje jedno ze čtyř podpásem: HH - kombinace dvou horních propustí, HL a LH - kombinace horní a dolní propusti a naopak, LL kombinace dvou dolních propustí. Transformace je separabilní – umožňuje zpracování nejprve po řádcích a následně po sloupcích. Tento postup umožňuje provádět dělení až na úroveň jednotlivých bodů obrazu. Standardně se provádí pouze několik kroků dělení. Správně zvolené filtry umožňují perfektní rekonstrukci původního signálu postupným slučováním podpásem zpracovaných rekonstrukčními filtry, je potřeba provést nadzorkování. DWT není sama o sobě ztrátová.

### 3.1.2 Kvantizace

Výstupem DWT je matice koeficientů, které reprezentují výskyt odpovídající frekvence v původním bloku obrazu. Na určitých koeficientech se lze dopustit větší ztráty přesnosti než na jiných, protože je lidské oko méně citlivé na vysoké frekvence než na nízké. Proto se provádí proces kvantizace. Kvantizace se provádí s fixním krokem vždy pro celé podpásmo. Koeficienty DWT jsou vyděleny velikostí kroku a zaokrouhleny dolů na celé číslo. Různě zvolenou délkou kvantizačního kroku lze docílit například dosažení určité úrovně kvality nebo požadované velikosti. Kvantizace obvykle není hlavní ztrátový krok komprese a kvantizační krok je tak nastaven na hodnotu 1,0. Ztrátové komprese je docíleno pomocí přerušování toku bitů ve vybraný moment, jelikož jsou méně podstatné informace řazeny na konec. Odstranění dat je tak přesunuto až do fáze kódování.

### 3.1.3 EBCOT

EBCOT (embedded block coding with optimized truncation) enkóduje každou bitovou rovinu ve třech průchodech. Tři kódovací průchody v pořadí v jakém jsou vykonávány na jednotlivých bitových rovinách jsou *significant propagation* (volně přeloženo jako propagace významnosti), *magnitude refinement* (volně přeloženo jako upřesnění rozsahu) a *cleanup pass* (volně přeloženo jako fáze úklidu). Všechny tři typy kódovacích průchodů prochází vzorek kódového bloku ve fixním pořadí. Kódový blok je rozdělen do horizontální pruhů, každý o výšce čtyř vzorků. Pruhy jsou procházeny zhora dolů. V rámci pruhu jsou sloupce procházeny zleva doprava. V rámci sloupců jsou vzorky procházeny zhora dolů.

Každý koeficient v bitové rovině je kódovaný jen v jednom ze tří kódovacích průchodů. Pro každý koeficient v bloku je přiřazena binární stavová proměnná, která je nazývána významný stav koeficientu. Na začátku kódování je významný stav koeficientu inicializován na nulu (nevýznamný). Významný stav se mění z nuly na jedničku (významný), když je nalezen první nenulový bit. Kontextový vektor pro daný koeficient je binární vektor, skládající se z významného stavu jeho osmi bezprostředně sousedících koeficientů (osmiokolí). V každém průchodu jsou vytvořeny takovéto kontexty, které jdou poskytovány aritmetickému kodéru. Níže je jsou popsány jednotlivé kódovací průchody:

- a) **Significance propagation** Během fáze propagace významnosti je bit zakódovaný pokud je jeho umístění nevýznamné a zároveň je alespoň jeden koeficient z jeho osmiokolí významný.
- b) **Magnitude refinement** V průběhu této fáze jsou zakódovány všechny bity, které se v předchozí fázi staly významné. Tato fáze zahrnuje také bity od koeficientů, které již významné byly.
- c) **Clean-up** Fáze úklidu je finální fází, ve které jsou všechny bity, které nebyly zakódovány během předchozích fází, zakódovány (např. koeficienty, které jsou nevýznamné

a mají hodnotu kontextu nula během fáze propagace významnosti). První průchod v novém kódovém bloku je vždy fáze úklidu.

## 3.2 Srovnání JPEG a JPEG 2000

V porovnání s JPEG formátem JPEG 2000 poskytuje zdroje pro obě komprese, jak ztrátovou tak bezztrátovou. V důsledku toho má JPEG 2000 mírný náskok před JPEG, jelikož může zachovat více detailů než JPEG obraz komprimovaný do stejné velikosti. Podle případových studií zvládá JPEG 2000 zkomprimovat obraz od 20 procent do 200 procent lépe než JPEG [7].

Mezi největší nevýhody formátu JPEG 2000 patří nulová zpětná kompatibilita, tzn. není schopen přečíst JPEG soubory. To způsobuje značné problémy, jelikož většina obrazů dnešní doby je právě ve formátu JPEG. Navíc vlnková technologie na které je JPEG 2000 založen je komplexnější než kód používaný pro JPEG obrazy, což znamená že čtení a zápis JPEG 2000 obrazu trvá déle. JPEG 2000 také vyžaduje komplexnější software a více paměti pro jeho běh. Z tohoto důvodu práce s JPEG 2000 obecně trvá déle než práce s JPEG obrazy. Pro všechny tyto důvody nebyl JPEG 2000 stále schopen převzít žezlo jako preferovaný a nejvíce používaný obrazový formát navzdory tomu, že je novější a technologicky vyspělejší.



## Kapitola 4

# Implementace

Framework pro analýzu nastavení formátu JPEG 2000 je implementován v jazyce Python3. Byly vybrány dvě knihovny implementující formát JPEG 2000. Jedna open-source OpenJPEG a druhá komerční Kakadu. Tyto knihovny byly zvoleny, jelikož patří mezi nejrozšířenější.

V následující kapitole je popsána složitost práce s jednotlivými knihovnami, popis implementace samotného frameworku a následně analýza dat – konkrétně použité srovnávací a zobrazovací metody.

### 4.1 Práce s knihovnou OpenJPEG

OpenJPEG je otevřená knihovna implementující kodek JPEG 2000, napsaná v jazyce C. Byla vyvinuta za účelem propagace používání standardu JPEG 2000. Od května roku 2015 je oficiálně uznána ISO/IEC a ITU-T jako referenční software pro JPEG 2000 [6].

Knihovna má velice stručnou dokumentaci, která je v mnoha případech nedostačující. Knihovna také při určitém nastavení skončí s chybovým kódem.

Konkrétní nastavení pro která knihovna padá jsou následující:

- Bezeztrátová i ztrátová komprese
  - dlaždice: 512; počet rozlišení: 9, 10; mód: Bypass, Restart, Bypass+Restart, Reset+Restart, Bypass+Reset+Restart
  - dlaždice: 1024; počet rozlišení: 10; mód: Bypass, Restart, Bypass+Restart, Reset+Restart, Bypass+Reset+Restart

Mimo výše zmíněné nebyly s knihovnou od verze 2.3.0, ve které byl opraven špatně fungující bypass mód, problémy.

### 4.2 Práce s knihovnou Kakadu

Kakadu je kompletní implementací první části JPEG 2000 standardu, např. ISO/IEC 15444-1 plus velké části části dva a tři. Kakadu framework lze využít jak pro komerční tak pro nekomerční účely [10]. Kodek je napsaný v jazyce C++. Autorem tohoto softwaru je David Taubman.

Software Kakadu poskytuje rozsáhlou a dobře propracovanou dokumentaci, včetně konkrétních případů použití, jsou zde dopodrobna rozebrána jednotlivá nastavení a vysvětleny

některé principy [9], [12], [11]. K tomuto kodeku funguje také živé diskuzní fórum, kde se vám dostane velice rychlé pomoci při potížích. Pokud vlastníte placenou verzi, můžete si problém sami opravit přímo ve zdrojových souborech. Jelikož se jedná o komerční implementaci kodeku, je bezplatný přístup pouze k již zkompilevanému binárnímu souboru, což vede k nemožnosti debugovat některé chyby, které mohou vzniknout během práce s ním. Některé z nich jsou popsány v následujícím textu.

### 4.3 Popis frameworku

Framework je implementovaný v jazyce Python3 a spouští se s jediným parametrem, a to `-i <vstupní složka>`, která obsahuje podsložku s datasetem, který má být zpracován. Skript si v dané složce vytvoří složky *compressed* a *decompressed* pro dočasné ukládání komprimovaných a dekomprimovaných fotek, které po sobě po úspěšně dokončeném běhu smaže. Skript hledá fotky ve formátu TIF, které poté podle zadaných parametru, které se dají nastavit v horní části kódu (podrobněji popsáno v manuálu, viz. Příloha A), zkomprimuje, porovná s originálem, dekomprimuje a opět porovná s originálem. Po zpracování celého datasetu s určitým nastavením se průměrné vypočtené hodnoty zapíše do výstupního souboru, který je ve formátu csv. Skript po sobě nakonec vše smaže a zanechá pouze pomocné textové a finální csv soubory.

Jednotlivé knihovny jsou volány pomocí systémového volání *call* s doplněním příslušných parametru, jednotlivé obrázky jsou zpracovány nejprve knihovnou Kakadu a následně pak knihovnou OpenJPEG.

### 4.4 Analýza dat

#### 4.4.1 Použité srovnávací metody

Pro kvalitativní srovnání komprimovaných a následně i dekomprimovaných obrázků s originálem je použita objektivní srovnávací metoda PSNR, nebo-li peak signal-to-noise ratio, česky špičkový poměr signálu k šumu. Tato metoda je založena na porovnávání dvou obrazů – původního a pozměněného – což je označováno jako *full reference* (FR).

#### MSE

MSE, nebo-li Mean Squared Error, česky střední kvadratická chyba je základem pro výpočet PSNR a zároveň se jedná o nejjednodušší a nejpoužívanější metodou pro objektivní hodnocení kvality obrazu. Pro dvourozměrný obrazový signál je definována takto [16]:

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (X_{i,j} - Y_{i,j})^2, \quad (4.1)$$

kde  $X_{i,j}$  a  $Y_{i,j}$  představují hodnoty pixelů dvou různých obrázků v rámci jednoho kanálu barvové informace (např. jasová složka) o rozměrech  $M \times N$ . Pokud X bude bráno za původní (nezkreslený) obrázek a Y za jeho zkreslenou verzi, u níž má být stanovena vizuální kvalita, potom lze MSE považovat za měřítko kvality obrazového signálu. Výraz  $X_{i,j} - Y_{i,j}$  pak udává chybový signál, který vznikne mezi hodnotami původního a zkresleného obrazového signálu. Pro barevný obrázek (se složkami RGB) je nutno výpočet MSE podle (4.1) upravit přidáním sumy sčítající chyby jednotlivých barevných složek [16].

$$MSE = \frac{1}{3MN} \sum_{k=1}^3 \sum_{i=1}^M \sum_{j=1}^N (X_{i,j} - Y_{i,j})^2, \quad (4.2)$$

Výhodou metody MSE je především jednoduchost. Její výpočet je nenáročný a velice rychlý. Není třeba uvažovat žádné závislosti mezi jednotlivými pixely, jelikož kvadratická chyba je počítána pro každý pixel zvlášť, nezávisle na ostatních hodnotách pixelů. Již od svého vzniku bylo MSE využíváno k porovnávání a optimalizaci v řadě různých aplikací pro zpracování signálů, zahrnující návrh filtrů, kompresi, třídění a restauraci signálů. Efektivita algoritmů byla hodnocena na základě hodnoty MSE, tudíž i pro nově vzniklé algoritmy představovalo MSE nejběžnější společný způsob srovnávání. Tak došlo k značnému rozšíření této metriky. Je zřejmé, že MSE má mnoho výhodných vlastností pro užití v různých aplikacích. Mohlo by se tedy zdát, že je ideálním prostředkem pro objektivní hodnocení kvality obrazu. Toto tvrzení v řadě případů rozhodně neplatí [16].

## PSNR

PSNR je nejčastěji používaný indikátor pro hodnocení kvality obrazu. Metoda vychází z výpočtu hodnoty MSE a zavádí se z důvodu možnosti srovnání obrázků s rozdílnými dynamickými rozsahy. Udává poměr mezi maximální možnou energií obrazového signálu a energií šumu v obraze. Rovnice má následující tvar [16]:

$$PSNR = 10 \log_{10} \frac{(2^n - 1)^2}{MSE} \quad (4.3)$$

kde  $n$  značí bitovou hloubku obrazu. PSNR je vyjadřováno v jednotkách decibel [dB]. Typická hodnota PSNR pro komprimované obrázky je mezi 30 a 40 dB, čím větší tím lepší. Korelace s výsledky subjektivních testů jsou okolo 80%. To je porovnatelné s některými komplikovanými algoritmy. Proto může být PSNR použito jako indikátor pro kvalitu obrazu.

## Měření časové náročnosti

Pro měření časové náročnosti jak komprese, tak dekomprese, je použita systémová funkce `/usr/bin/time`, konkrétně uživatelský čas získaný použitím argumentu `%U`.

### 4.4.2 Vykreslení grafů

Pro vykreslení grafů je použit program GNUplot, který vykresluje grafy na základě matematických funkcí, jsou tedy zcela škálovatelné. Konkrétně byl použit terminál `epslatex`.

## Kapitola 5

# Analýza nastavení JPEG 2000 pro bezeztrátovou kompresi

Bezeztrátová komprese se používá, v případě požadavku na zachování co nejvyšší možné kvality, při relativně malé velikosti dat. Většinou se využívá při kompresi skenovaných knih nebo map pro účely archivace v knihovnách nebo obdobných zařízeních pro uchovávání písemností (viz. například [15]).

OpenJPEGu byly při této kompresi nastaveny tyto parametry:

- -r 1 - počet vrstev kvality, u bezeztrátové komprese je vrstva vždy právě jedna
- -n 1-10 - počet úrovní rozlišení, nebo-li počet provedených DWT dekompozicí + 1. Tento parametr patří k jednomu ze zkoumaných a proto je uveden v rozsahu 1-10
- -t  $512 \times 512$ | $1096 \times 1096$ | $2048 \times 2048$ | $4096 \times 4096$  - velikost dlaždice, je taktéž jedním ze zkoumaných parametrů. Udává velikost dlaždic, po kterých se obraz zpracovává. Při větší velikosti vstupního obrazu, by bylo možné přidat ještě dlaždici  $8192 \times 8192$
- -b  $64 \times 64$  - velikost code-blocks. Tento parametr je pevně daný, jelikož se po několika experimentech ukázala tato velikost jako nejvhodnější.
- -p RPCL - pořadí paket v code streamu. Jelikož se jedná o všeobecně používaný profil pořadí, nebylo zde k jeho nastavení vyžadováno žádných experimentů
- -d 0,0 - umístění počátku souřadnic obrazu. Experimentováno s počátkem v  $\langle 0,0 \rangle$  a  $\langle 1,1 \rangle$ . Nejlépe vyšel počátek v  $\langle 0,0 \rangle$
- -M 0-7 - použití módů. Označení jednotlivých módů je číselné. BYPASS [1], RESET [2], RESTART [4]. Kombinaci módů lze docílit sečtením jejich číselných hodnot. Tedy použití všech znamená nastavit tento parametr na číslo 7.

U dekomprese nebyly nastaveny žádné speciální parametry. Pouze vstupní a výstupní soubor.

U knihovny Kakadu bylo pro kompresi obdobné nastavení, a to:

- Creversible = yes - explicitní volba ztrátové, či bezeztrátové komprese.
- Stiles =  $512 \times 512$ | $1096 \times 1096$ | $2048 \times 2048$ | $4096 \times 4096$  - velikost dlaždice.

- Clevels = 0-9 - počet úrovní rozlišení. Zde je na rozdíl od OpenJPEGu rozsah posunut o jedno směrem dolů, jelikož přímo odpovídá počtu DWT dekompozice.
- Clayers = 1 - počet vrstev kvality
- Cblk =  $64 \times 64$  - velikost code blocks
- Corder = RPCL - pořadí paket v code streamu
- Cmodes = 0, BYPASS, RESET, RESTART, BYPASS|RESET, BYPASS|RESTART, BYPASS|RESET|RESTART - použití módů, tentokrát jsou označeny slovně
- Sorigin = 0,0 - umístění počátku souřadnic obrazu, tzv. offsetu
- -num\_threads 4 - nastavení počtu vláken, ve kterých může běžet (Kakadu umí pracovat s vlákny).

U dekomprese opět nebyly nastaveny žádné speciální parametry. Pouze vstupní a výstupní soubor.

U bezztrátové komprese byl zkoumán čas komprese, počet bitů na pixel u komprimovaného obrazu a čas dekomprese v závislosti na počtu úrovní rozlišení.

Kvůli padajícímu OpenJPEGu při určitém nastavení, zmíněném v sekci 4.1 chybí v grafech některá data.

## 5.1 Dataset č.1 - 6 Mpx fotografie

Dataset 6 Mpx fotografie, je složen z fotografií o velikosti 6 Megapixelů. Obsah fotografií je různorodý (portréty, krajina, ...). Všechny fotografie v tomto datasetu jsou ve formátu RGB24 - tzn. 8 bitů na jeden barevný kanál.

### 5.1.1 Knihovna OpenJPEG

#### Počet bitů na pixel u komprimovaného obrazu

Původní fotografie měla 24 bitů na pixel, z grafu 5.1 je vidět, že se po kompresi počet bitů zmenšil na polovinu. Což je logické, jelikož při bezztrátové kompresi dochází k odstranění redundantních dat, právě z barevného profilu obrazu.

Mezi jednotlivými nastaveními nejsou markantní rozdíly, k nejlepším patří nastavení

- dlaždice  $512 \times 512$ , mód 7 (Bypass+restart+reset)
- dlaždice  $512 \times 512$ , mód 5 (Bypass+restart)
- dlaždice  $4096 \times 4096$ , mód 7 (Bypass+restart+reset)
- dlaždice  $1024 \times 1024$ , mód 7 (Bypass+restart+reset).

Mezi nejhorší potom

- dlaždice  $4096 \times 4096$ , mód 0 (žádný).

Z grafu je také vidět, že nejvhodnější počet úrovní rozlišení je 5, od kterého se výkyvy v hodnotách stabilizují. Je tedy zbytečné používat větších úrovní rozlišení, které by jen zbytečně využívali více výpočetního výkonu, při dosažení stejných výsledků.

## Čas komprese a dekomprese

Obecně lze z grafů 5.2 a 5.3 vyčíst, že čas potřebný pro kompresi a dekompresi obrazu z tohoto datsetu je velice krátký.

U komprese patří k nejrychlejším nastavení

- dlaždice  $1024 \times 1024$ , mód 1 (Bypass)
- dlaždice  $1024 \times 1024$ , mód 5 (Bypass+restart)
- dlaždice  $512 \times 512$ , mód 1 (Bypass)
- dlaždice  $512 \times 512$ , mód 3 (Bypass+reset)
- dlaždice  $512 \times 512$ , mód 5 (Bypass+restart).

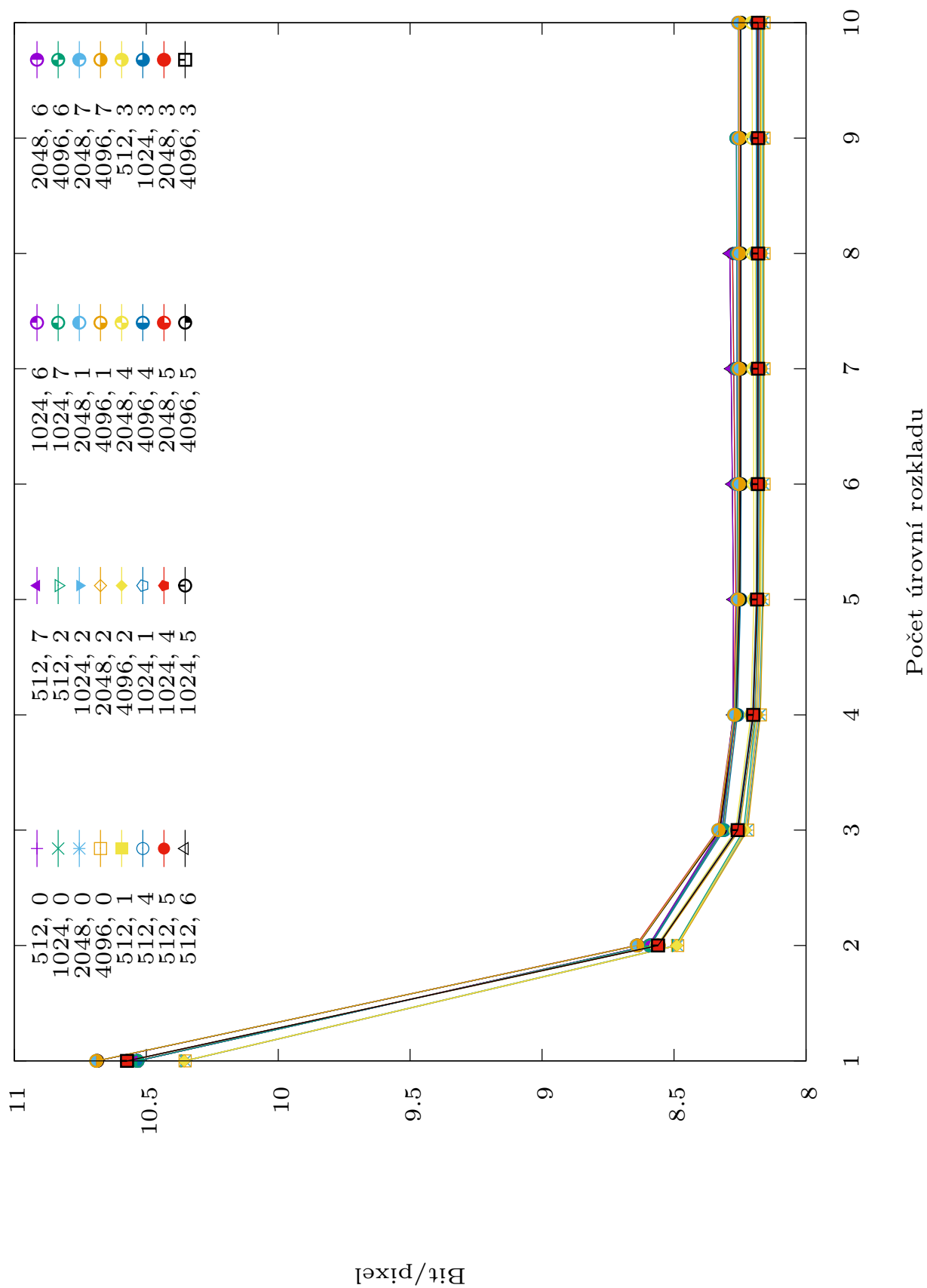
U dekomprese pak nastavení

- dlaždice  $512 \times 512$ , mód 1 (Bypass)
- dlaždice  $512 \times 512$ , mód 5 (Bypass+restart)
- dlaždice  $512 \times 512$ , mód 3 (Bypass+reset)
- dlaždice  $1024 \times 1024$ , mód 5 (Bypass+restart)
- dlaždice  $1024 \times 1024$ , mód 1 (Bypass).

K nejpomalejším nastavením v obou případech patří

- dlaždice  $4096 \times 4096$ , mód 0 (žádný).

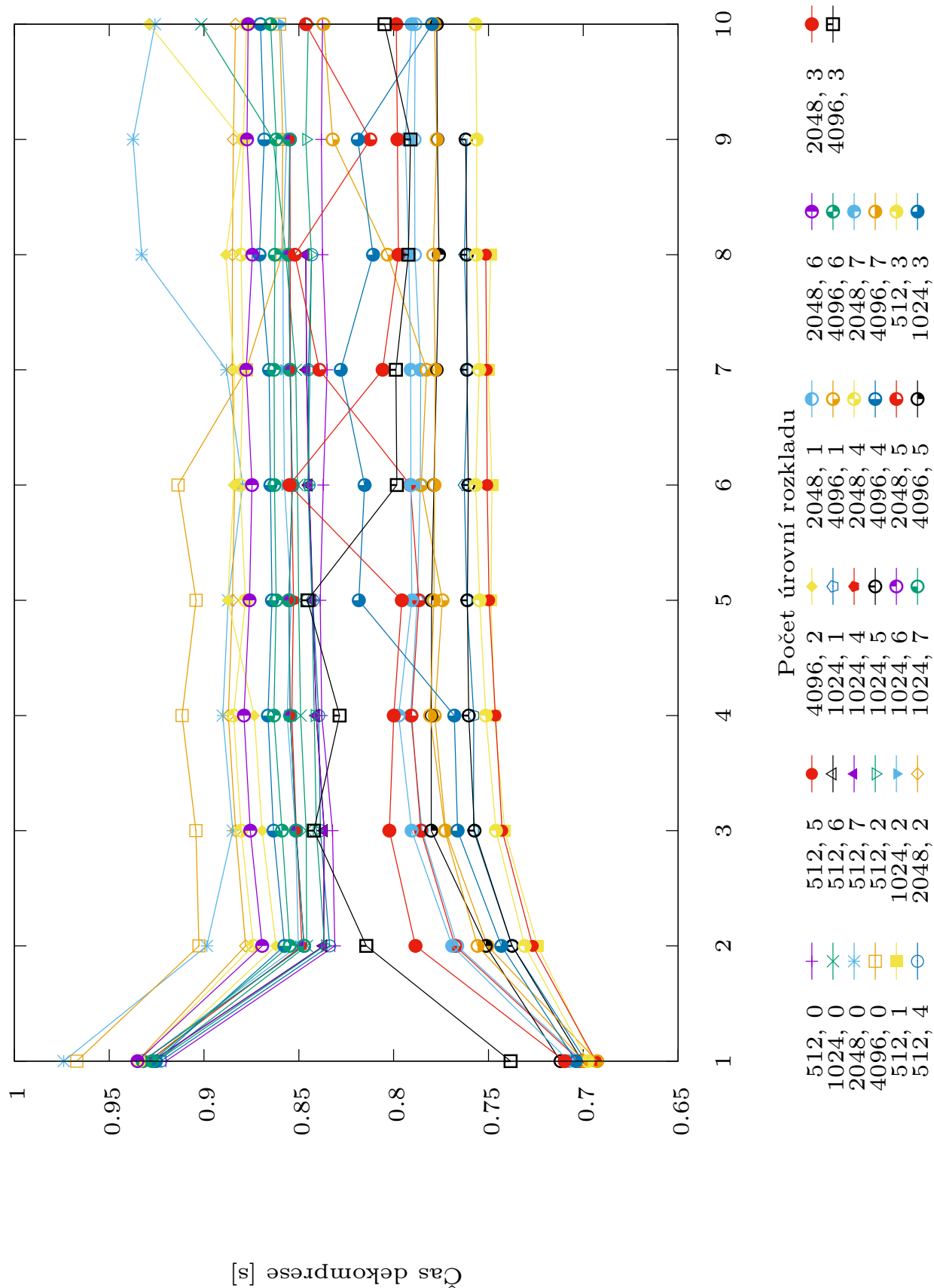
Potvrdilo se také nejlepší nastavení počtu úrovní rozlišení jako u předchozího grafu a to na hodnotu 5.



Obrázek 5.1: Open.JPEG - bezztrátová komprese, fotografie s rozlišením 6Mpx - průměrný počet bitů na pixel u komprimovaného obrazu







Obrázek 5.3: OpenJPEG - bezeztrátová komprese, fotografie s rozlišením 6Mpx - průměrný čas dekomprese jednoho obrazu

## 5.1.2 Knihovna Kakadu

### Počet bitů na pixel u komprimovaného obrazu

Graf 5.4 je téměř totožný s grafem 5.1 z předchozí sekce.

Mezi jednotlivými nastaveními nejsou opět markantní rozdíly a k nejlepším nastavení pak patří

- dlaždice  $4096 \times 4096$ , mód Restart+reset
- dlaždice  $512 \times 512$ , mód Bypass+reset
- dlaždice  $512 \times 512$ , mód Bypass+restart
- dlaždice  $4096 \times 4096$ , mód Bypass+restart+reset.

Mezi nejhorší potom

- dlaždice  $4096 \times 4096$ , mód 0 (žádný).

Z grafu je také opět vidět, že nevhodnější počet úrovní rozlišení je 5.

### Čas komprese a dekomprese

V grafech 5.5 a 5.6 lze vidět, že čas pro kompresi a dekompresi obrazu je téměř zanedbatelný.

U komprese patří k nejrychlejším nastavení

- dlaždice  $4096 \times 4096$ , mód Bypass+restart+reset
- dlaždice  $512 \times 512$ , mód Bypass+restart+reset
- dlaždice  $2048 \times 2048$ , mód Bypass+restart+reset .

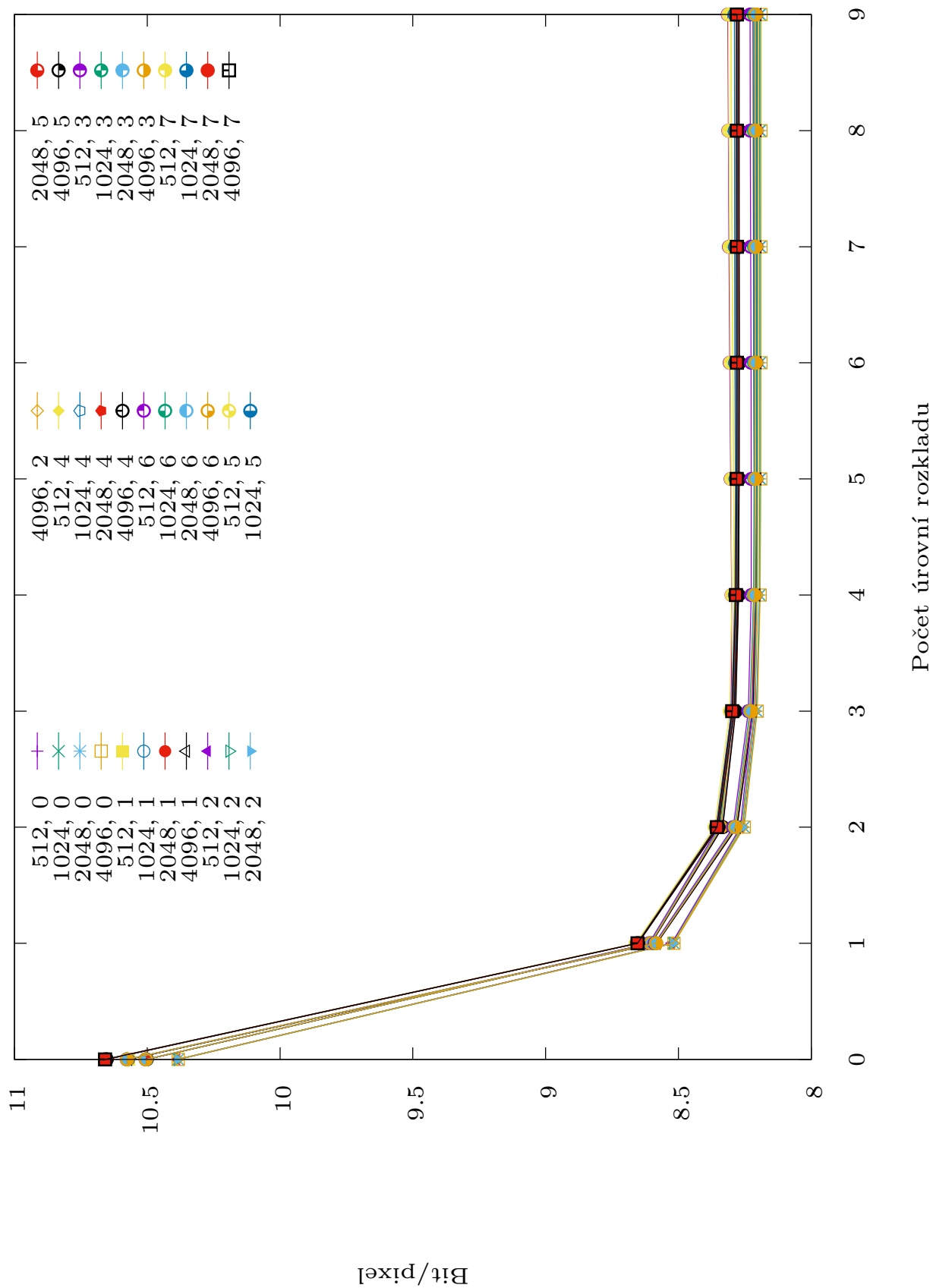
U dekomprese pak nastavení

- dlaždice  $4096 \times 4096$ , mód Bypass
- dlaždice  $1024 \times 1024$ , mód Bypass+restart
- dlaždice  $2048 \times 2048$ , mód Bypass+restart
- dlaždice  $2048 \times 2048$ , mód Bypass
- dlaždice  $2048 \times 2048$ , mód Bypass+restart+reset.

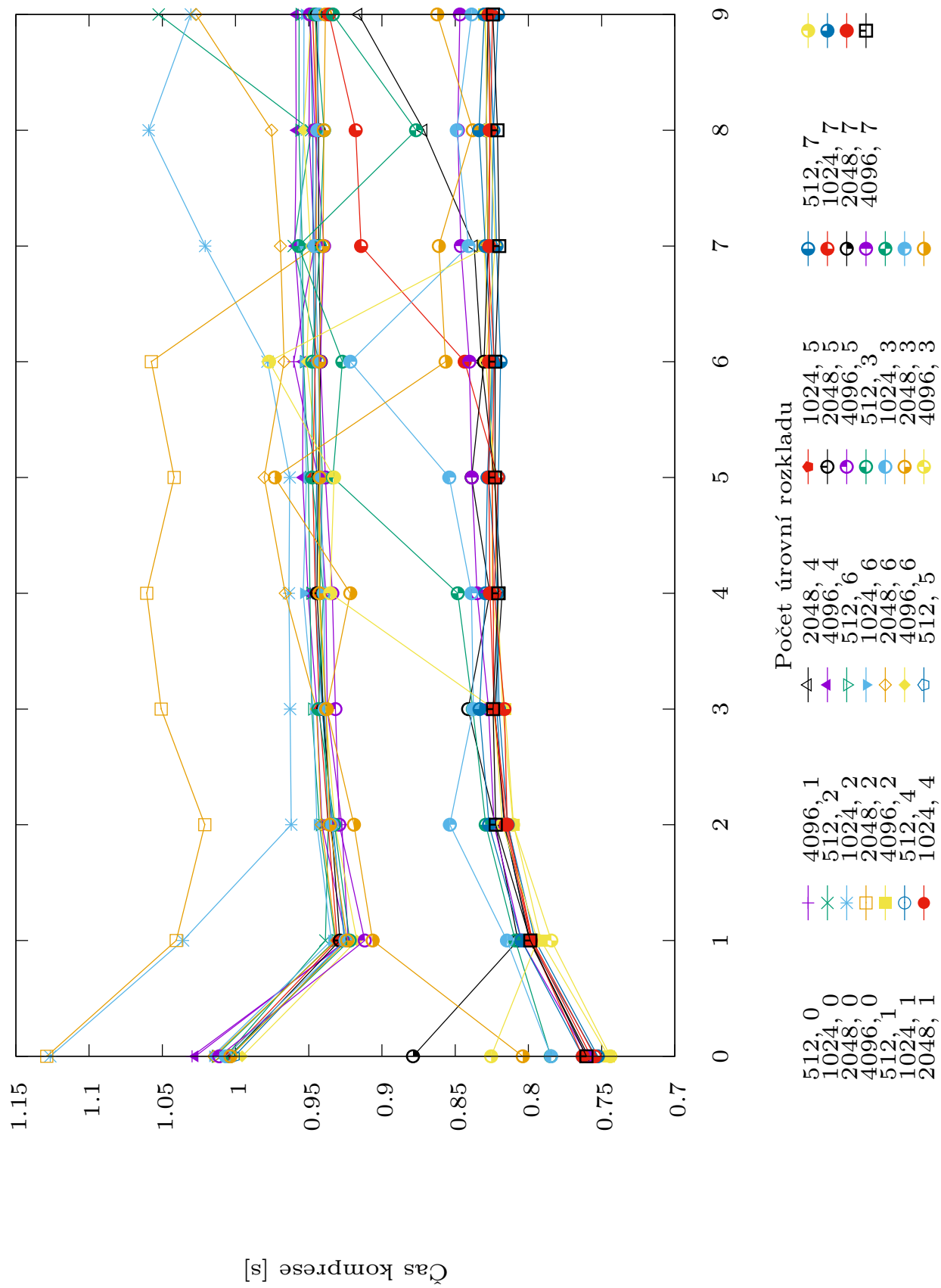
K nejpomalejším nastavením v obou případech patří

- dlaždice  $4096 \times 4096$ , mód 0 (žádný).

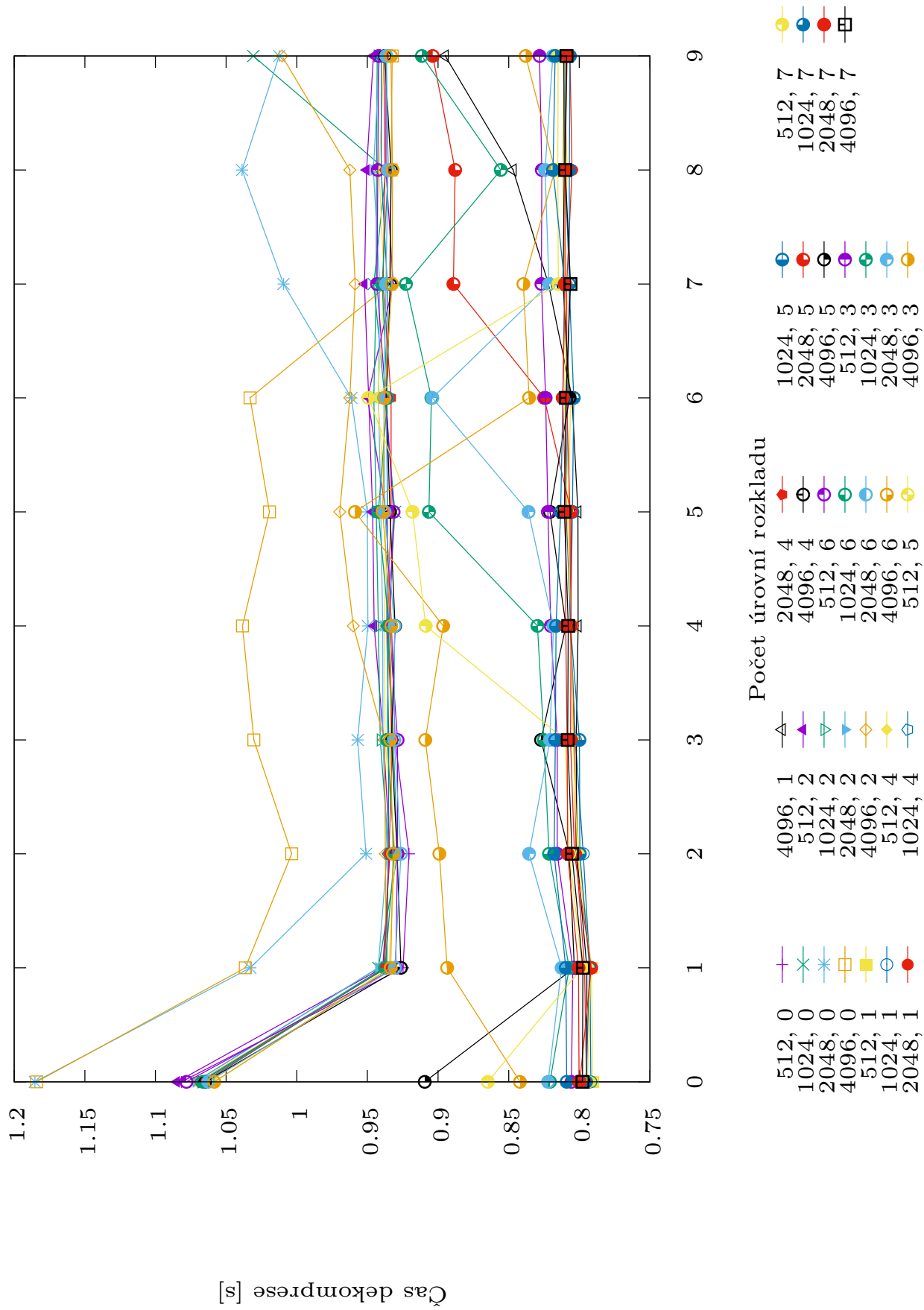
Potvrdilo se také nejlepší nastavení počtu úrovní rozlišení jako u předchozího grafu a to na hodnotu 5.



Obrázek 5.4: Kakadu - bezztrátová komprese, fotografie s rozlišením 6Mpx - průměrný počet bitů na pixel u komprimovaného obrazu



Obrázek 5.5: Kakadu - bezztrátová komprese, fotografie s rozlišením 6Mpx - průměrný čas komprese jednoho obrazu



Obrázek 5.6: Kakadu - bezztrátová komprese, fotografie s rozlišením 6Mpx - průměrný čas dekomprese jednoho obrázu

### 5.1.3 Shrnutí

Obecně jsou obě knihovny celkem vyrovnané. OpenJPEG je o trochu rychlejší při dekompresi, oproti tomu Kakadu je o trochu rychlejší při kompresi. V počtu bitů na pixel jsou si téměř rovni. Při použití knihovny OpenJPEG se jeví jako neoptimálnější nastavení dlaždice  $512 \times 512$ , mód Bypass+restart+reset. U knihovny Kakadu potom dlaždice  $2048 \times 2048$ , mód Bypass+restart+reset. Jako nejhorší nastavení vyšlo u obou knihoven dlaždice  $4096 \times 4096$ , mód 0 (žádný), což dokazuje, že používání módů je žádoucí a dokáže značně urychlit kompresi i dekompresi. Počet úrovní rozlišení vyšel jako optimální u obou knihoven 5.

## 5.2 Dataset č.2 - Skenované časopisy

Dataset skenované časopisy obsahuje skenované celobarevné stránky časopisů, rozlišení přibližně 33 Megapixelů. Celý dataset je opět ve formátu RGB24.

### 5.2.1 Knihovna OpenJPEG

#### Počet bitů na pixel u komprimovaného obrazu

Z grafu 5.7 je vidět, že se po kompresi počet bitů zmenšil o třetinu, což je oproti předchozímu datasetu výrazně méně.

Mezi jednotlivými nastaveními nejsou opět markantní rozdíly a k nejlepším nastavení pak patří

- dlaždice  $512 \times 512$ , mód 7 (Bypass+restart+reset)
- dlaždice  $512 \times 512$ , mód 4 (Restart)
- dlaždice  $1024 \times 1024$ , mód 7 (Bypass+restart+reset)
- dlaždice  $1024 \times 1024$ , mód 6 (Restart+reset).

Mezi nejhorší potom

- dlaždice  $4096 \times 4096$ , mód 0 (žádný).

Z grafu je také opět vidět, že nevhodnější počet úrovní rozlišení je 5.

#### Čas komprese a dekomprese

V grafech 5.8 a 5.9 lze vidět, že čas pro kompresi a dekompresi obrazu se oproti předchozímu datasetu výrazně zvýšil, což je vzhledem k velikosti obrazu očekávané. Zde si lze také všimnout již několika sekundových rozdílů, při použití různých nastavení.

U komprese patří k nejrychlejším nastavení

- dlaždice  $1024 \times 1024$ , mód 1 (Bypass)
- dlaždice  $1024 \times 1024$ , mód 3 (Bypass+restart)
- dlaždice  $512 \times 512$ , mód 5 (Bypass+restart)
- dlaždice  $1024 \times 1024$ , mód 5 (Bypass+reset).

K nejpomalejším nastavením u komprese patří

- dlaždice  $4096 \times 4096$ , mód 6 (Restart+reset)
- dlaždice  $4096 \times 4096$ , mód 4 (Restart).

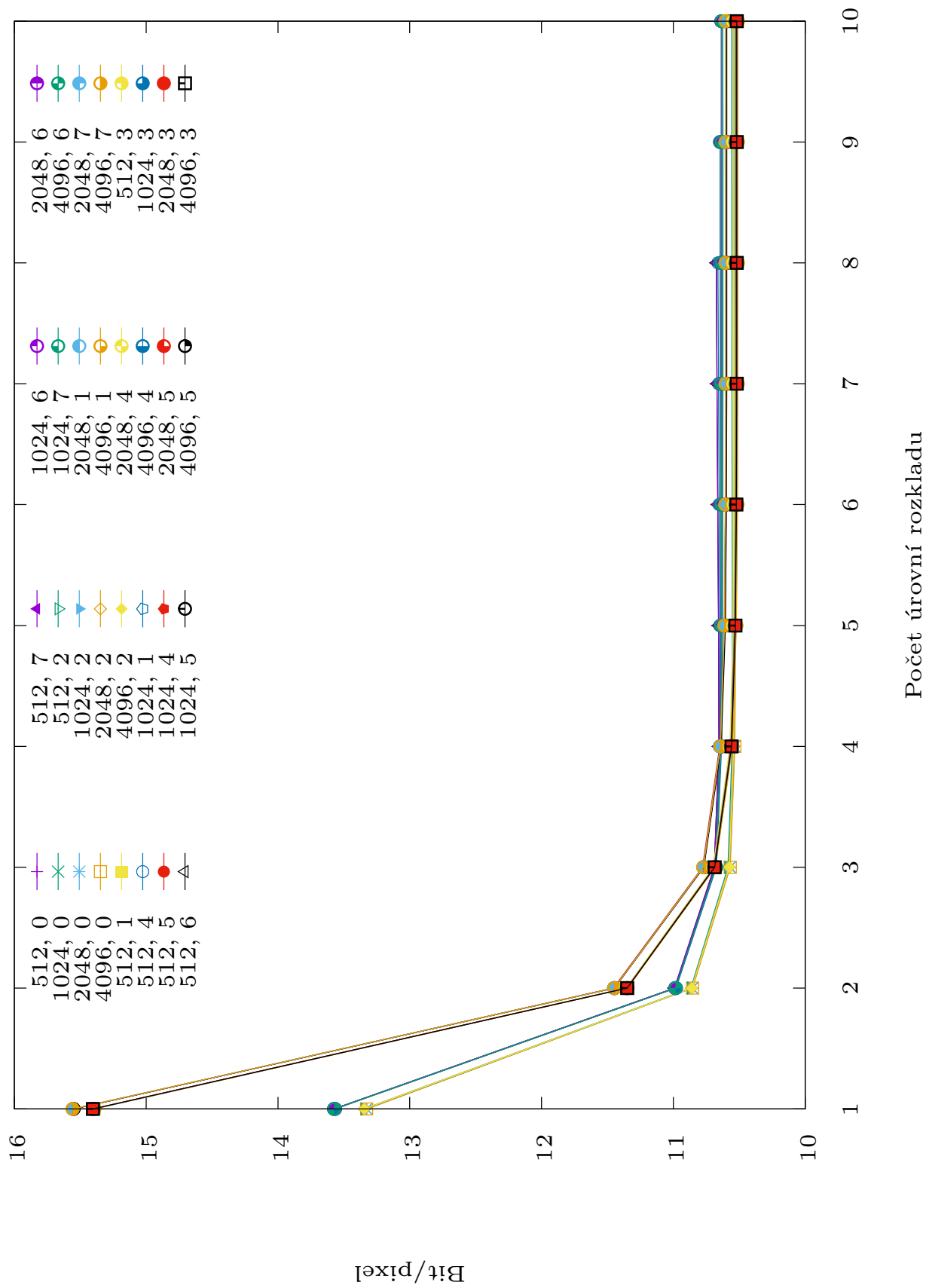
U dekomprese pak nastavení

- dlaždice  $512 \times 512$ , mód 3 (Bypass+reset)
- dlaždice  $512 \times 512$ , mód 5 (Bypass+restart).

K nejpomalejším nastavením u dekomprese patří

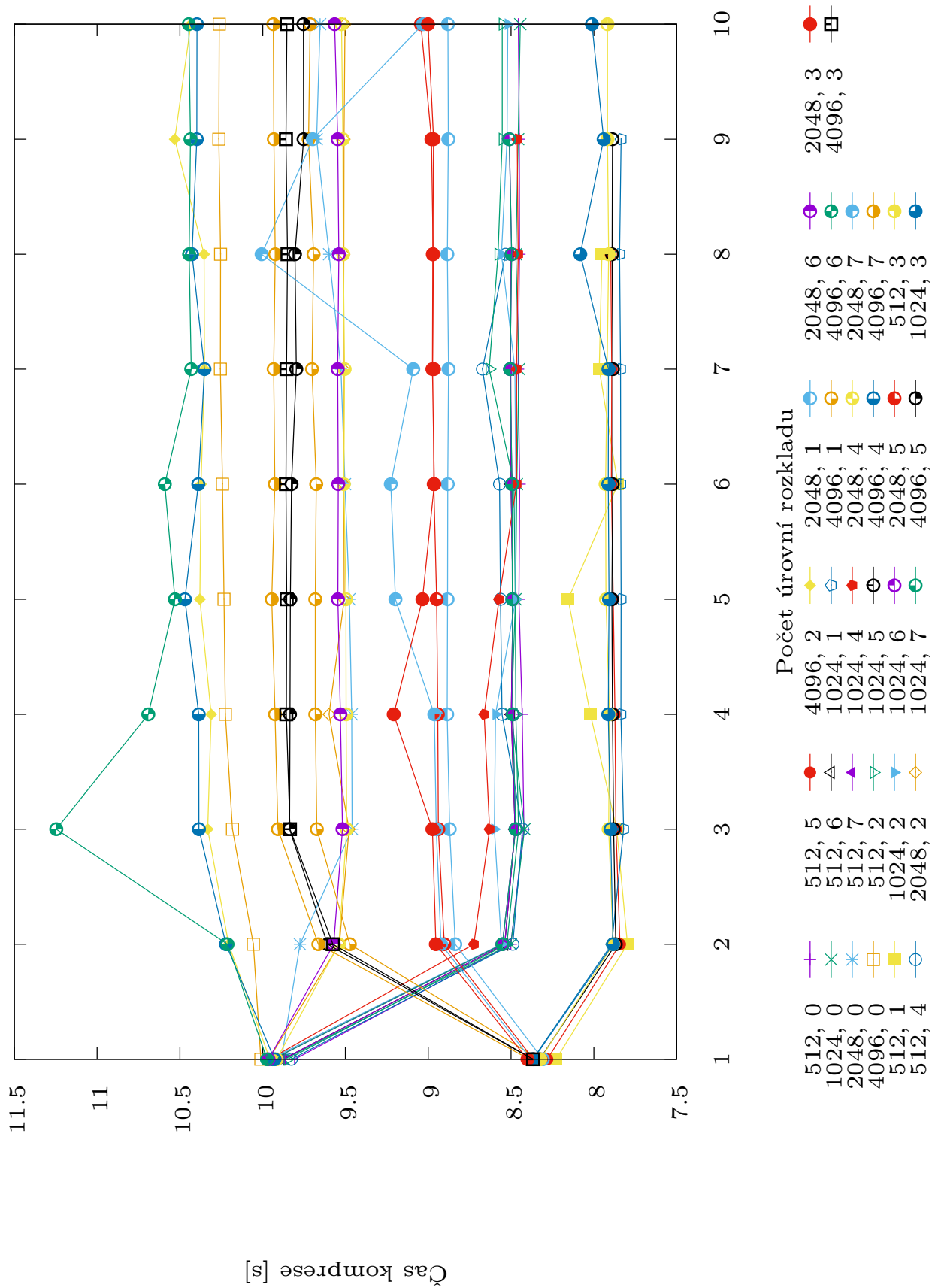
- dlaždice  $4096 \times 4096$ , mód 6 (Restart+reset)
- 
- dlaždice  $4096 \times 4096$ , mód 4 (Restart).

Potvrdilo se také nejlepší nastavení počtu úrovní rozlišení jako u předchozího grafu a to na hodnotu 5.

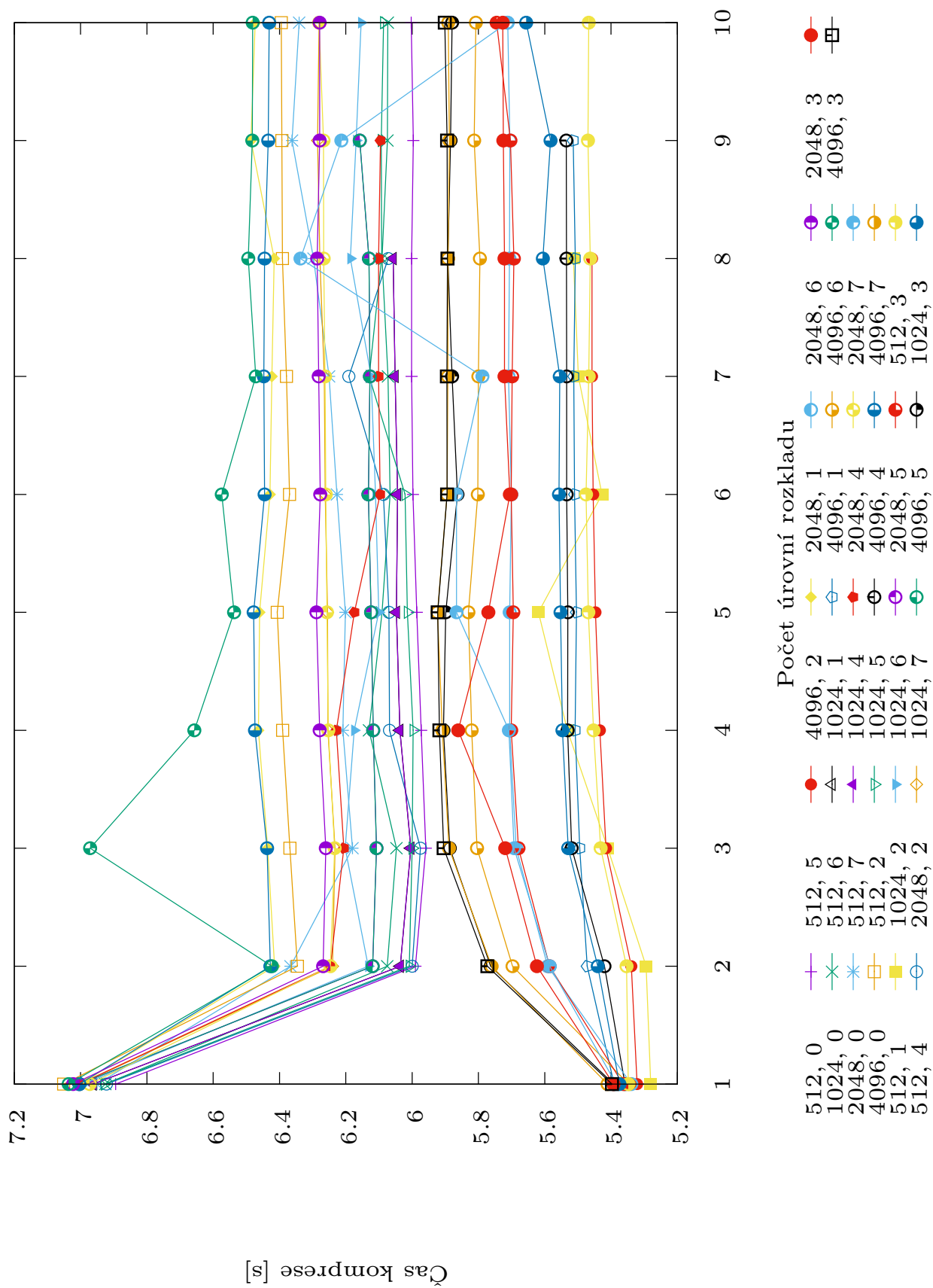


Obrázek 5.7: OpenJPEG - bezztrátová komprese, skenované časopisy - průměrný počet bitů na pixel u komprimovaného obrazu





Obrázek 5.8: OpenJPEG - bezztrátová komprese, skenované časopisy - průměrný čas komprese jednoho obrazu



Obrázek 5.9: Open.JPEG - bezetrátová komprese, skenované časopisy - průměrný čas dekomprese jednoho obrazu

## 5.2.2 Knihovna Kakadu

### Počet bitů na pixel u komprimovaného obrazu

Z grafu 5.10 je vidět, že se po kompresi počet bitů zmenšil téměř o polovinu.

Mezi jednotlivými nastaveními nejsou opět markantní rozdíly a k nejlepším nastavení pak patří

- dlaždice  $512 \times 512$ , mód Restart+reset
- dlaždice  $512 \times 512$ , mód Restart
- dlaždice  $512 \times 512$ , mód Bypass+restart
- dlaždice  $512 \times 512$ , mód Bypass+restart+reset.

Mezi nejhorší potom

- dlaždice  $4096 \times 4096$ , mód Bypass+reset.

Z grafu je také opět vidět, že nevhodnější počet úrovní rozlišení je 5.

### Čas komprese a dekomprese

V grafech 5.11 a 5.12 lze vidět, že doba kompresi obrazu je přibližně osmkrát delší, než tomu bylo u předchozího datasetu.

U komprese patří k nejrychlejším nastavení

- dlaždice  $4096 \times 4096$ , mód Bypass
- dlaždice  $2048 \times 2048$ , mód Bypass+restart+reset
- dlaždice  $1024 \times 1024$ , mód Bypass.

K nejpomalejším nastavením u komprese patří

- dlaždice  $512 \times 512$ , mód Restart
- dlaždice  $512 \times 512$ , mód Reset.

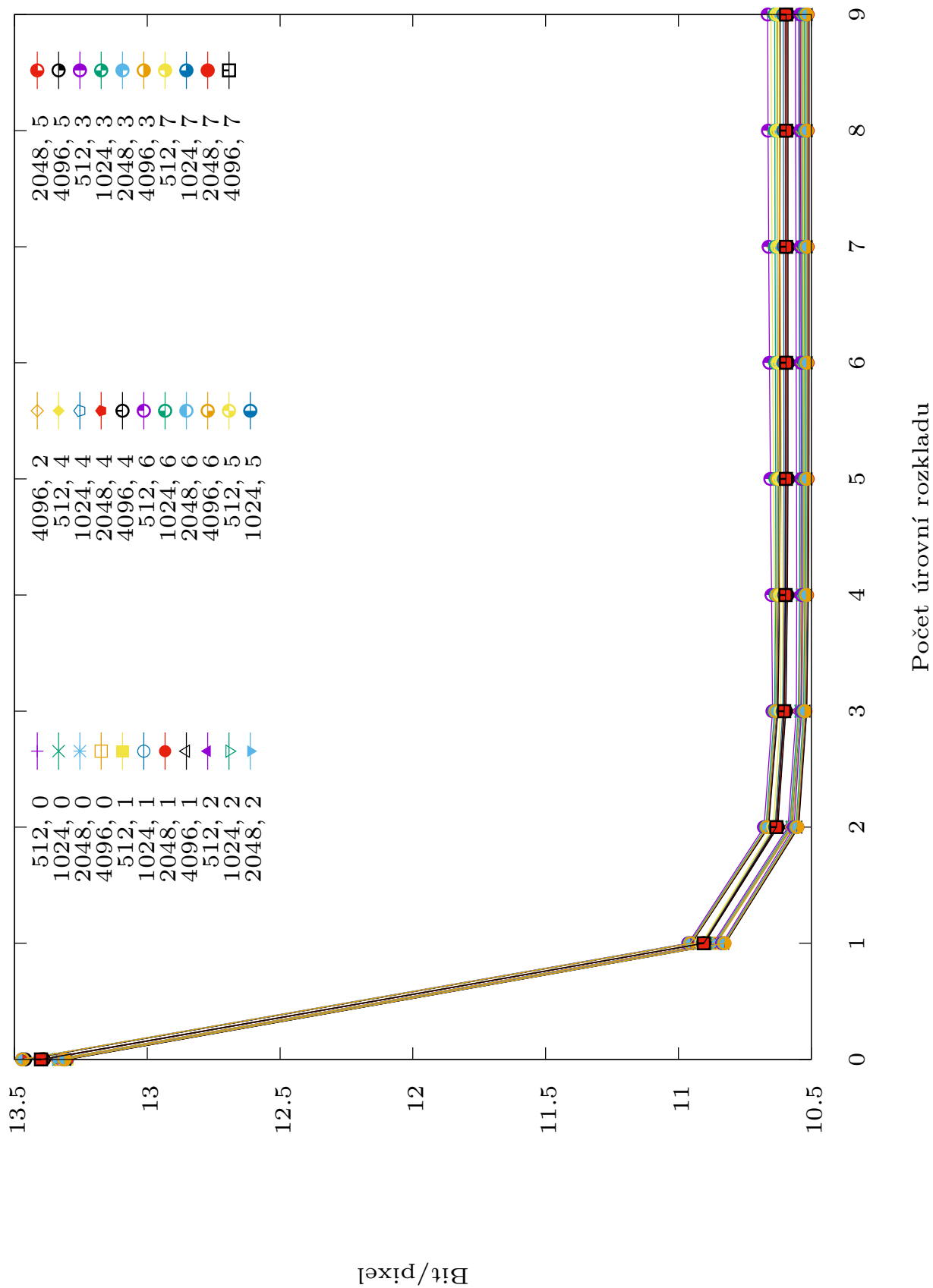
U dekomprese pak nastavení

- dlaždice  $4096 \times 4096$ , mód Bypass
- dlaždice  $2048 \times 2048$ , mód Bypass
- dlaždice  $1024 \times 1024$ , mód Bypass.

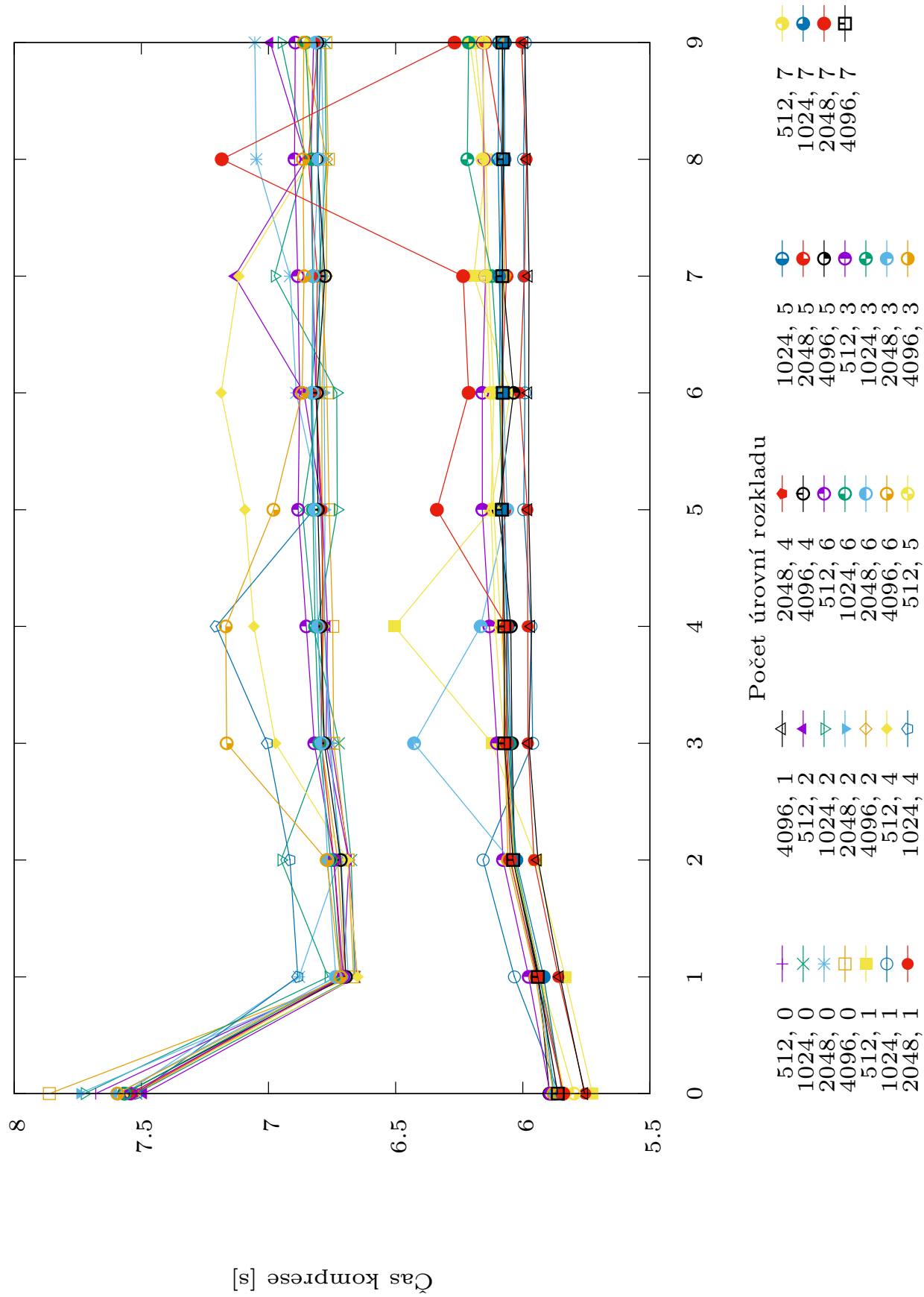
K nejpomalejším nastavením u dekomprese patří

- dlaždice  $512 \times 512$ , mód Restart
- dlaždice  $4096 \times 4096$ , mód Restart+reset.

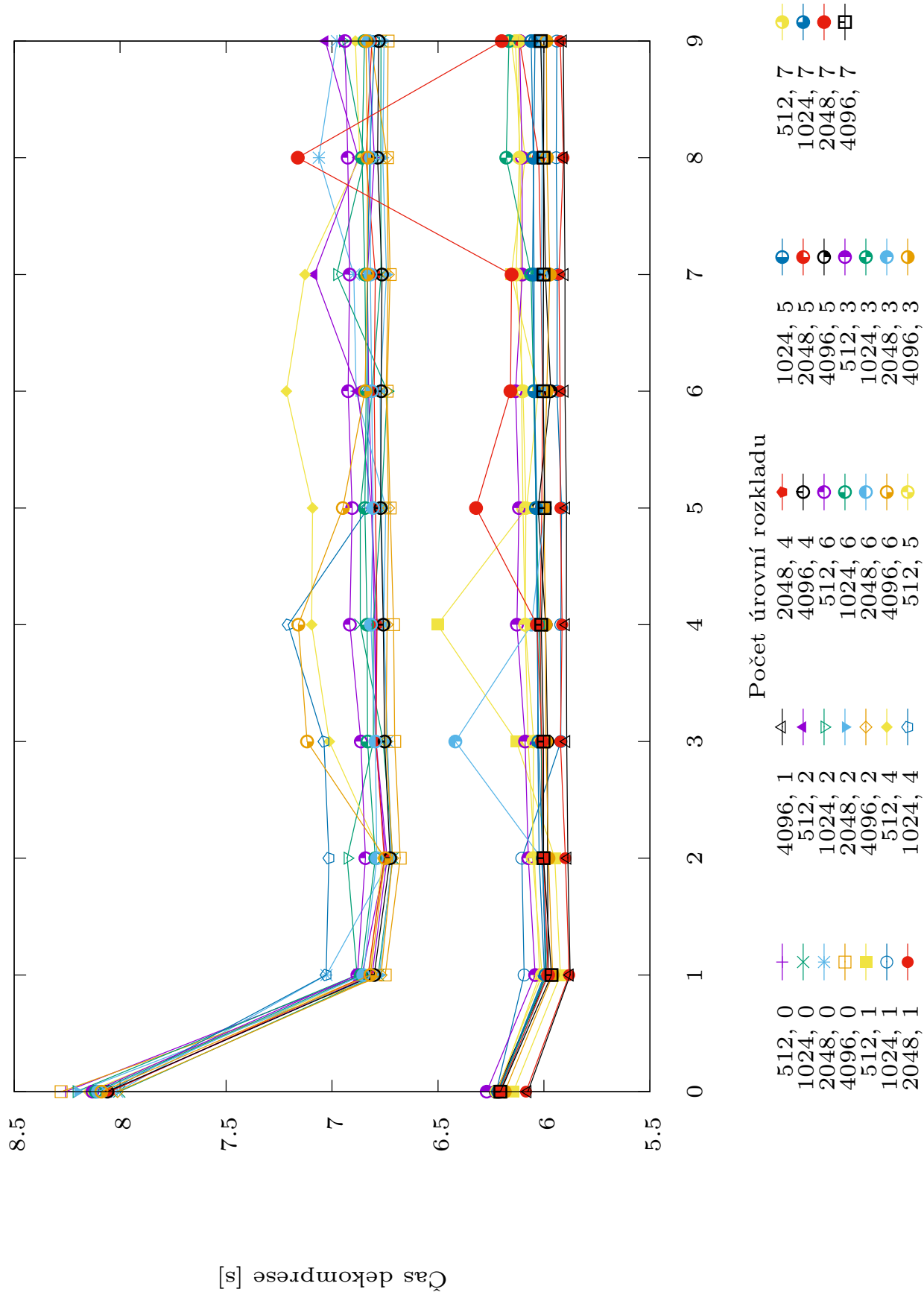
Potvrdilo se také nejlepší nastavení počtu úrovní rozlišení jako u předchozího grafu a to na hodnotu 5.



Obrázek 5.10: Kakadu - bezztrátová komprese, skenované časopisy - průměrný počet bitů na pixel u komprimovaného obrazu



Obrázek 5.11: Kakadu - bezztrátová komprese, skenované časopisy - průměrný čas komprese jednoho obrázu



Obrázek 5.12: Kakadu - beztrátová komprese, skenované časopisy - průměrný čas dekomprese jednoho obrazu

### 5.2.3 Shrnutí

OpenJPEG je o trochu rychlejší jak při dekompresi, tak při kompresi. Oproti tomu Kakadu je sice pomalejší, zato je schopná zachovat větší počet bitů na pixel, při jinak stejných výsledcích. Při použití knihovny OpenJPEG se jeví jako neoptimálnější nastavení dlaždice  $512 \times 512$ , mód Bypass+restart+reset. U knihovny Kakadu potom dlaždice  $512 \times 512$ , mód Restart+reset. Jako nejhorší nastavení vyšlo u obou knihoven téměř jakékoli nastavení s dlaždicí  $4096 \times 4096$ . Počet úrovní rozlišení vyšel jako optimální u obou knihoven 5.

## 5.3 Dataset č.3 - Skenované mapy

Dataset skenované mapy obsahuje skenované celobarevné mapy v rozlišení přibližně 53 Megapixelů.

### 5.3.1 Knihovna OpenJPEG

#### Počet bitů na pixel u komprimovaného obrazu

Z grafu 5.13 je vidět, že se po kompresi počet bitů zmenšil téměř o polovinu.

Mezi jednotlivými nastaveními nejsou opět markantní rozdíly a k nejlepším nastavení pak patří

- dlaždice  $4096 \times 4096$ , mód 1 (Bypass)
- dlaždice  $4096 \times 4096$ , mód 7 (Bypass+restart+reset)
- dlaždice  $2048 \times 2048$ , mód 1 (Bypass)
- dlaždice  $2048 \times 2048$ , mód 7 (Bypass+restart+reset).

Mezi nejhorší potom

- dlaždice  $4096 \times 4096$ , mód 0 (žádný mód)
- dlaždice  $2048 \times 2048$ , mód 0 (žádný mód)
- dlaždice  $4096 \times 4096$ , mód 2 (reset).

Z grafu je také opět vidět, že nevhodnější počet úrovní rozlišení je 5.

#### Čas komprese a dekomprese

V grafech 5.14 a 5.15 lze vidět, že doba kompresi obrazu je přibližně osmnáctkrát delší, než tomu bylo u prvního datasetu. Což je bohužel díky ohromné velikosti těchto souborů očekávaný jev. Čas dekomprese je oproti tomu pouze desetinásobný.

U komprese i dekomprese patří k nejrychlejším nastavení

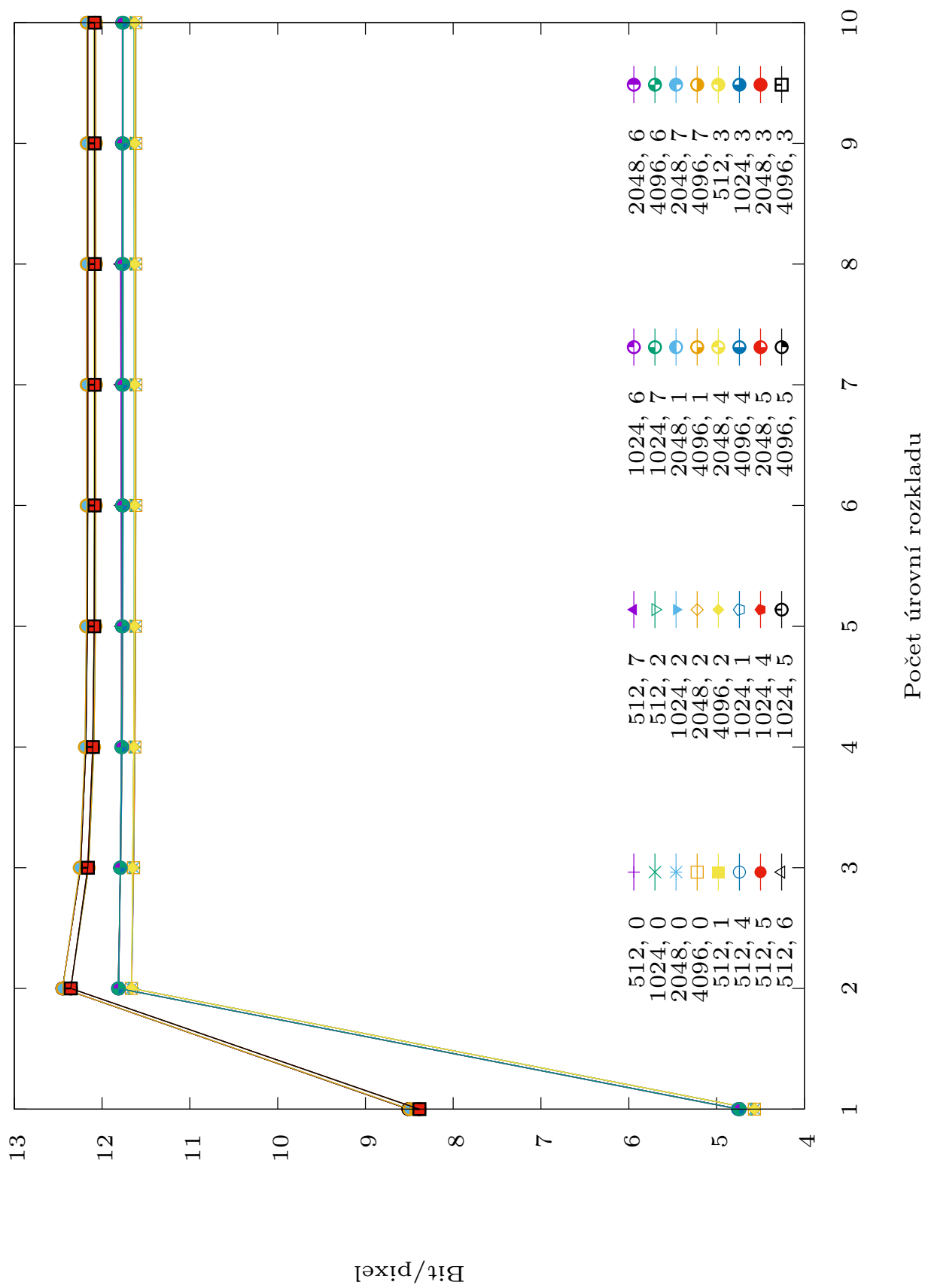
- dlaždice  $512 \times 512$ , mód 1 (Bypass)
- dlaždice  $512 \times 512$ , mód 5 (Bypass+restart)
- dlaždice  $512 \times 512$ , mód 3 (Bypass+reset)
- dlaždice  $1024 \times 1024$ , mód 1 (Bypass).

K nejpomalejším nastavením také u obou pak patří

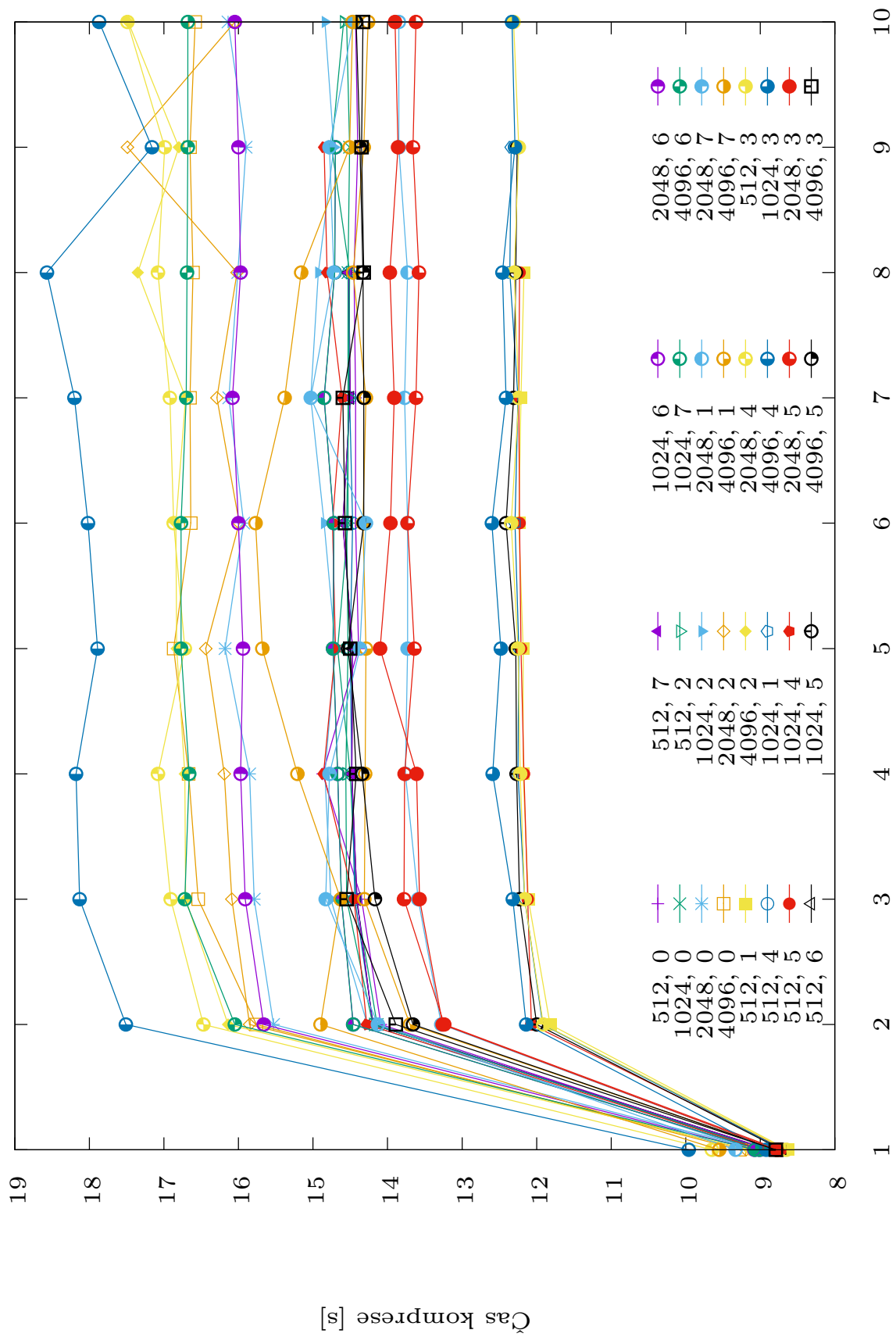
- dlaždice  $4096 \times 4096$ , mód 4 (restart)
- dlaždice  $2048 \times 2048$ , mód 4 (restart)
- dlaždice  $4096 \times 4096$ , mód 2 (reset)
- dlaždice  $4096 \times 4096$ , mód 6 (restart+restart).

Potvrdilo se také nejlepší nastavení počtu úrovní rozlišení jako u předchozího grafu a to na hodnotu 5.



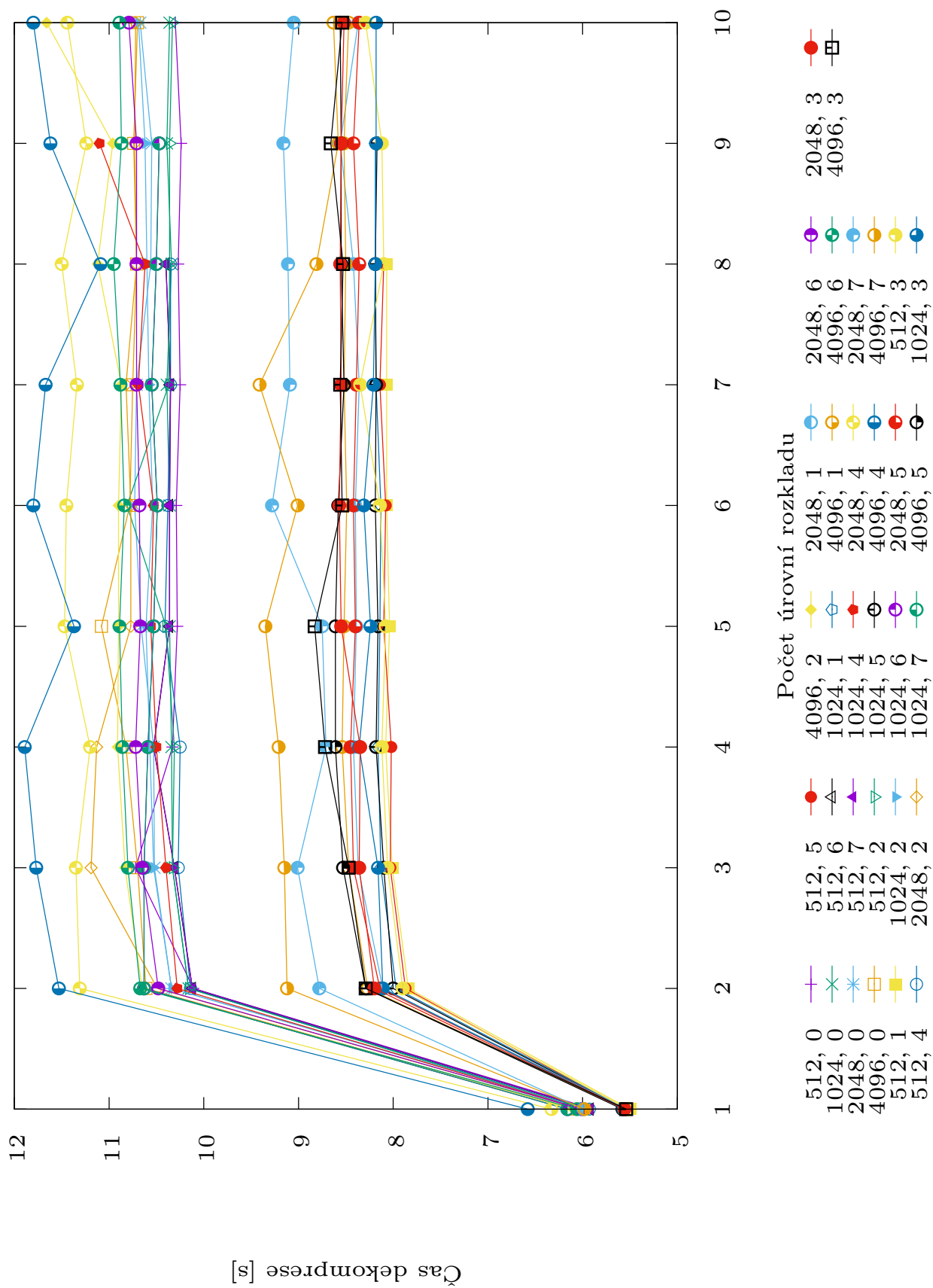


Obrázek 5.13: OpenJPEG - bezztrátová komprese, skenovaná mapa - průměrný počet bitů na pixel u komprimovaného obrazu



Počet úrovní rozkladu

Obrázek 5.14: OpenJPEG - bezztrátová komprese, skenované mapy - průměrný čas komprese jednoho obrazu



Obrázek 5.15: OpenJPEG - bezztrátová komprese, skenované mapy - průměrný čas dekomprese jednoho obrazu

### 5.3.2 Knihovna Kakadu

#### Počet bitů na pixel u komprimovaného obrazu

Z grafu 5.16 je vidět, že se po kompresi počet bitů zmenšil téměř o polovinu.

Mezi jednotlivými nastaveními nejsou opět markantní rozdíly a k nejlepším nastavení pak patří

- dlaždice  $4096 \times 4096$ , mód Bypass+restart+reset
- dlaždice  $2048 \times 2048$ , mód Bypass+restart+reset
- dlaždice  $512 \times 512$ , mód Bypass+restart+reset
- dlaždice  $512 \times 512$ , mód Bypass+restart
- dlaždice  $4096 \times 4096$ , mód Bypass+reset .

Mezi nejhorší potom

- dlaždice  $4096 \times 4096$ , mód 0 (žádný mód)
- dlaždice  $2048 \times 2048$ , mód 0 (žádný mód)
- dlaždice  $2048 \times 2048$ , mód Reset.

Z grafu je také opět vidět, že nevhodnější počet úrovní rozlišení je 5.

#### Čas komprese a dekomprese

V grafech 5.17 a 5.18 lze vidět, že doba komprese i dekomprese obrazu je přibližně šestnáctkrát delší, než tomu bylo u prvního datasetu.

U komprese patří k nejrychlejším nastavení

- dlaždice  $2048 \times 2048$ , mód Bypass+reset
- dlaždice  $4096 \times 4096$ , mód Bypass+reset
- dlaždice  $4096 \times 4096$ , mód Bypass+restart
- dlaždice  $4096 \times 4096$ , mód Bypass
- dlaždice  $1024 \times 1024$ , mód Bypass+restart.

K nejpomalejším nastavením u komprese patří

- dlaždice  $4096 \times 4096$ , mód Restart
- dlaždice  $2048 \times 2048$ , mód Restart
- dlaždice  $1024 \times 1024$ , mód Restart.

U dekomprese pak nastavení

- dlaždice  $4096 \times 4096$ , mód Bypass+reset
- dlaždice  $4096 \times 4096$ , mód Bypass

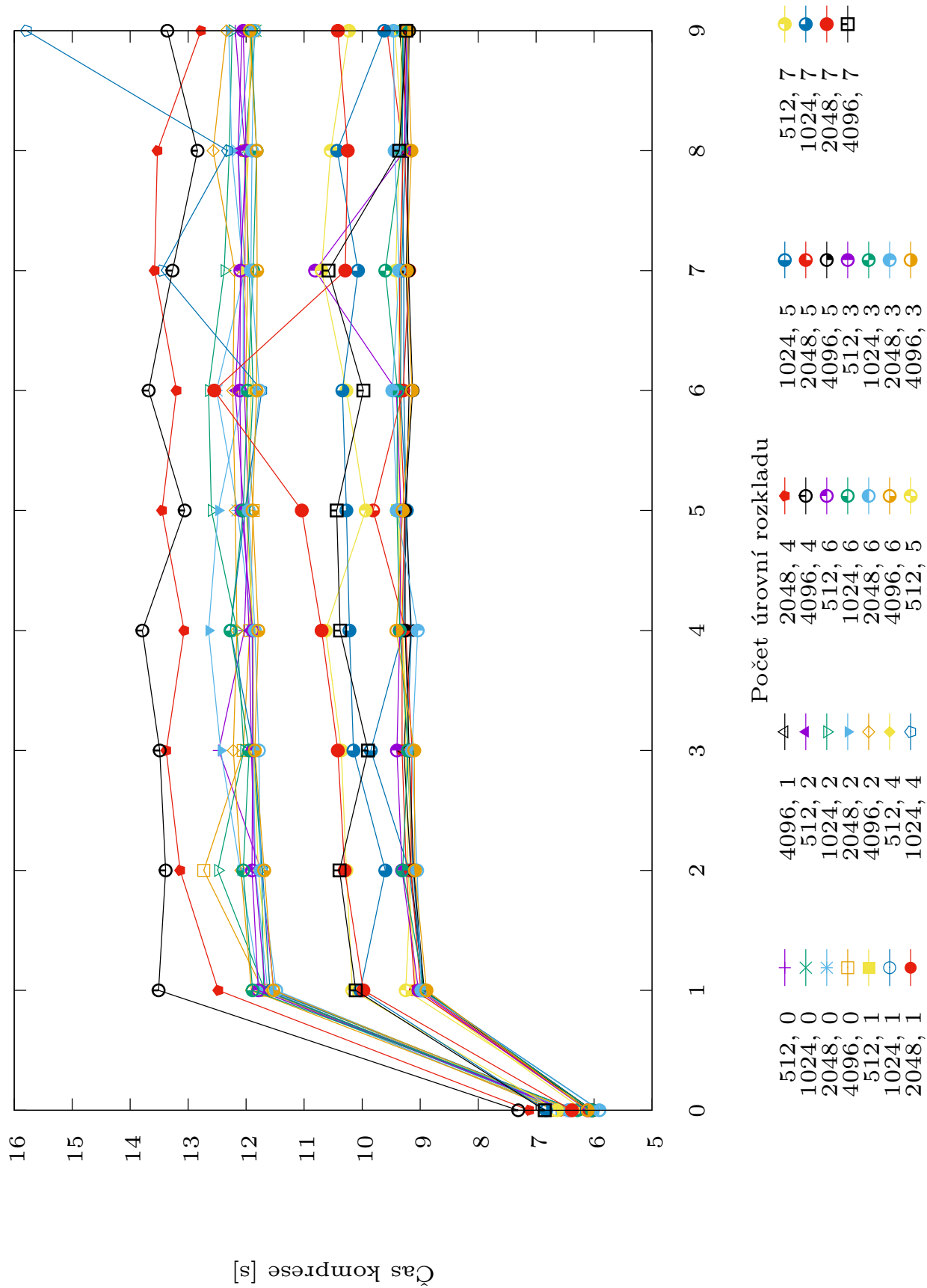
- dlaždice  $2048 \times 2048$ , mód Bypass+reset
- dlaždice  $4096 \times 4096$ , mód Bypass+restart.

K nejpomalejším nastavením u dekomprese patří

- dlaždice  $4096 \times 4096$ , mód Restart
- dlaždice  $2048 \times 2048$ , mód Restart
- dlaždice  $2048 \times 2048$ , mód Reset.

Potvrdilo se také nejlepší nastavení počtu úrovní rozlišení jako u předchozího grafu a to na hodnotu 5.





Obrázek 5.17: Kakadu - bezztrátová komprese, skenované mapy - průměrný čas komprese jednoho obrazu





### 5.3.3 Shrnutí

Při práci s tímto datasetem jsou obě knihovny opět velice vyrovnané. OpenJPEG je o trochu rychlejší při dekompresi, oproti tomu Kakadu je rychlejší při kompresi. V počtu bitů na pixel jsou obě knihovny téměř totožné. Při použití knihovny OpenJPEG je neoptimálnější nastavení dlaždice  $4096 \times 4096$ , mód Bypass. U knihovny Kakadu potom dlaždice  $4096 \times 4096$ , mód Restart+reset. Jako nejhorší nastavení vyšlo u obou knihoven nastavení dlaždice  $4096 \times 4096$ , bez módu. Počet úrovní rozlišení vyšel jako optimální u obou knihoven 5.

## 5.4 Dataset č.4 - Skenované knihy

Dataset skenované knihy obsahuje skenované černobílé knihy bez obrázků (pouze text), které mají rozlišení přibližně 13 Megapixelů.

### 5.4.1 Knihovna OpenJPEG

#### Počet bitů na pixel u komprimovaného obrazu

Z grafu 5.19 je vidět, že se po kompresi počet bitů zmenšil téměř na čtvrtinu, což je očekávané chování, jelikož je možné se na pouhém černobílém textu dopustit větší komprese. Mezi jednotlivými nastaveními nejsou opět markantní rozdíly a k nejlepším nastavení patří

- dlaždice  $4096 \times 4096$ , mód 1 (Bypass)
- dlaždice  $4096 \times 4096$ , mód 7 (Bypass+restart+reset)
- dlaždice  $2048 \times 2048$ , mód 1 (Bypass)
- dlaždice  $2048 \times 2048$ , mód 7 (Bypass+restart+reset)
- dlaždice  $4096 \times 4096$ , mód 3 (Bypass+reset) .

Mezi nejhorší potom

- dlaždice  $4096 \times 4096$ , mód 0 (žádný mód)
- dlaždice  $4096 \times 4096$ , mód 4 (Restart)
- dlaždice  $4096 \times 4096$ , mód 6 (Reset+Restart)
- dlaždice  $4096 \times 4096$ , mód 2 (Reset).

Z grafu je také opět vidět, že nevhodnější počet úrovní rozlišení je 5.

#### Čas komprese a dekomprese

V grafech 5.20 a 5.21 lze vidět, že doba komprese i dekomprese obrazu je přibližně dvakrát delší, než tomu bylo u prvního datasetu, což s přihlédnutím k velikosti je očekávané.

U komprese patří k nejrychlejším nastavení

- dlaždice  $512 \times 512$ , mód 5 (Bypass+restart)
- dlaždice  $1024 \times 1024$ , mód 1 (Bypass)

- dlaždice  $1024 \times 1024$ , mód 5 (Bypass+restart)
- dlaždice  $1024 \times 1024$ , mód 3 (Bypass+reset)
- dlaždice  $512 \times 512$ , mód 1 (Bypass).

K nejpomalejším nastavením u komprese patří

- dlaždice  $4096 \times 4096$ , mód 0 (žádný mód)
- dlaždice  $4096 \times 4096$ , mód 4 (Restart)
- dlaždice  $4096 \times 4096$ , mód 6 (Restart+Reset)
- dlaždice  $4096 \times 4096$ , mód 2 (Reset).

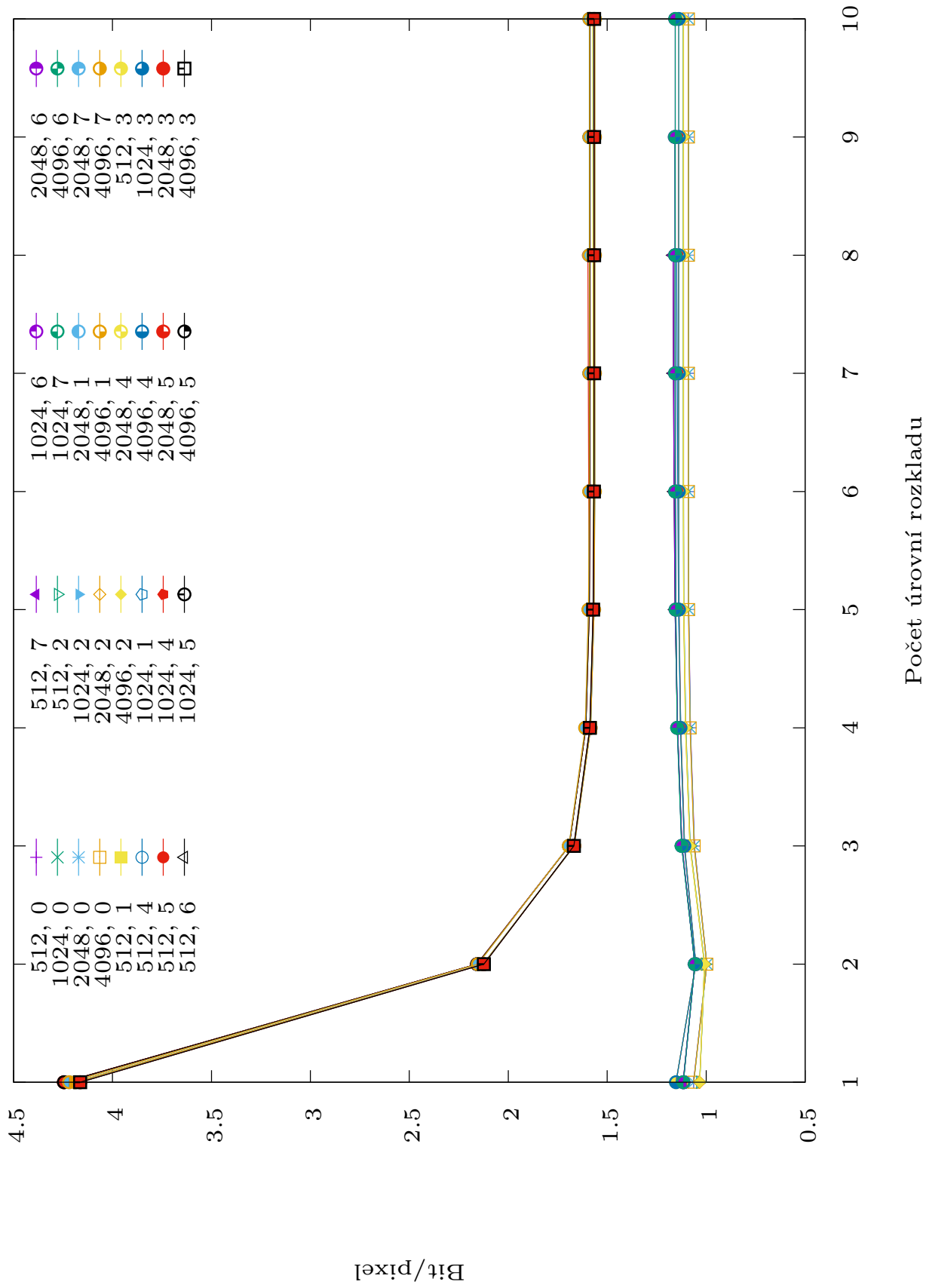
U dekomprese pak nastavení

- dlaždice  $512 \times 512$ , mód 5 (Bypass+restart)
- dlaždice  $512 \times 512$ , mód 3 (Bypass+reset)
- dlaždice  $512 \times 512$ , mód 1 (Bypass).

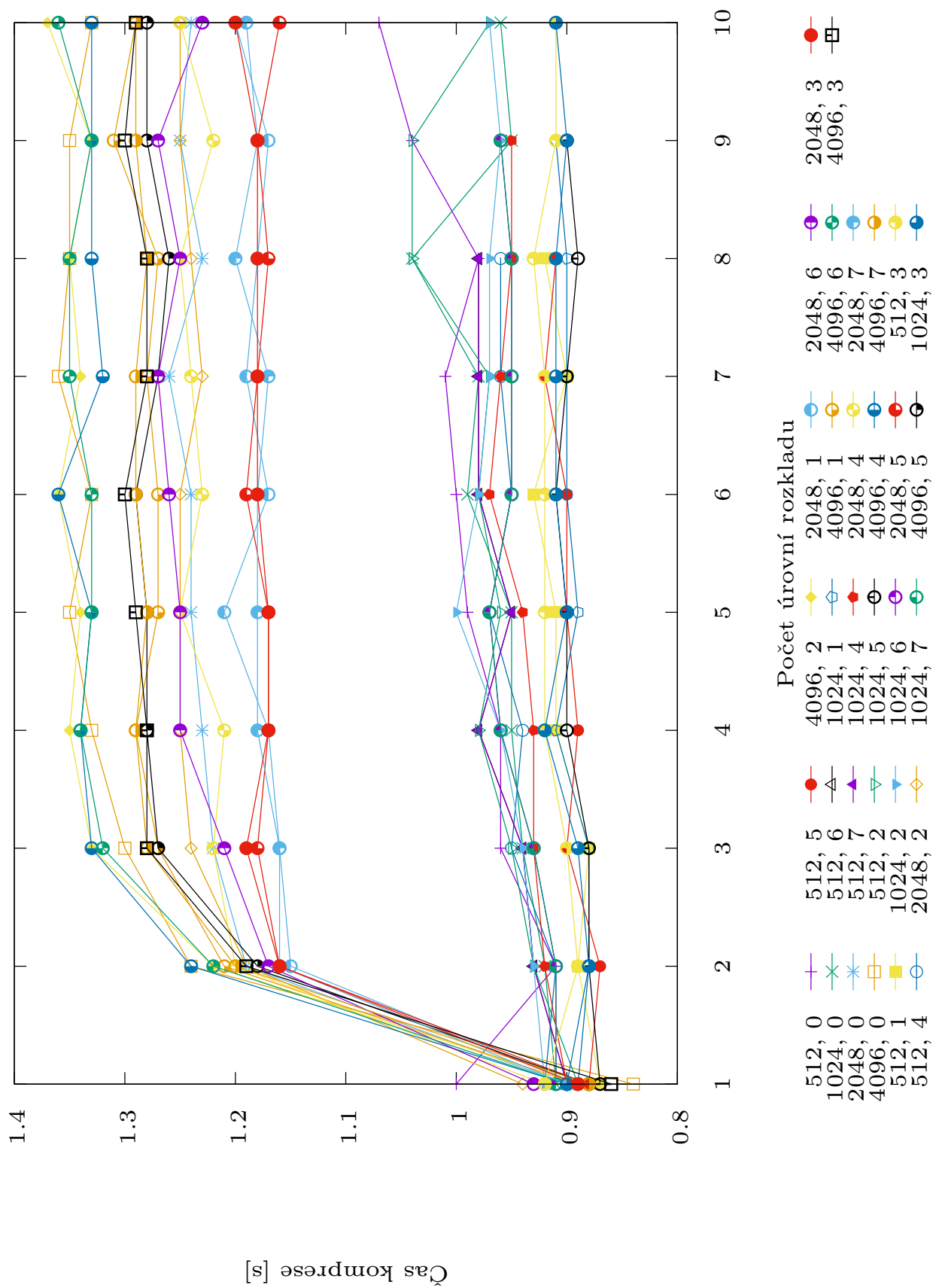
K nejpomalejším nastavením u dekomprese patří

- dlaždice  $4096 \times 4096$ , mód 0 (žádný mód)
- dlaždice  $4096 \times 4096$ , mód 2 (Reset)
- dlaždice  $4096 \times 4096$ , mód 4 (Restart)
- dlaždice  $4096 \times 4096$ , mód 6 (Restart+Reset).

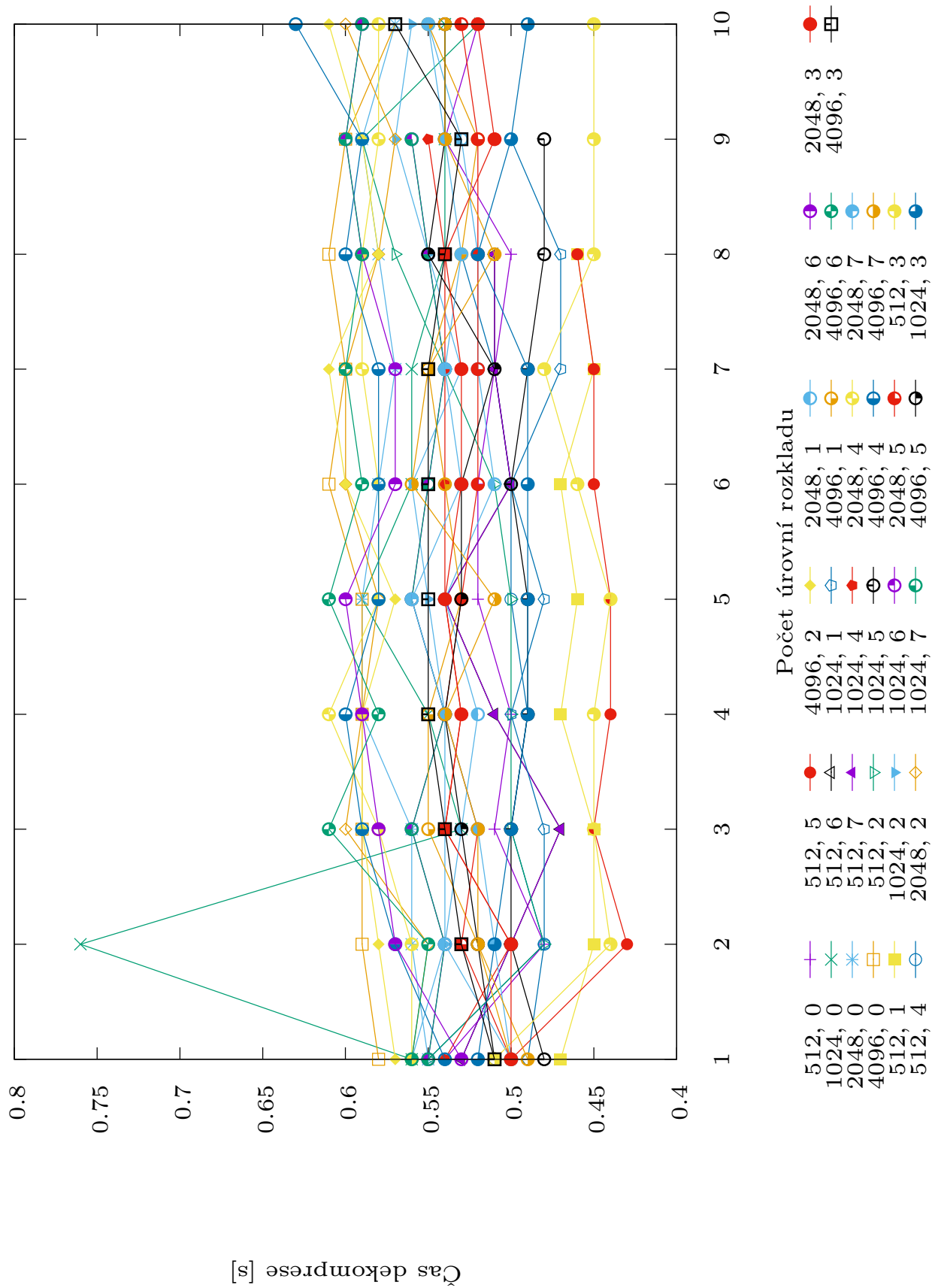
Potvrdilo se také nejlepší nastavení počtu úrovní rozlišení jako u předchozího grafu a to na hodnotu 5.



Obrázek 5.19: OpenJPEG - bezztrátová komprese, skenované knihy - průměrný počet bitů na pixel u komprimovaného obrazu



Obrázek 5.20: OpenJPEG - bezeztrátová komprese, skenované knihy - průměrný čas komprese jednoho obrázku



Obrázek 5.21: OpenJPEG - bezztrátová komprese, skenované knihy - průměrný čas dekomprese jednoho obrazu

## 5.4.2 Knihovna Kakadu

### Počet bitů na pixel u komprimovaného obrazu

Z grafu 5.22 je vidět, že se po kompresi počet bitů zmenšil téměř na desetinu. Mezi jednotlivými nastaveními jsou vidět lehké rozdíly (pokud přihlídneme k měřítku, rozdíl není moc velký) a k nejlepším nastavení pak patří

- dlaždice  $4096 \times 4096$ , mód Bypass+restart+reset
- dlaždice  $2048 \times 2048$ , mód Bypass+restart+reset
- dlaždice  $512 \times 512$ , mód Bypass+restart+reset
- dlaždice  $4096 \times 4096$ , mód Bypass+restart
- dlaždice  $215 \times 512$ , mód Bypass+restart. .

Mezi nejhorší potom

- dlaždice  $4096 \times 4096$ , mód 0 (žádný mód)
- dlaždice  $2048 \times 2048$ , mód 0 (žádný mód)
- dlaždice  $4096 \times 4096$ , mód Restart

Z grafu je také opět vidět, že nevhodnější počet úrovní rozlišení je 5.

### Čas komprese a dekomprese

V grafech 5.23 a 5.24 lze vidět, že doba komprese i dekomprese obrazu je opět téměř zanedbatelná.

U komprese patří k nejrychlejším nastavení

- dlaždice  $512 \times 512$ , mód Bypass+reset+restart
- dlaždice  $1024 \times 1024$ , mód Bypass+restart
- dlaždice  $1024 \times 1024$ , mód Bypass+restart+reset
- dlaždice  $512 \times 512$ , mód Bypass+restart
- dlaždice  $2048 \times 2048$ , mód Bypass+reset.

K nejpomalejším nastavením u komprese patří

- dlaždice  $1024 \times 1024$ , mód 0 (žádný mód)
- dlaždice  $4096 \times 4096$ , mód Reset
- dlaždice  $4096 \times 4096$ , mód Bypass.

U dekomprese pak nastavení

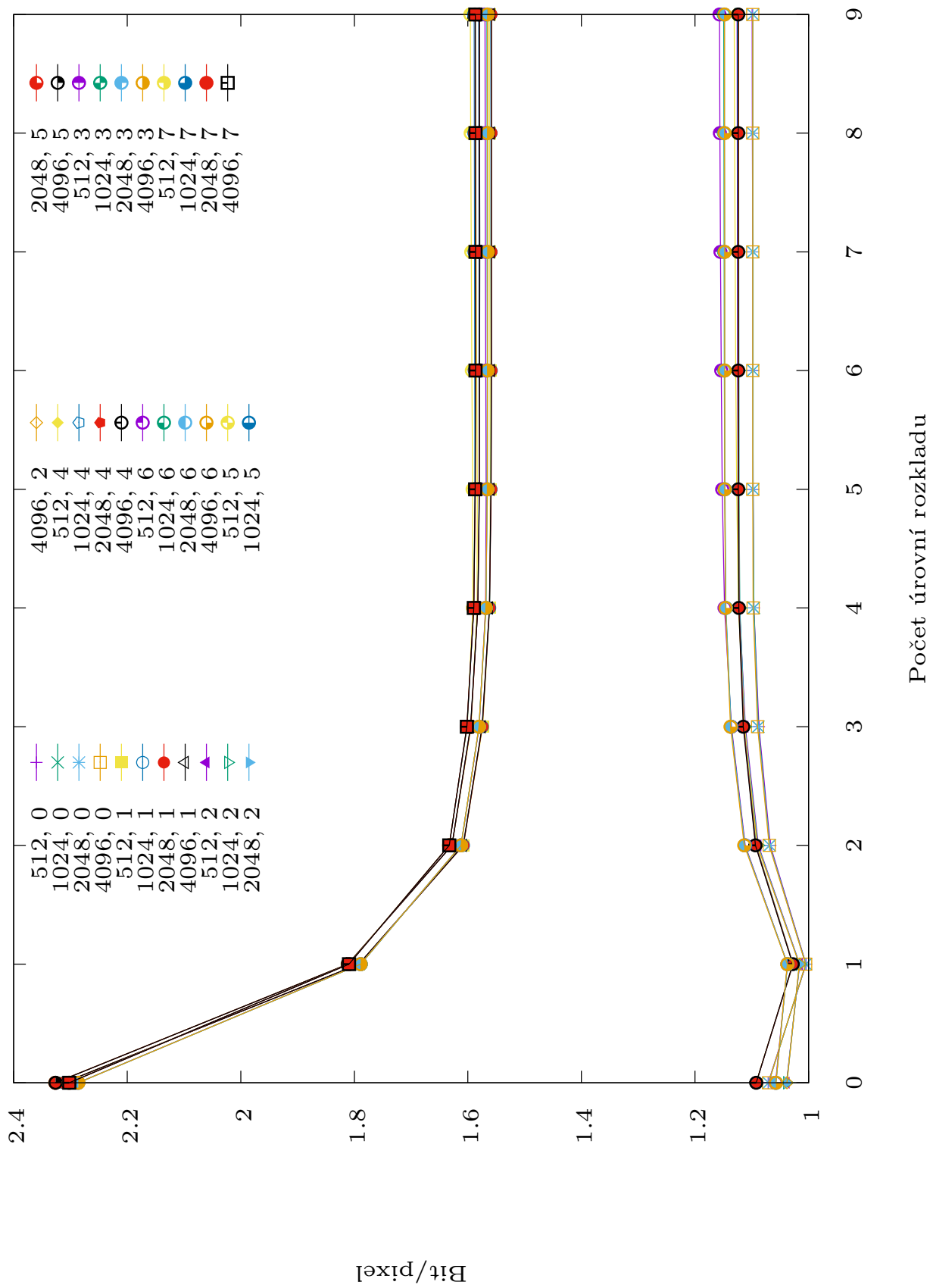
- dlaždice  $1024 \times 1024$ , mód Bypass+reset+restart
- dlaždice  $1024 \times 1024$ , mód Bypass+restart

- dlaždice  $2048 \times 2048$ , mód Bypass+reset+restart
- dlaždice  $4096 \times 4096$ , mód Bypass+restart.

K nejpomalejším nastavením u dekomprese patří

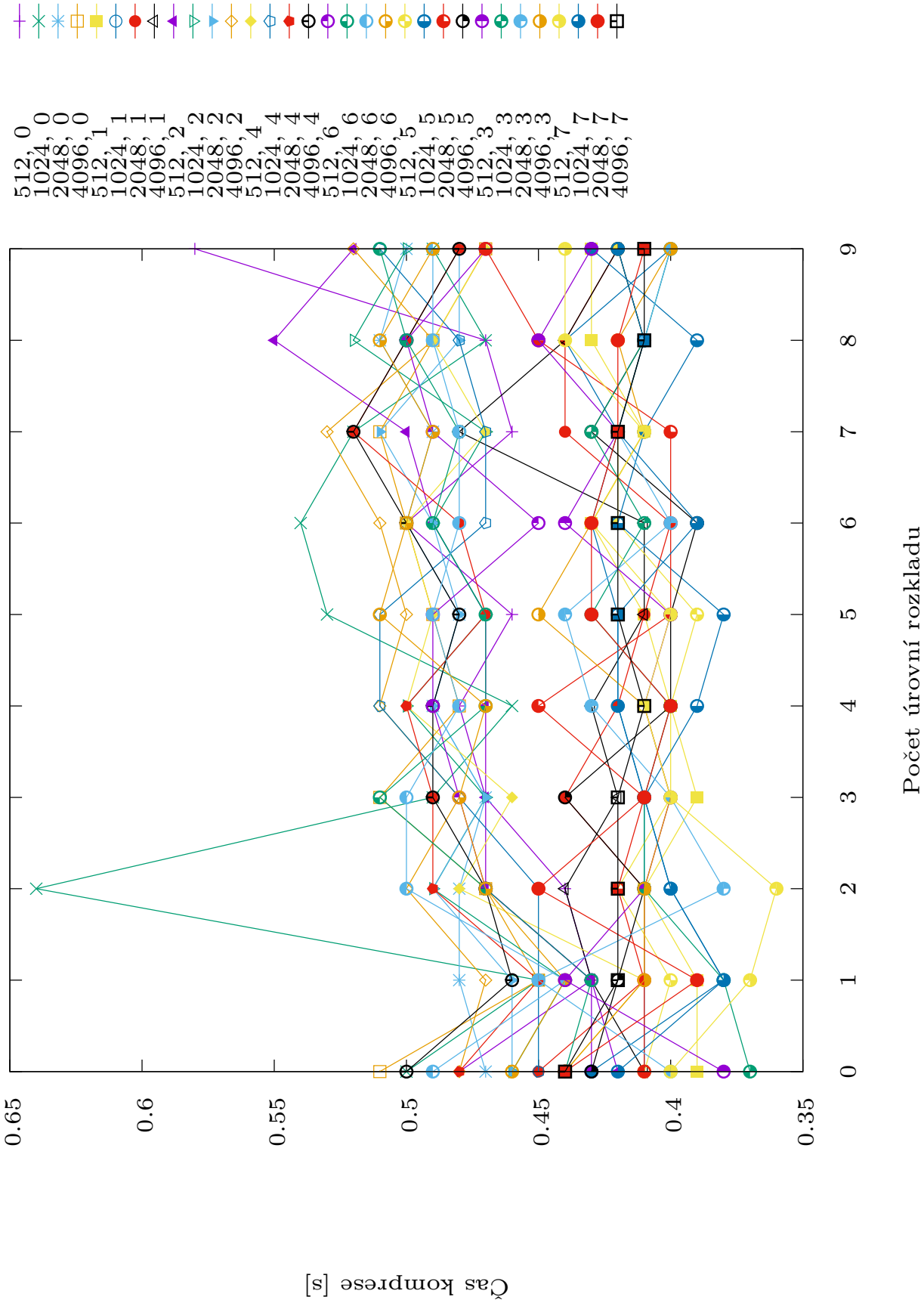
- dlaždice  $512 \times 512$ , mód Reset
- dlaždice  $1024 \times 1024$ , mód Reset
- dlaždice  $1024 \times 1024$ , mód 0 (žádný mód).

Potvrdilo se také nejlepší nastavení počtu úrovní rozlišení jako u předchozího grafu a to na hodnotu 5.

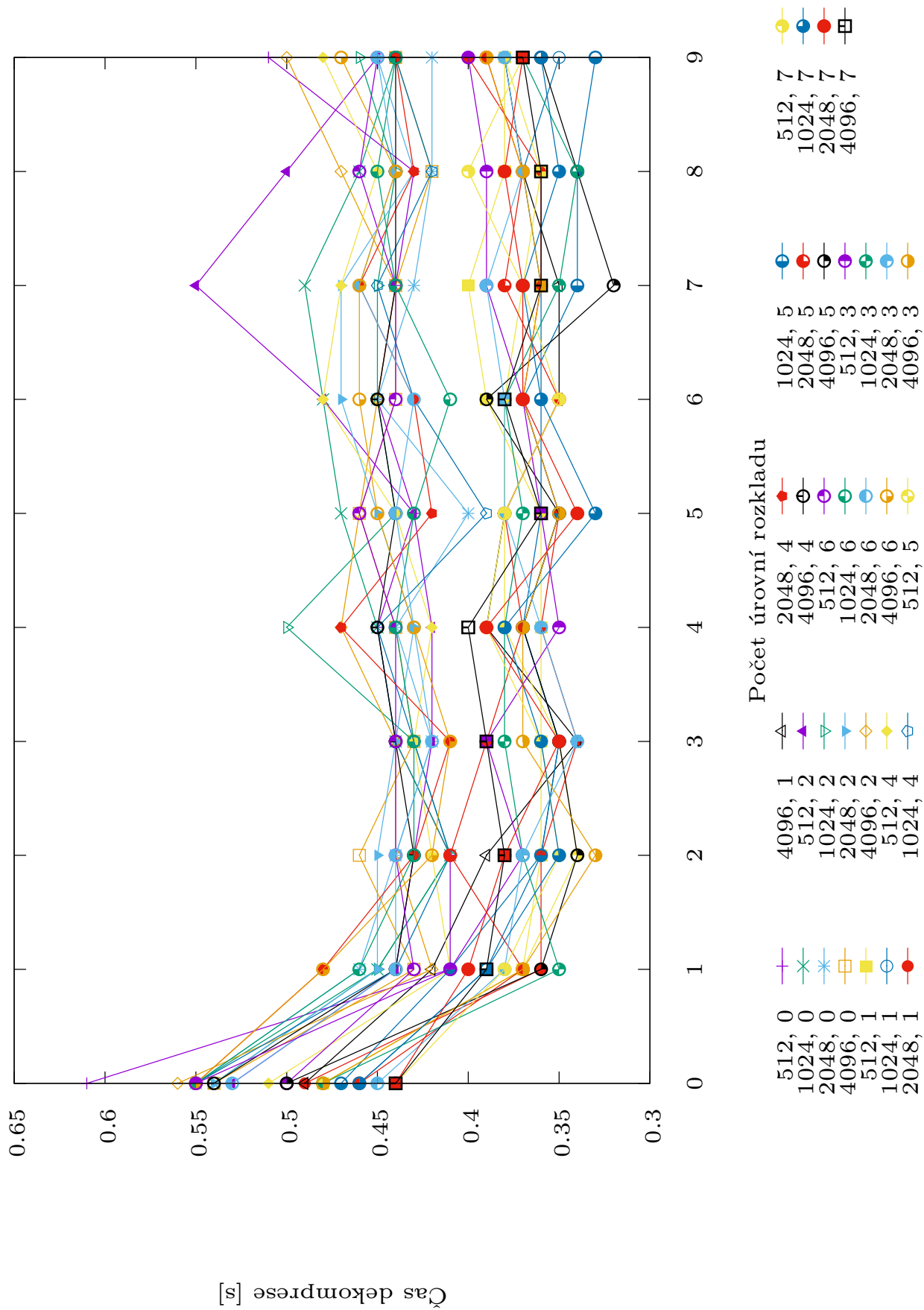


Obrázek 5.22: Kakadu - bezztrátová komprese, skenované knihy - průměrný počet bitů na pixel u komprimovaného obrazu





Obrázek 5.23: Kakadu - bezztrátová komprese, skenované knihy - průměrný čas komprese jednoho obrazu



Obrázek 5.24: Kakadu - bezztrátová komprese, skenované knihy - průměrný čas dekomprese jednoho obrazu

### 5.4.3 Shrnutí

Kakadu je o trochu rychlejší jak při kompresi, tak při dekompresi. Oproti tomu OpenJPEG zachovává větší počet bitů na pixel. Při použití knihovny OpenJPEG je neoptimálnější nastavení dlaždice  $4096 \times 4096$ , mód Bypass. U knihovny Kakadu potom dlaždice  $4096 \times 4096$ , mód Restart+reset. Jako nejhorší nastavení vyšlo u obou knihoven nastavení dlaždice  $4096 \times 4096$ , bez módu. Počet úrovní rozlišení vyšel jako optimální u obou knihoven 5. Analýza tohoto datasetu dopadla v mnoha ohledech stejně, jako nastavení které používá Národní digitální knihovna [15].

## Kapitola 6

# Analýza nastavení JPEG 2000 pro ztrátovou kompresi

Ztrátová komprese se používá, v případě požadavku na co nejmenší prostorou zátěž, při určitém stupni ztrátovosti dat. Většinou se využívá při kompresi fotografií, za účelem úspory místa na disku nebo pro přenos obrazu po síti.

Pro OpenJPEG byly při této kompresi nastaveny tyto parametry:

- -r 8 - kompresní poměr. Číslo 8 odpovídá osminásobnému zmenšení tedy 1:8.
- -n 1-10 - počet úrovní rozlišení, nebo-li počet provedených DWT dekompozicí + 1. Tento parametr patří k jednomu ze zkoumaných a proto je uveden v rozsahu 1-10
- -t 512 × 512|1096 × 1096|2048 × 2048|4096 × 4096 - velikost dlaždice, je taktéž jedním ze zkoumaných parametrů. Udává velikost dlaždic, po kterých se obraz zpracovává. Při větší velikosti vstupního obrazu, by bylo možné přidat ještě dlaždici 8192x8192
- -b 64 × 64 - velikost code-blocks. Tento parametr je pevně daný, jelikož se po několika experimentech ukázala tato velikost jako nejvhodnější.
- -p RPCL - pořadí paket v code streamu. Jelikož se jedná o všeobecně používaný profil pořadí, nebylo zde k jeho nastavení vyžadováno žádných experimentů
- -d 0, 0 - umístění počátku souřadnic obrazu. Experimentováno s počátkem v  $\langle 0, 0 \rangle$  a  $\langle 1, 1 \rangle$ . Nejlépe vyšel počátek v  $\langle 0, 0 \rangle$
- -M 0, 1 - použití módů. Označení jednotlivých módů je číselné. BYPASS [1].
- -I - zapnutí ztrátové komprese.

U dekomprese nebyly nastaveny žádné speciální parametry. Pouze vstupní a výstupní soubor.

U knihovny Kakadu bylo pro kompresi obdobné nastavení, a to:

- Creversible = no - explicitní volba ztrátové, či bezztrátové komprese.
- Stiles = 512 × 512|1096 × 1096|2048 × 2048|4096 × 4096 - velikost dlaždice.
- Clevels = 0-9 - počet úrovní rozlišení. Zde je na rozdíl od OpenJPEGu rozsah posunut o jedno směrem dolů, jelikož přímo odpovídá počtu DWT dekompozice.

- Clayers = 1 - počet vrstev kvality
- Cblk =  $64 \times 64$  - velikost code blocks
- Corder = RPCL - pořadí paket v code streamu
- Cmodes = 0, BYPASS - použití módů, tentokrát jsou označeny slovně
- Sorigin = 0,0 - umístění počátku souřadnic obrazu, tzv. offsetu
- -num\_threads 4 - nastavení počtu vláken, ve kterých může běžet (Kakadu umí pracovat s vlákny).
- -rate 3 - kompresní poměr. Číslo 3 kompresnímu poměru 1:8.

U dekomprese opět nebyly nastaveny žádné speciální parametry. Pouze vstupní a výstupní soubor.

U ztrátové komprese byl zkoumán čas komprese, PSNR u komprimovaného obrazu, PSNR u dekomprimovaného obrazu a čas dekomprese v závislosti na počtu úrovní rozlišení.

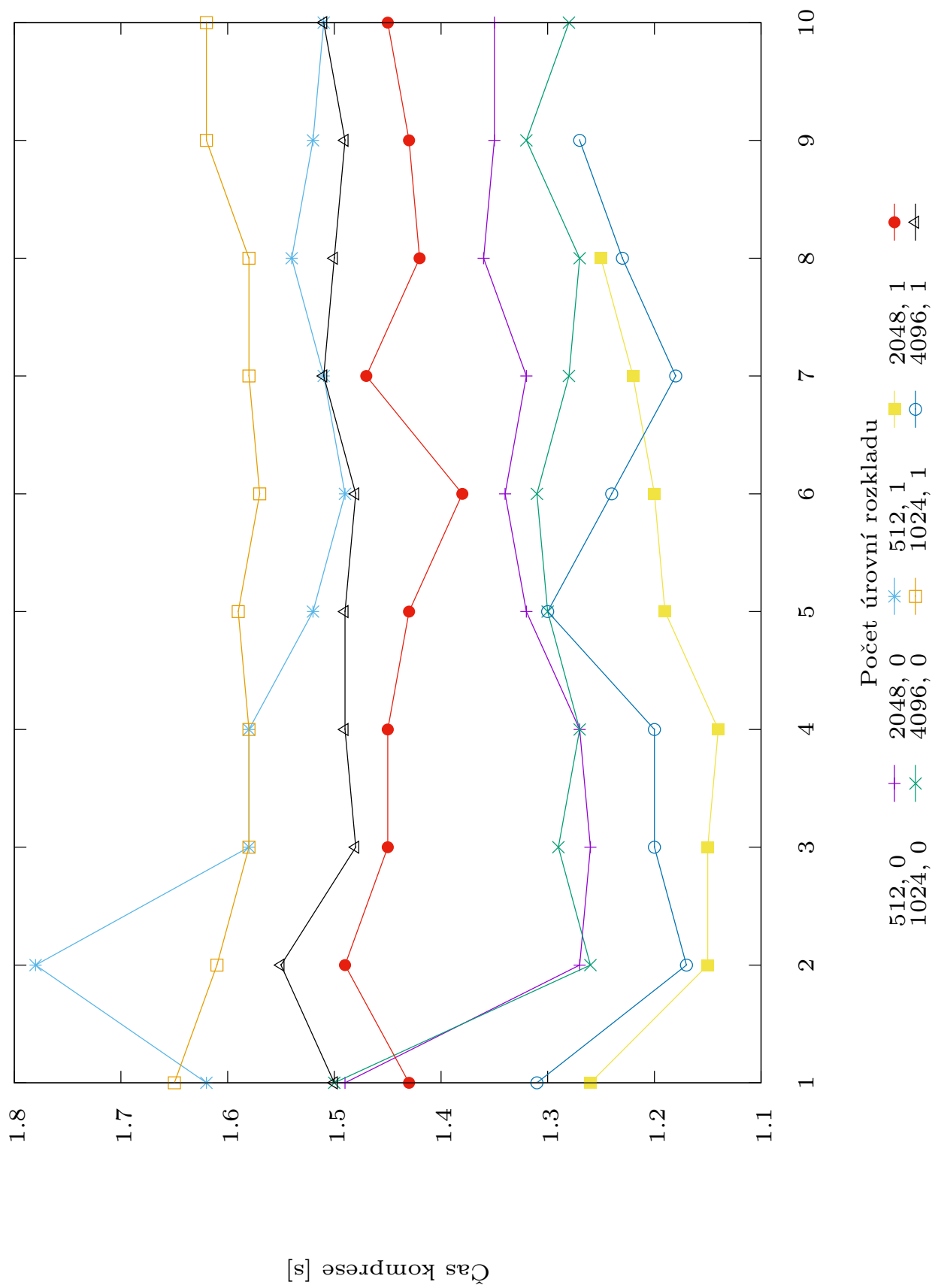
Kvůli padajícímu OpenJPEGu při určitém nastavení, zmíněném v sekci 4.1 chybí v grafech některá data.

Při analýze ztrátové komprese bylo experimentováno pouze s bypass módem, jelikož se ukázal jako nejúčinnější.

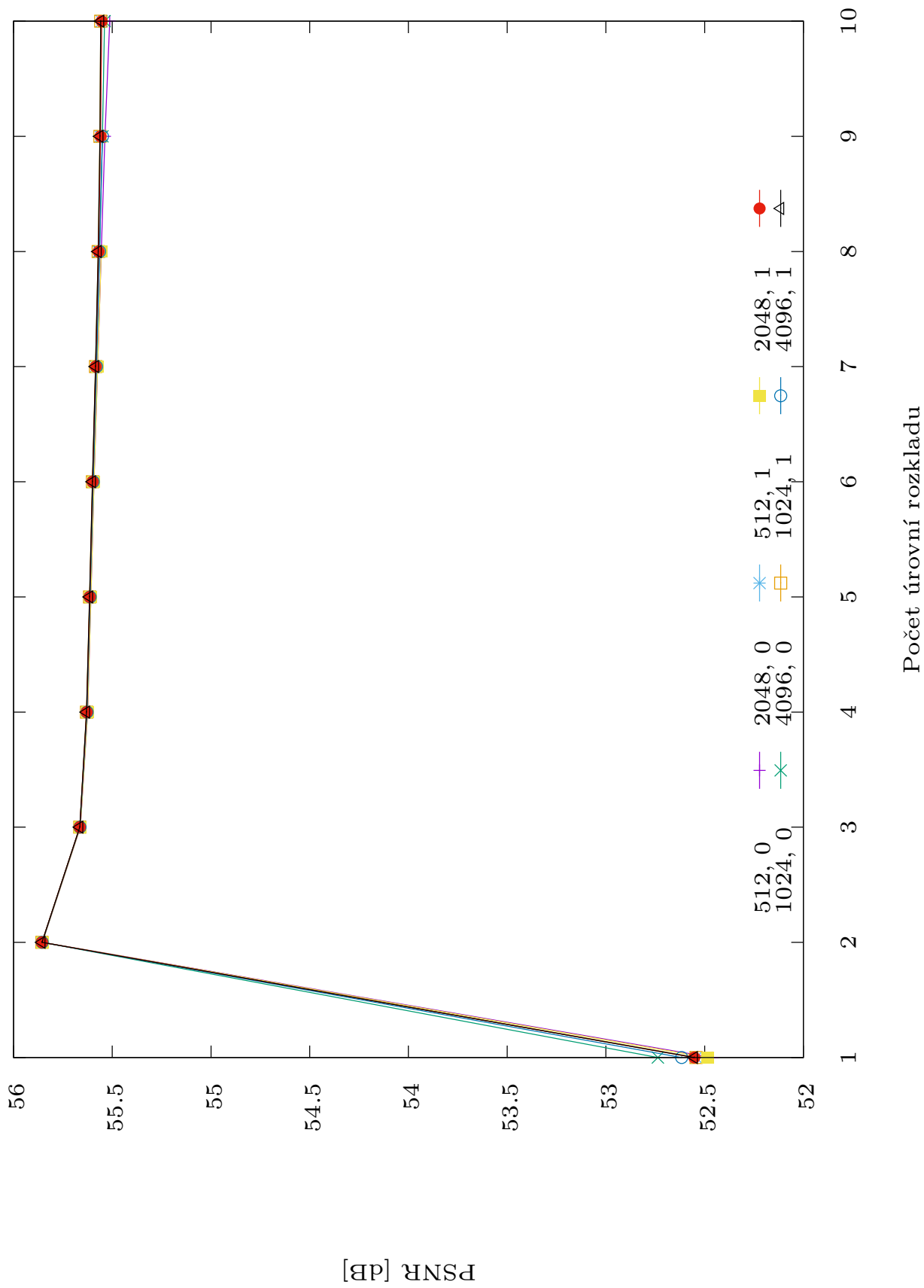
## 6.1 Dataset č.1 - 6 Mpx fotografie

### 6.1.1 Knihovna OpenJPEG

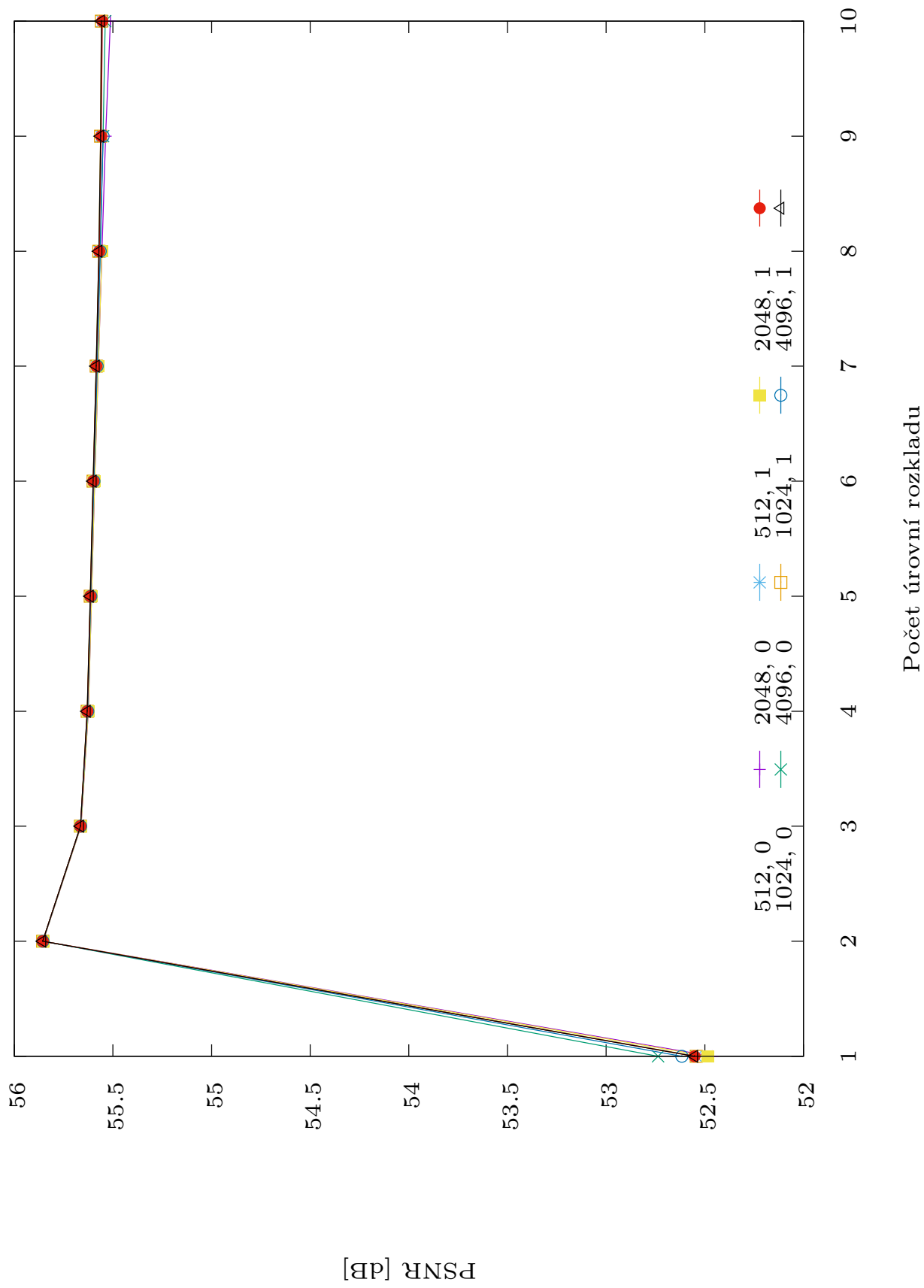
Pro dataset fotografií se v grafech 6.1, 6.2, 6.3 a 6.4 ukázalo, že nejvhodnější je použití módu bypass, jelikož je ve všech směrech rychlejší než běh bez módu. PSNR je ve všech případech shodné. Jako nejvhodnější lze tedy jednoznačně doporučit nastavení dlaždice  $512 \times 512$ , mód bypass, počet úrovní rozlišení 5.



Obrázek 6.1: OpenJPEG - ztrátová komprese, fotografie s rozlišením 6Mpx - průměrný čas komprese jednoho obrázu

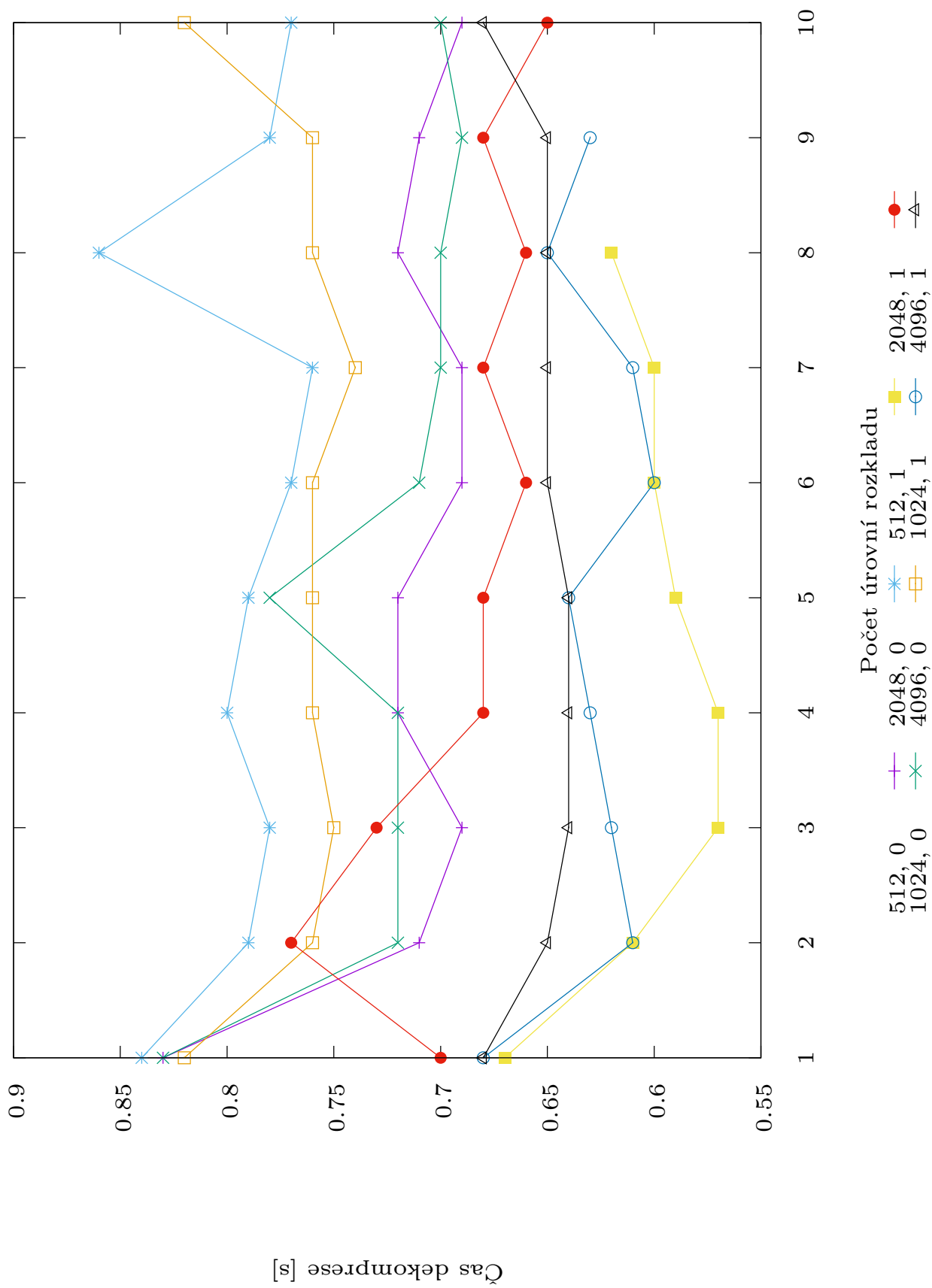


Obrázek 6.2: OpenJPEG - ztrátová komprese, fotografie s rozlišením 6Mpx - průměrné PSNR komprimovaného obrazu



Obrázek 6.3: Open.JPEG - ztrátová komprese, fotografie s rozlišením 6Mpx - průměrné PSNR dekomprimovaného obrazu

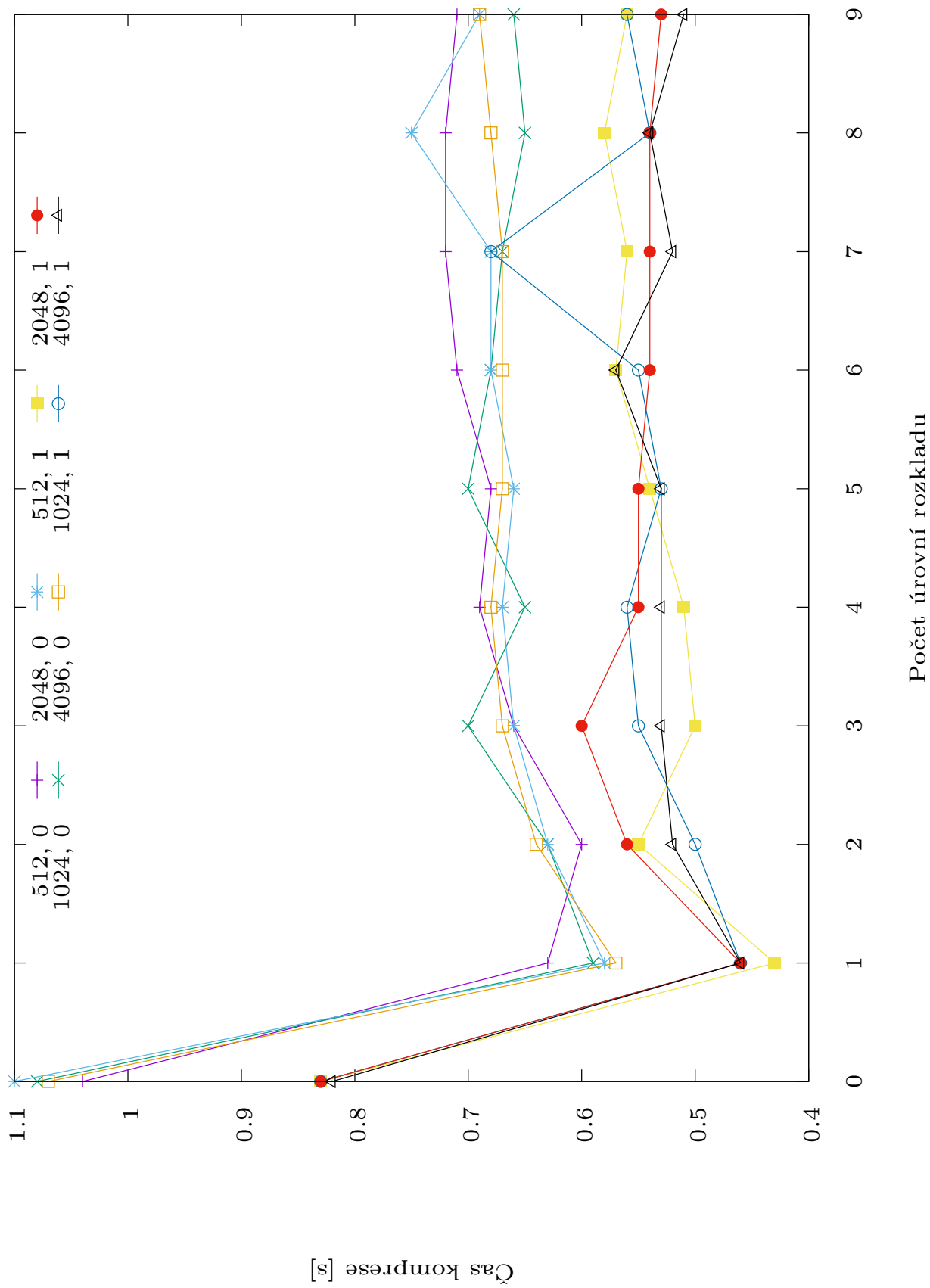




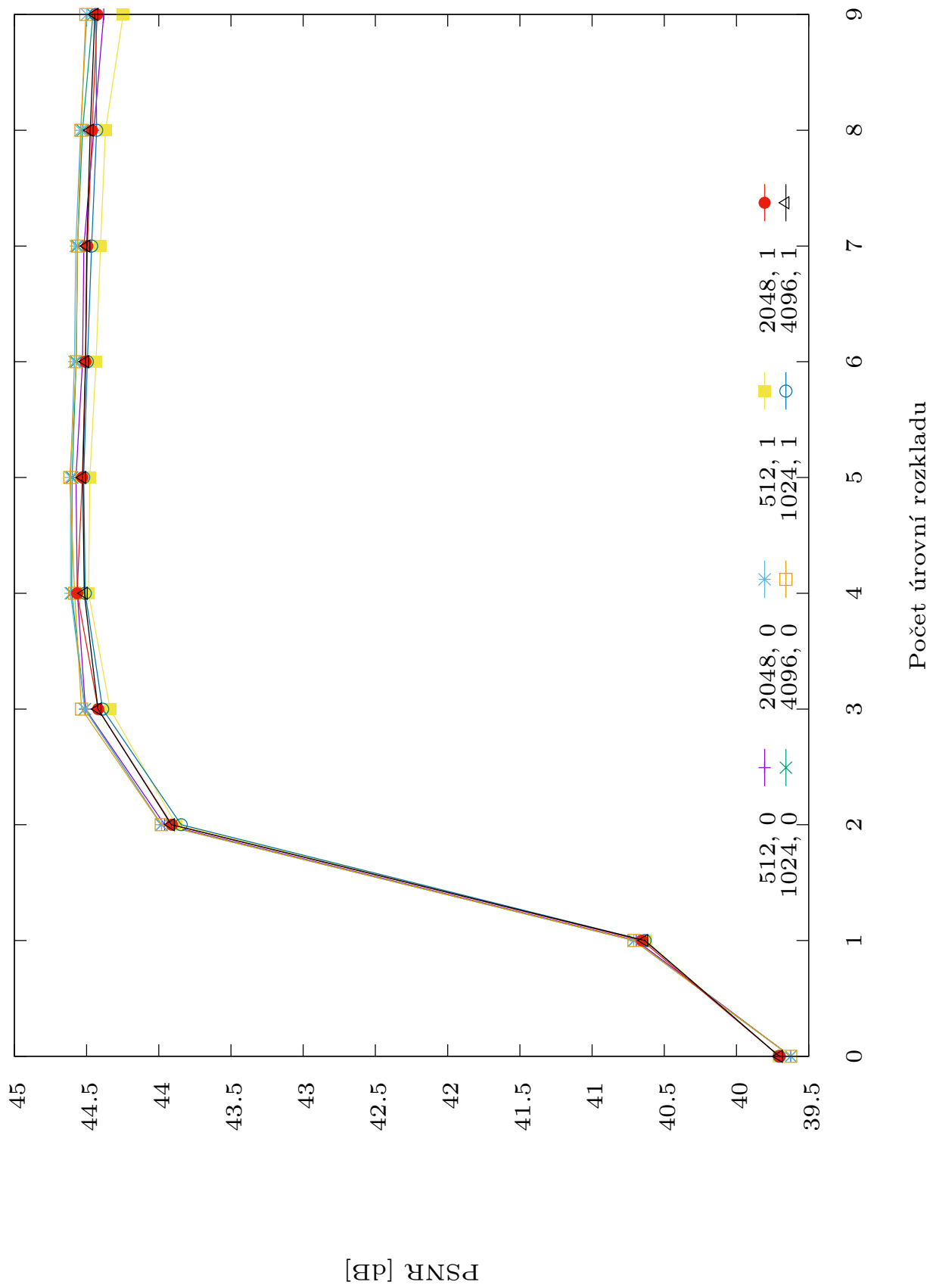
Obrázek 6.4: OpenJPEG - ztrátová komprese, fotografie s rozlišením 6Mpx - průměrný čas dekompresce jednoho obrazu

### 6.1.2 Knihovna Kakadu

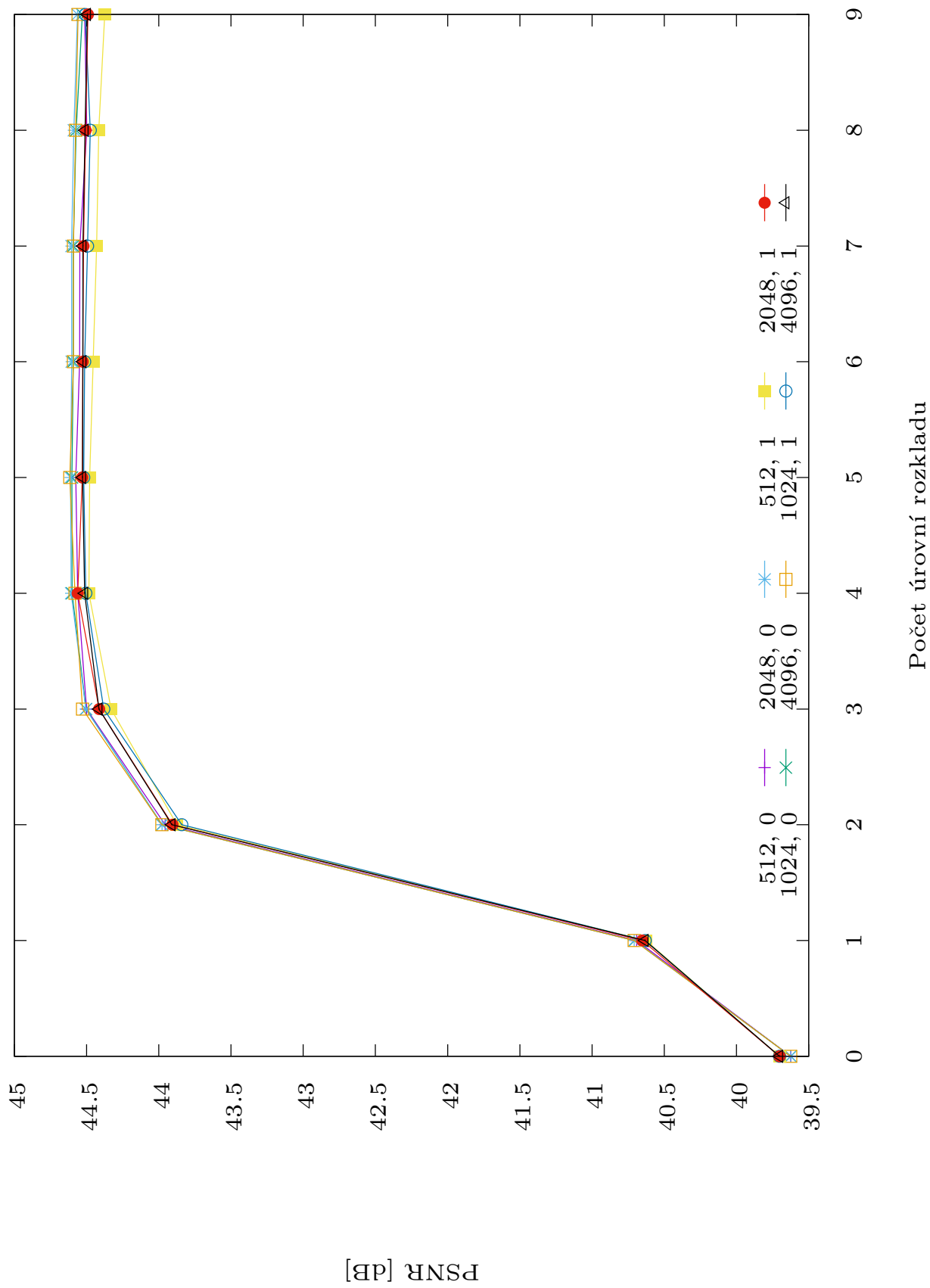
Pro dataset fotografií se v grafech 6.5, 6.6, 6.7 a 6.8 ukázalo, že nejvhodnější je použití módu bypass, jelikož je ve všech směrech rychlejší než běh bez módu. PSNR je ve všech případech shodné. Jako nejvhodnější lze tedy jednoznačně doporučit nastavení dlaždice  $2048 \times 2048$ , mód bypass, počet úrovní rozlišení 5.



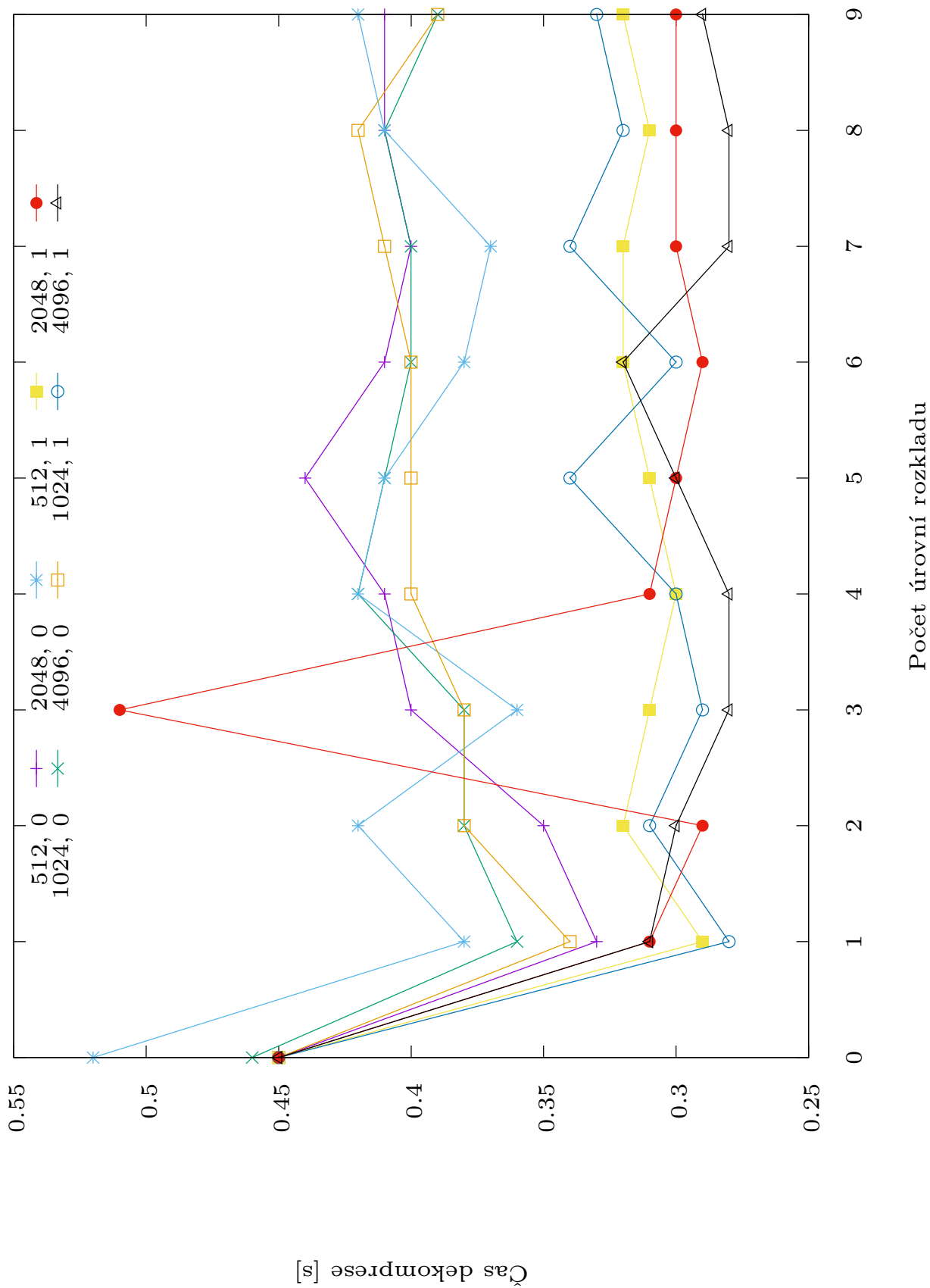
Obrázek 6.5: Kakadu - ztrátová komprese, fotografie s rozlišením 6Mpx - průměrný čas komprese jednoho obrazu



Obrázek 6.6: Kakadu - ztrátová komprese, fotografie s rozlišením 6Mpx - průměrné PSNR komprimovaného obrazu



Obrázek 6.7: Kakadu - ztrátová komprese, fotografie s rozlišením 6Mpx - průměrné PSNR dekomprimovaného obrazu



Obrázek 6.8: Kakadu - ztrátová komprese, fotografie s rozlišením 6Mpx - průměrný čas dekomprese jednoho obrazu

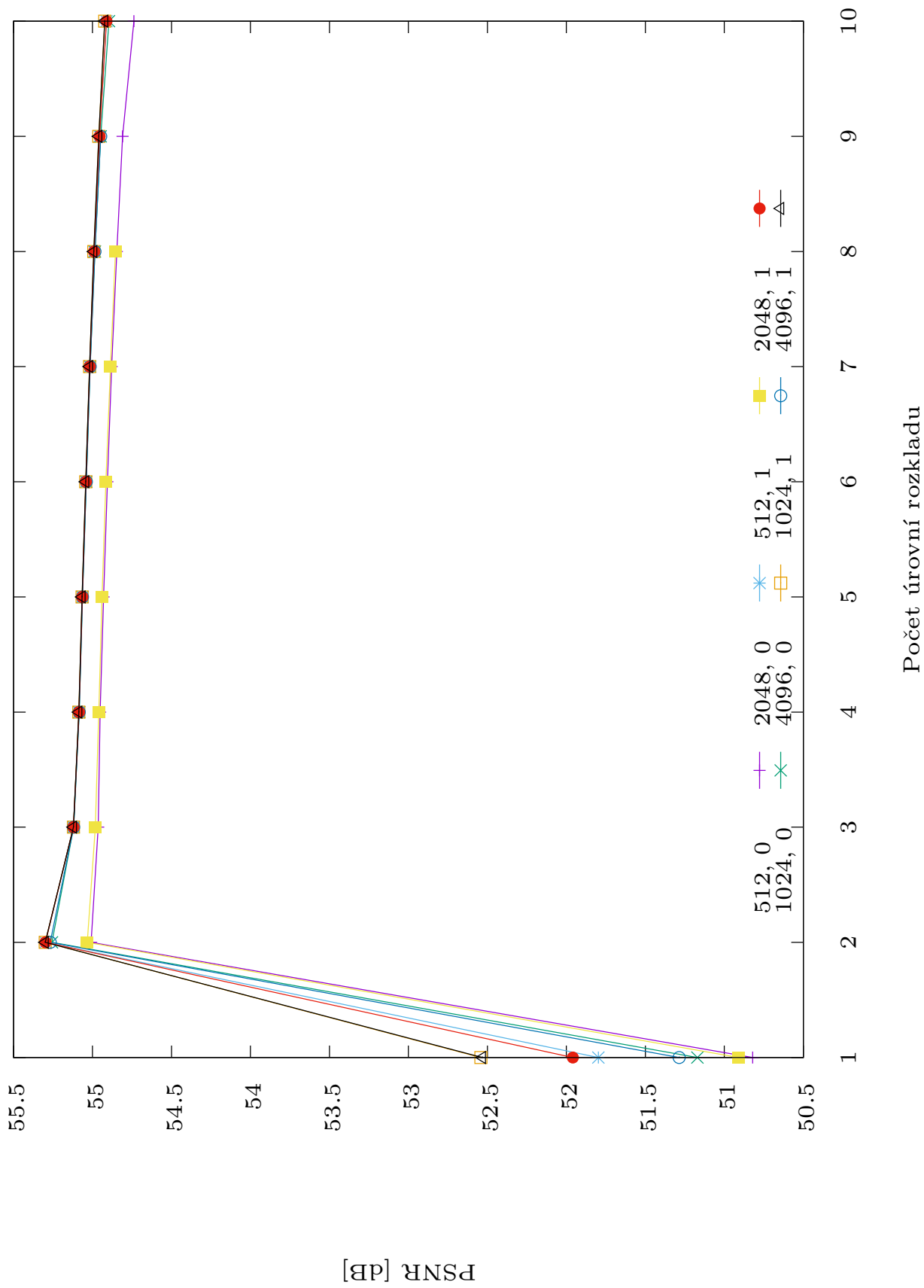
### 6.1.3 Shrnutí

Všeobecně lze říct, že jak komprese, tak dekomprese je s módem bypass rychlejší. U kvality potom nějak moc nezáleží ani na zvolené dlaždici, ani na módu. PSNR u tohoto datasetu se u knihovny Kakadu pohybovalo kolem 45 dB, u knihovny OpenJPEG kolem 45 56. Lze tedy doporučit spíše knihovnu OpenJPEG.

## 6.2 Dataset č.2 - Skenované časopisy

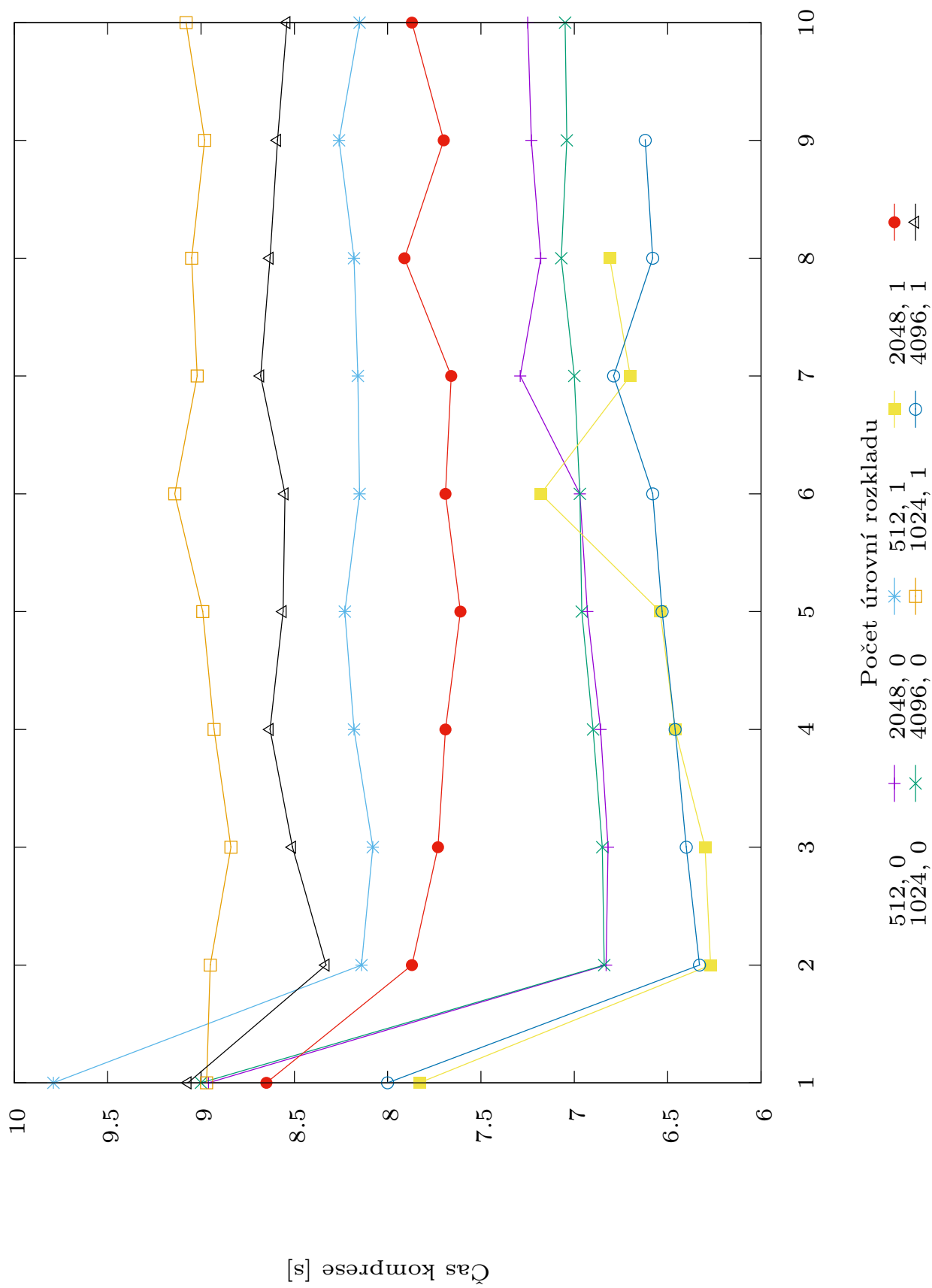
### 6.2.1 Knihovna OpenJPEG

Pro dataset skenované časopisy se v grafech 6.9, 6.10, 6.11 a 6.12 ukázalo, že nejvhodnější je použití módu bypass, jelikož je ve všech směrech rychlejší než běh bez módu. PSNR je ve všech případech shodné. Jako nejvhodnější lze tedy doporučit nastavení dlaždice  $1024 \times 1024$ , mód bypass, počet úrovní rozlišení 5.

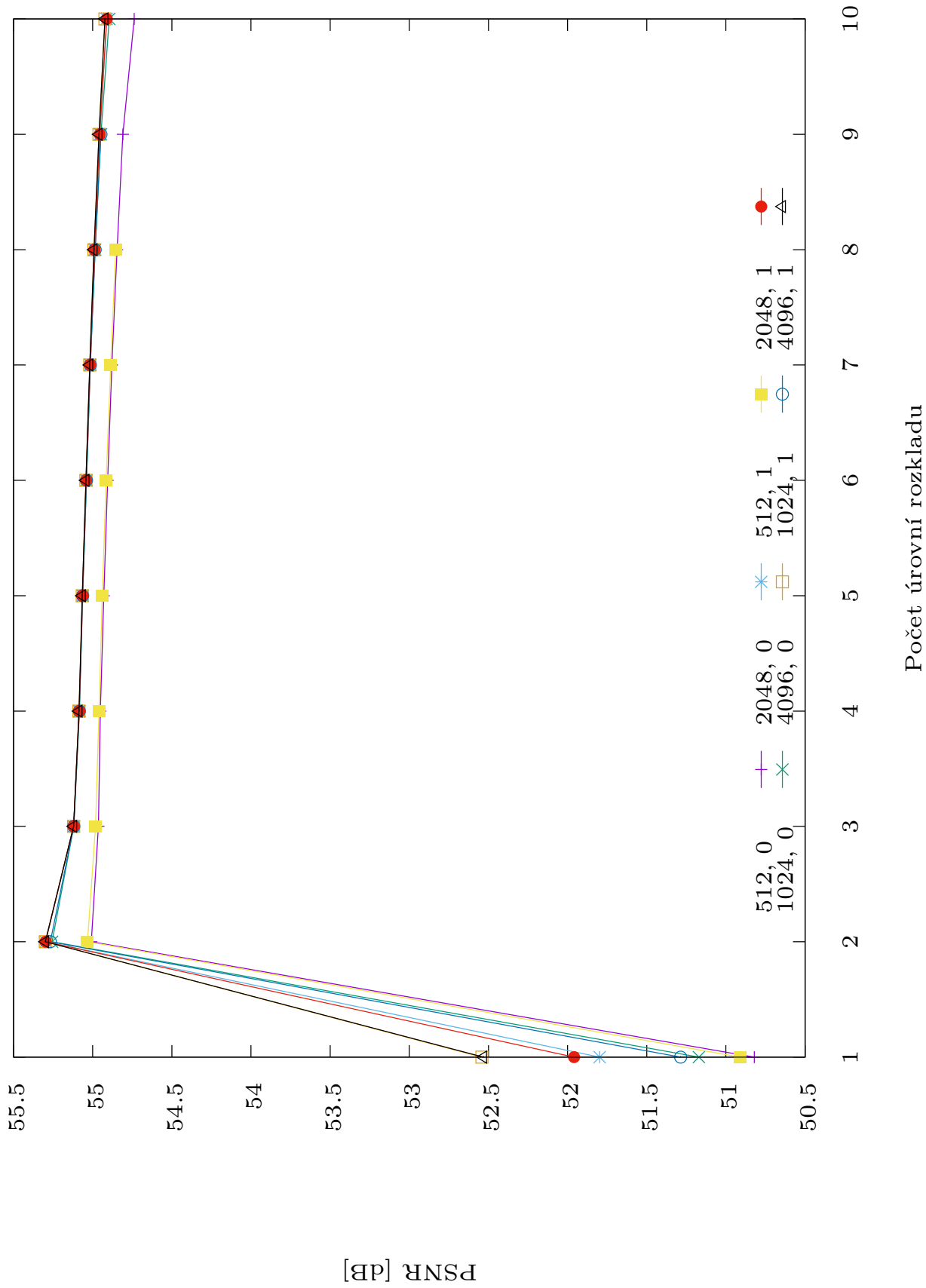


Obrázek 6.9: OpenJPEG - ztrátová komprese, skenované časopisy - průměrné PSNR komprimovaného obrazu

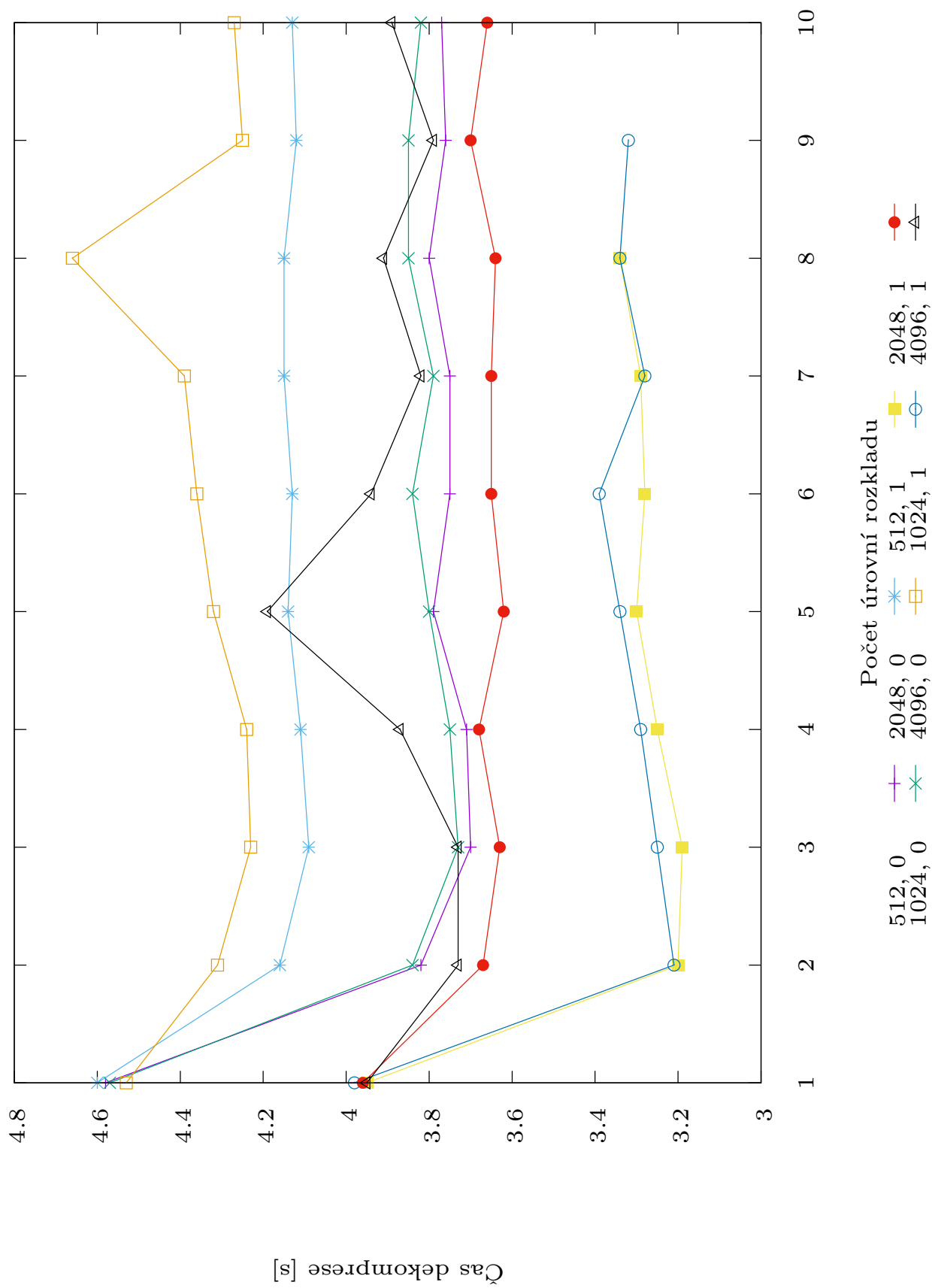




Obrázek 6.10: OpenJPEG - ztrátová komprese, skenované časopisy - průměrný čas komprese jednoho obrazu



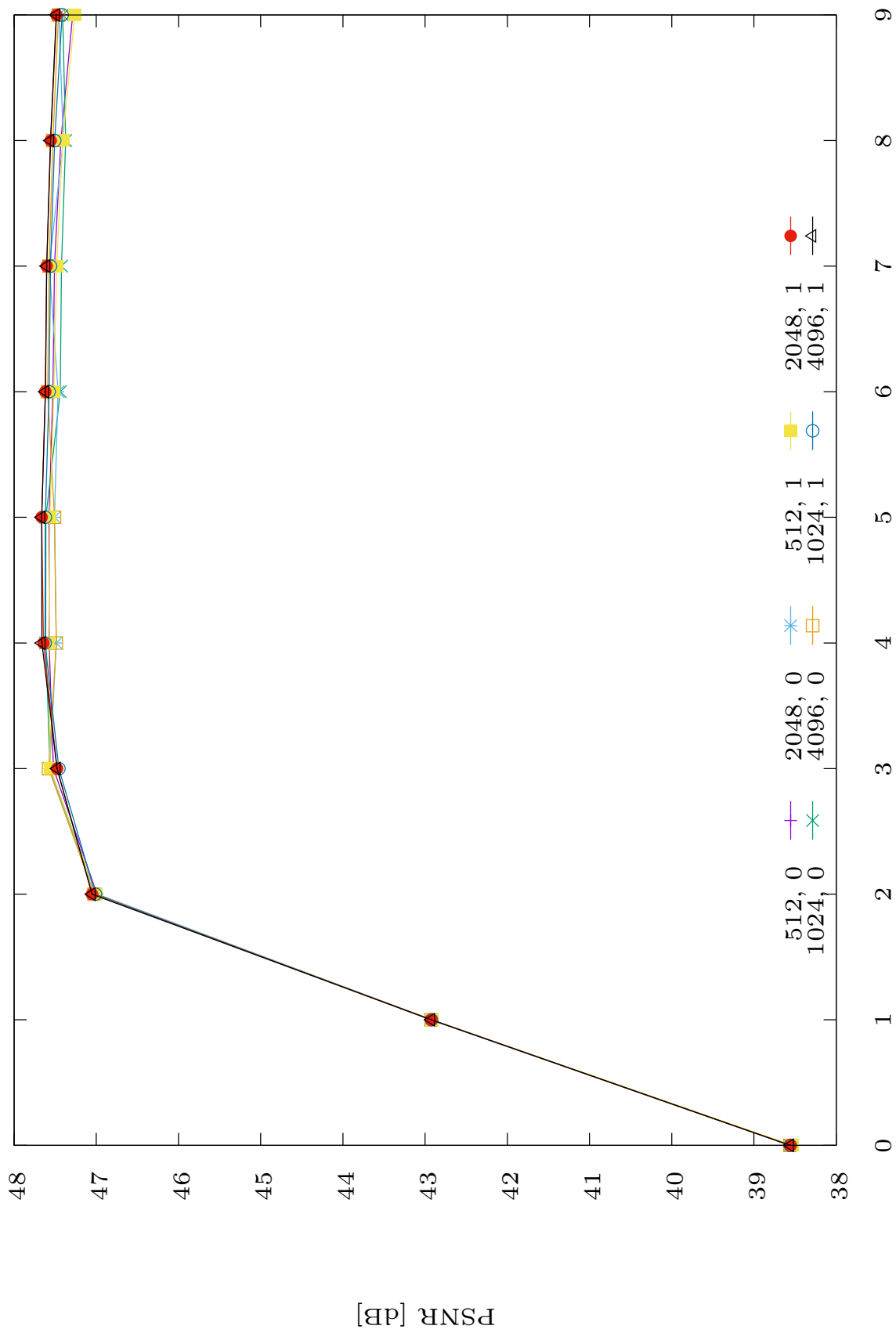
Obrázek 6.11: OpenJPEG - ztrátová komprese, skenované časopisy - průměrné PSNR dekomprimovaného obrazu



Obrázek 6.12: OpenJPEG - ztrátová komprese, skenované časopisy - průměrný čas dekomprese jednoho obrazu

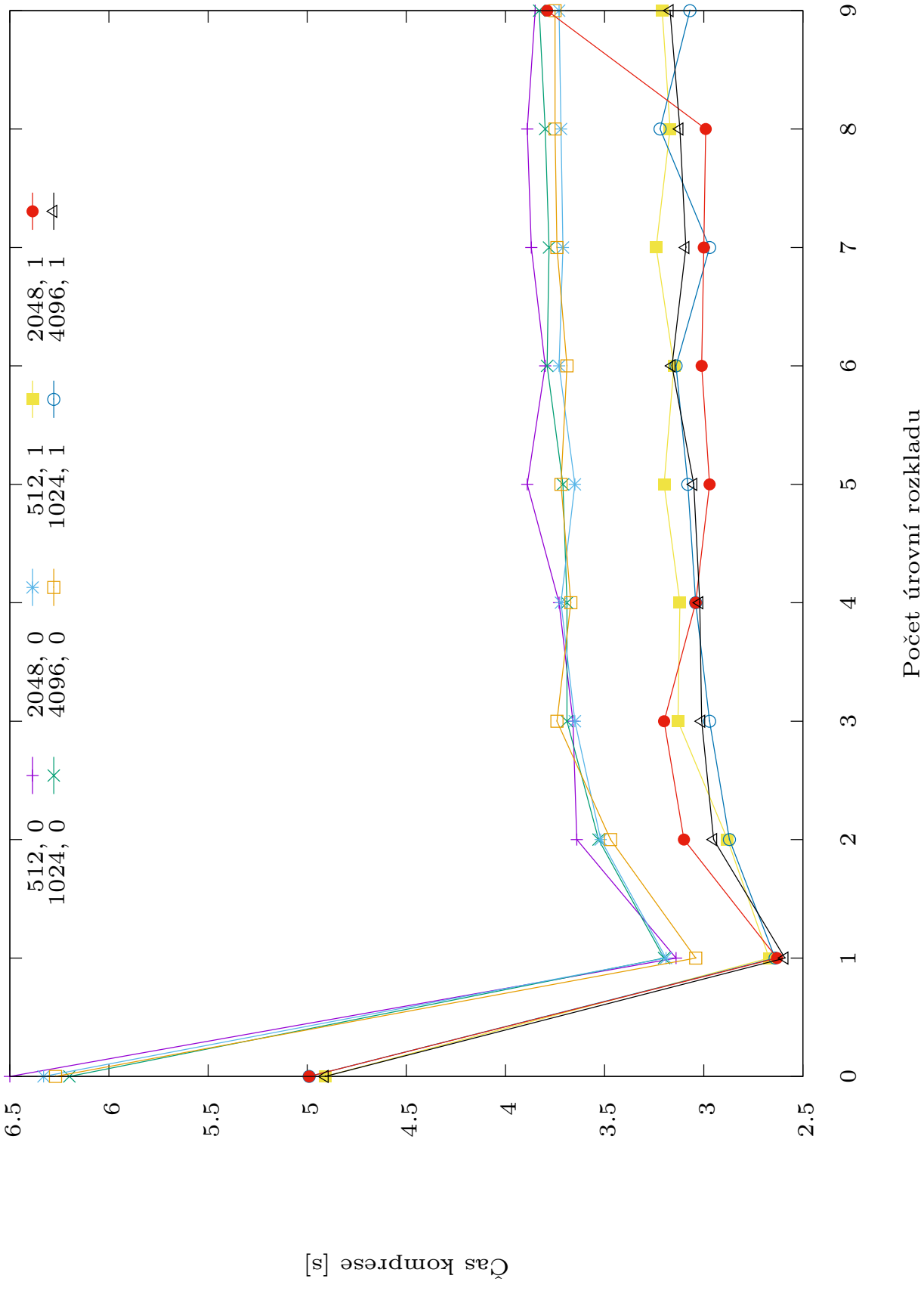
### 6.2.2 Knihovna Kakadu

Pro dataset fotografií se v grafech 6.13, 6.14, 6.15 a 6.16 ukázalo, že nejvhodnější je použití módu bypass, jelikož je ve všech směrech rychlejší než běh bez módu. PSNR je ve všech případech shodné. Jako nejvhodnější lze jednoznačně doporučit nastavení dlaždice  $4096 \times 4096$ , mód bypass, počet úrovní rozlišení 5

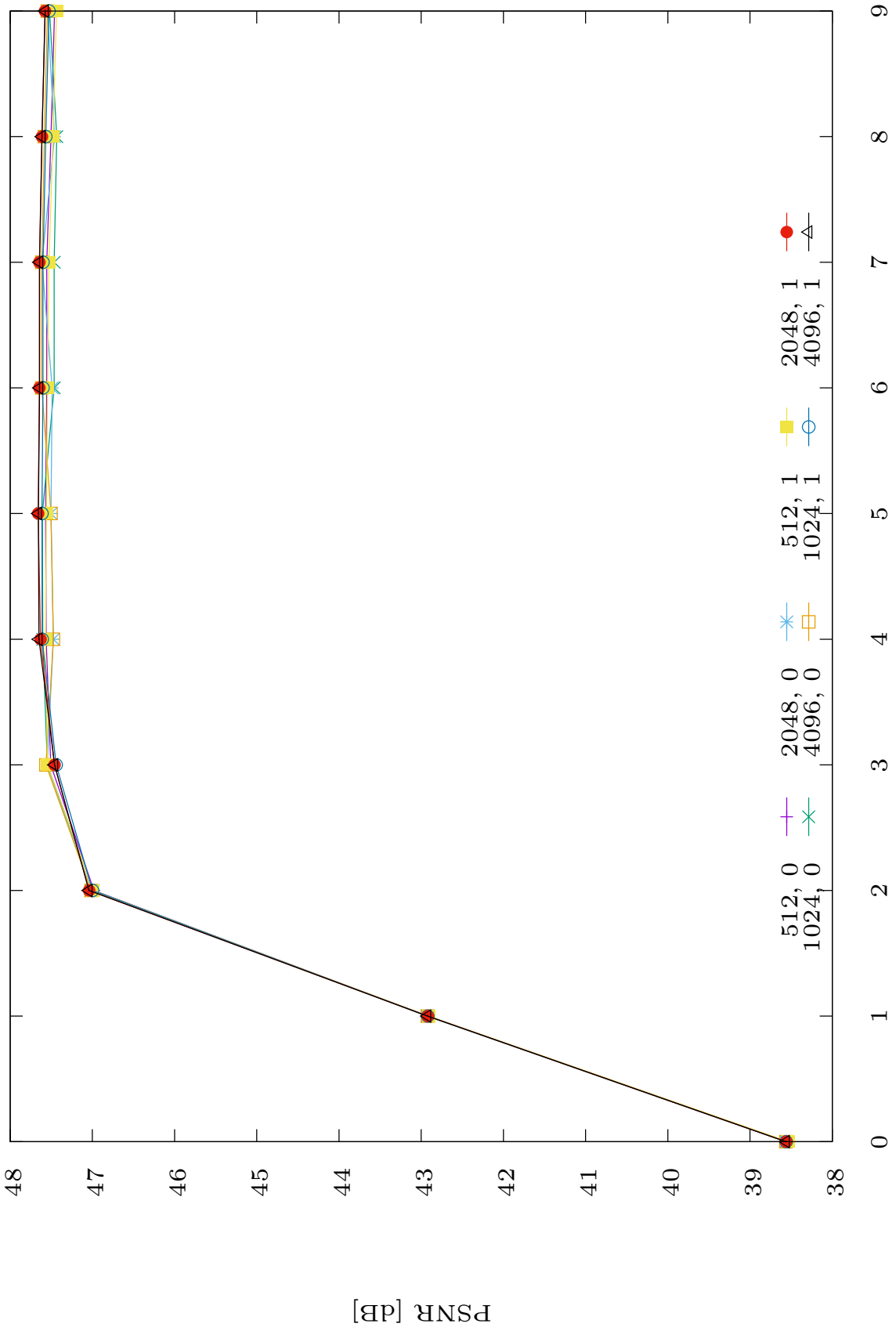


Počet úrovní rozkladu

Obrázek 6.13: Kakadu - ztrátová komprese, skenované časopisy - průměrné PSNR komprimovaného obrazu

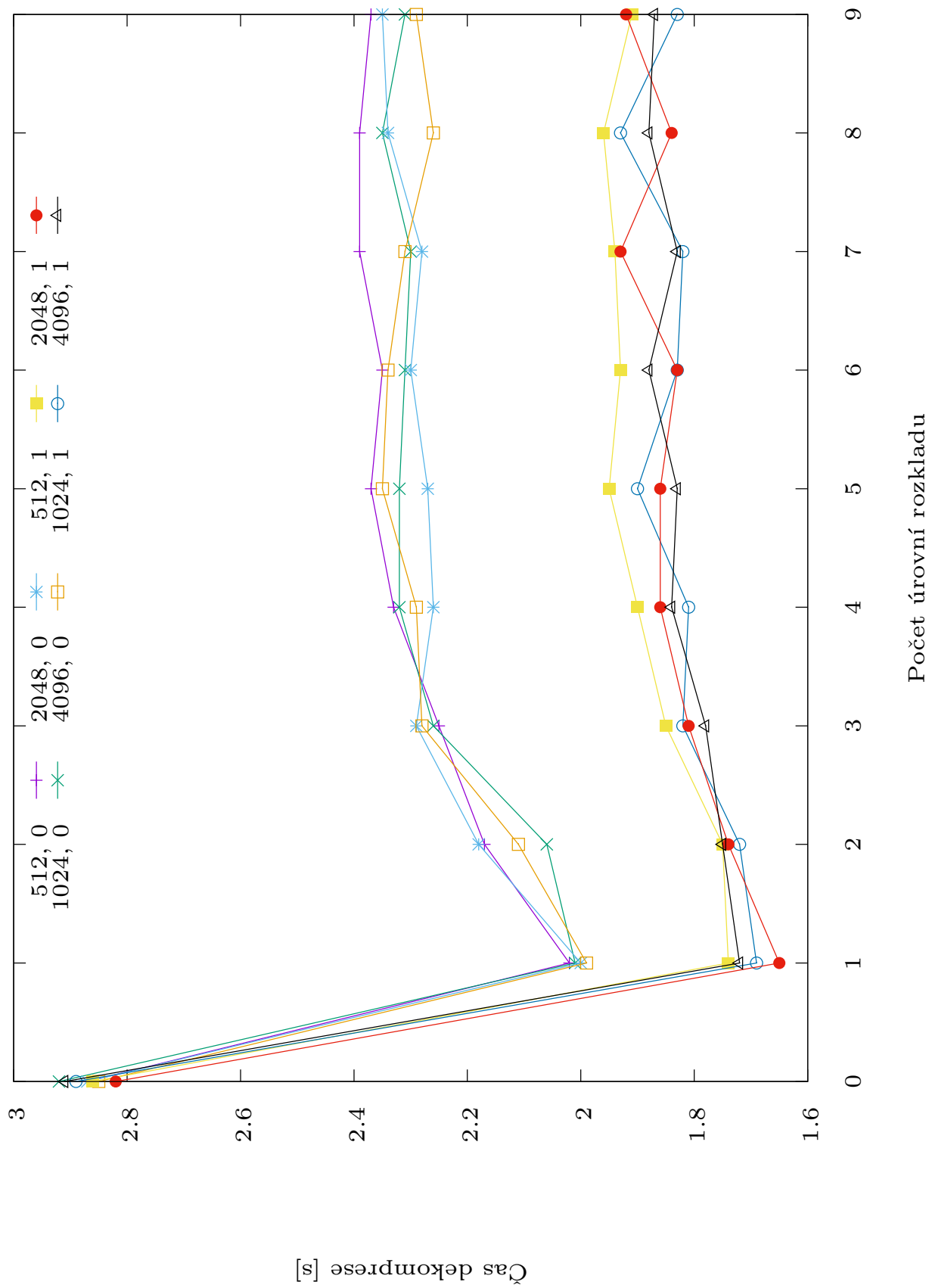


Obrázek 6.14: Kakadu - ztrátová komprese, skenované časopisy - průměrný čas komprese jednoho obrazu



Počet úrovní rozkladu

Obrázek 6.15: Kakadu - ztrátová komprese, skenované časopisy - průměrné PSNR dekomprimovaného obrazu



Obrázek 6.16: Kakadu - ztrátová komprese, skenované časopisy - průměrný čas dekomprese jednoho obrazu



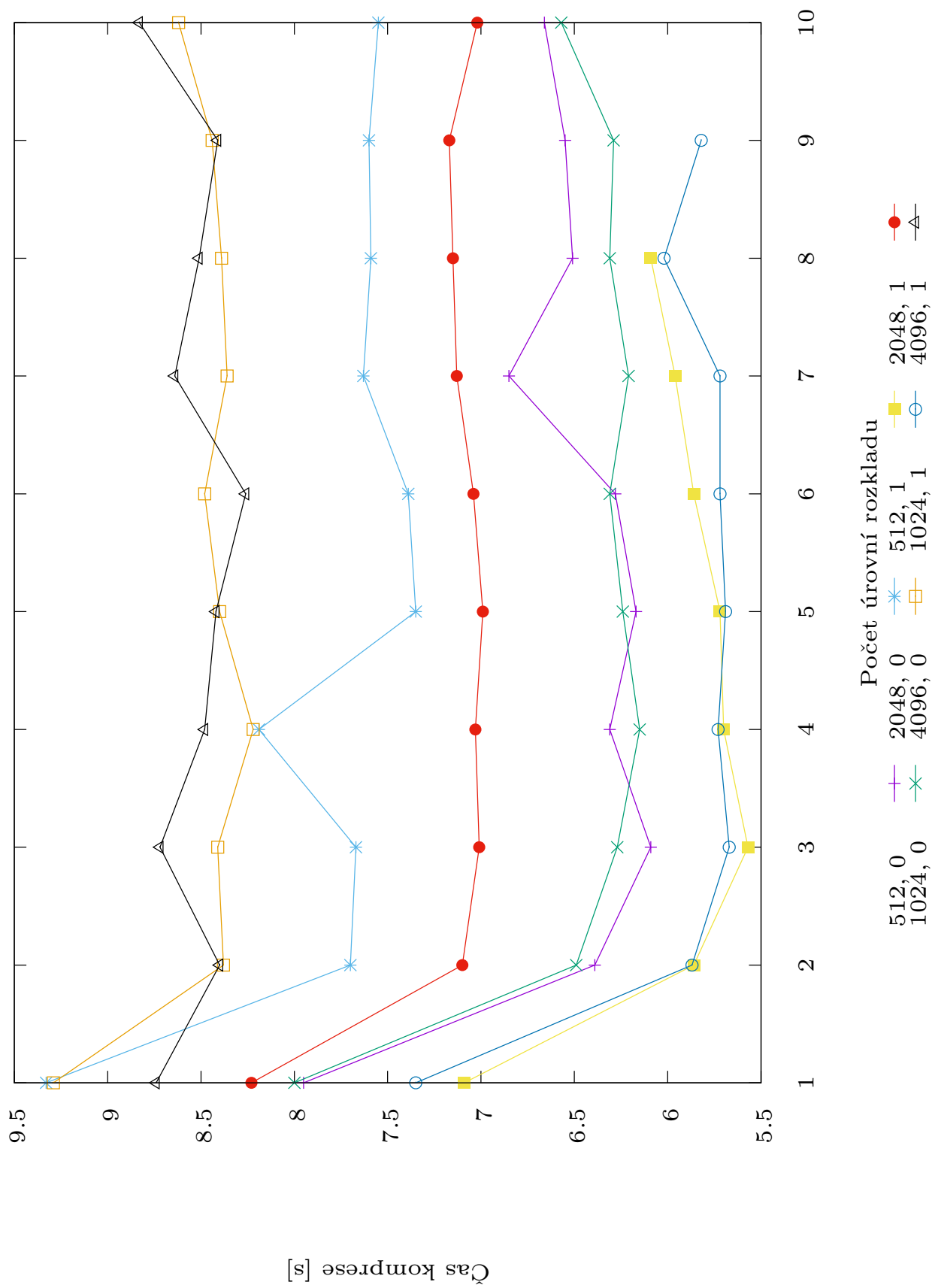
### 6.2.3 Shrnutí

Všeobecně lze říct, že jak komprese, tak dekomprese je s módem bypass rychlejší. U kvality potom nějak moc nezáleží ani na zvolené dlaždici, ani na módu, nejhůř se u obou knihoven jeví kombinace dlaždice  $512 \times 512$  a módu bypass. PSNR u tohoto datasetu se u knihovny Kakadu pohybovalo kolem 48 dB, u knihovny OpenJPEG kolem 55 dB. Lze tedy doporučit spíše knihovnu Kakadu.

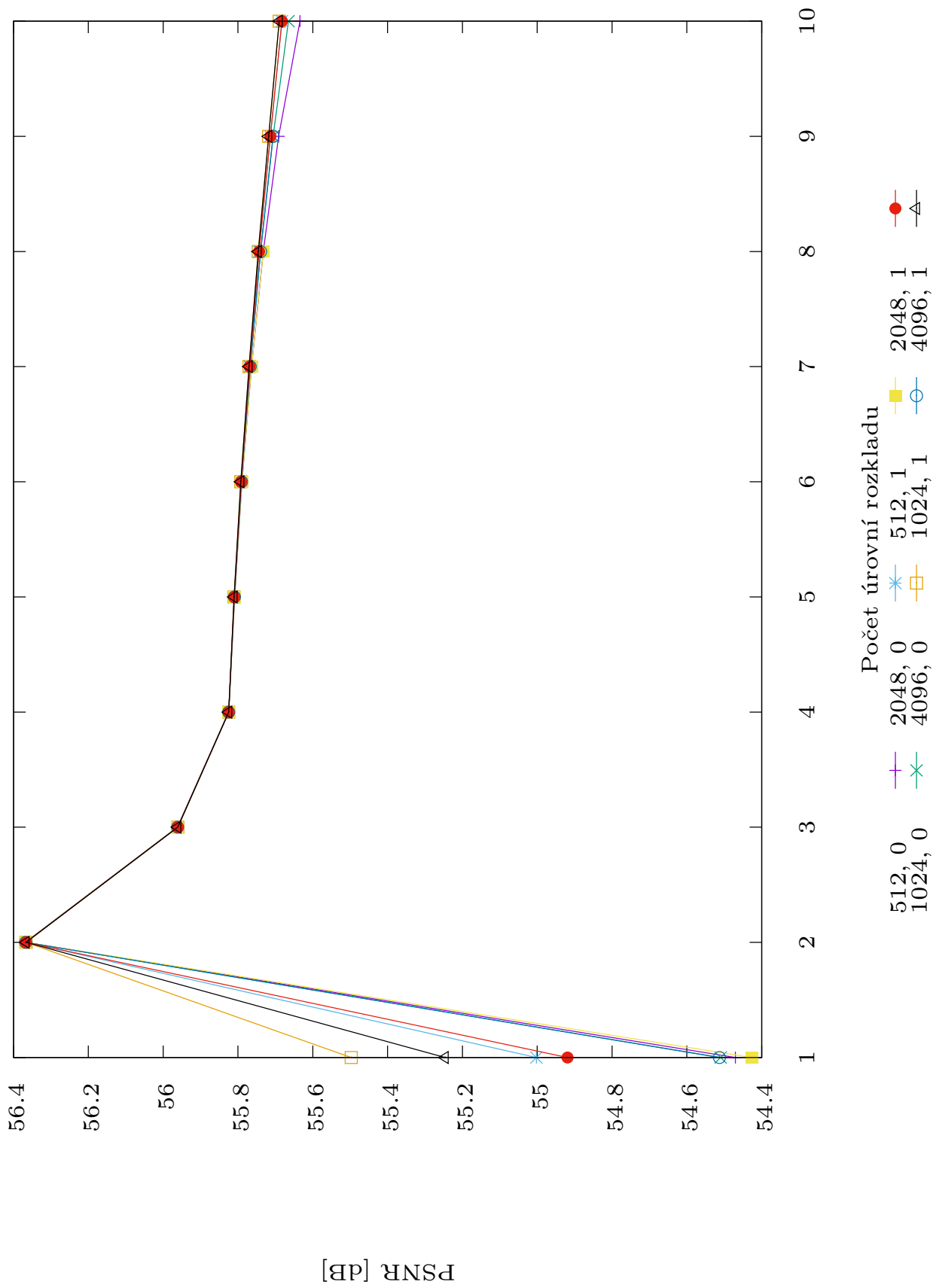
## 6.3 Dataset č.3 - Skenované mapy

### 6.3.1 Knihovna OpenJPEG

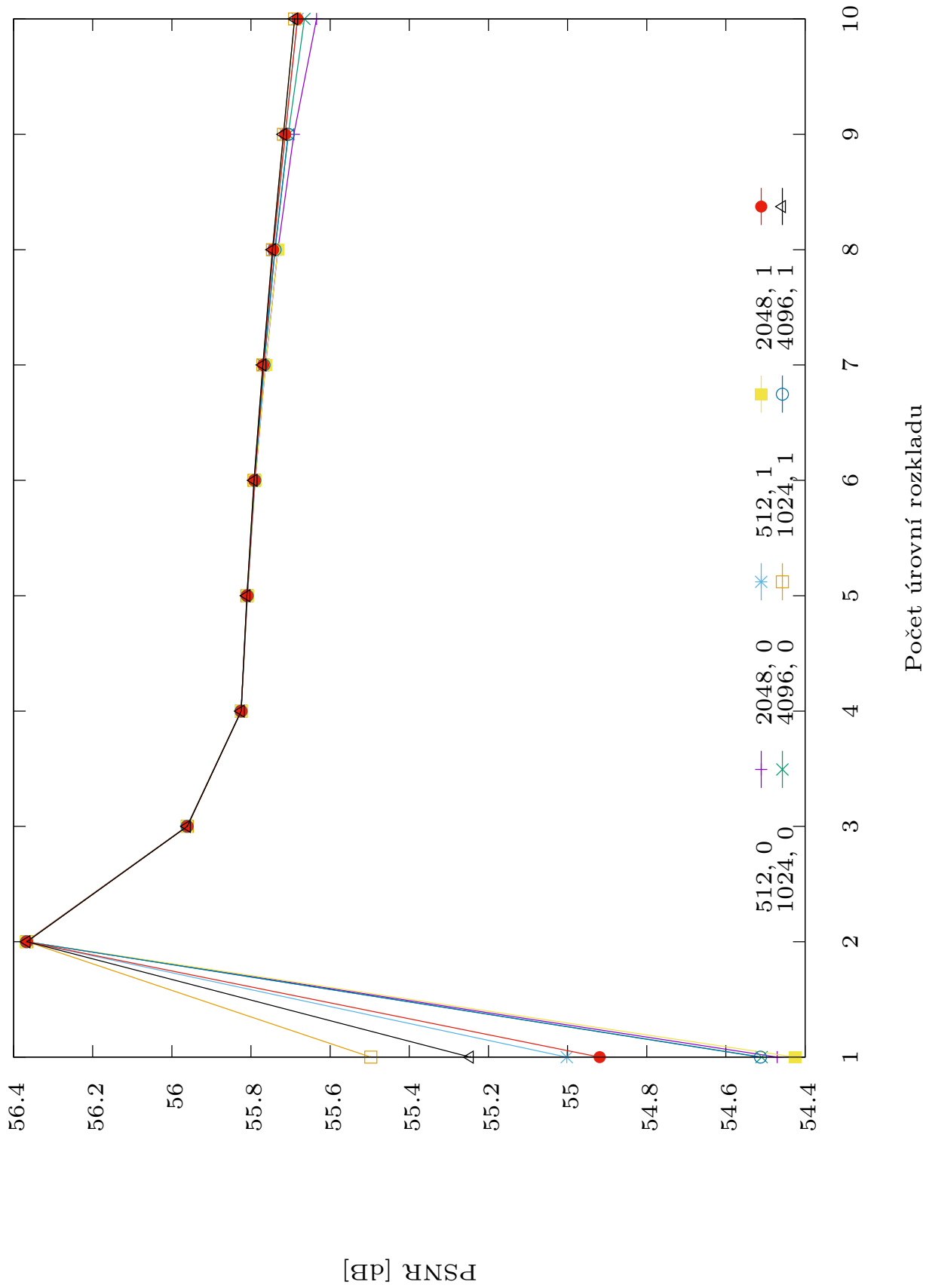
Pro dataset fotografií se v grafech 6.17, 6.18, 6.19 a 6.20 ukázalo, že nejvhodnější je použití módu bypass, jelikož je ve všech směrech rychlejší než běh bez módu. PSNR je ve všech případech shodné. Jako nejvhodnější lze tedy jednoznačně doporučit nastavení dlaždice  $512 \times 512$ , mód bypass, počet úrovní rozlišení 5



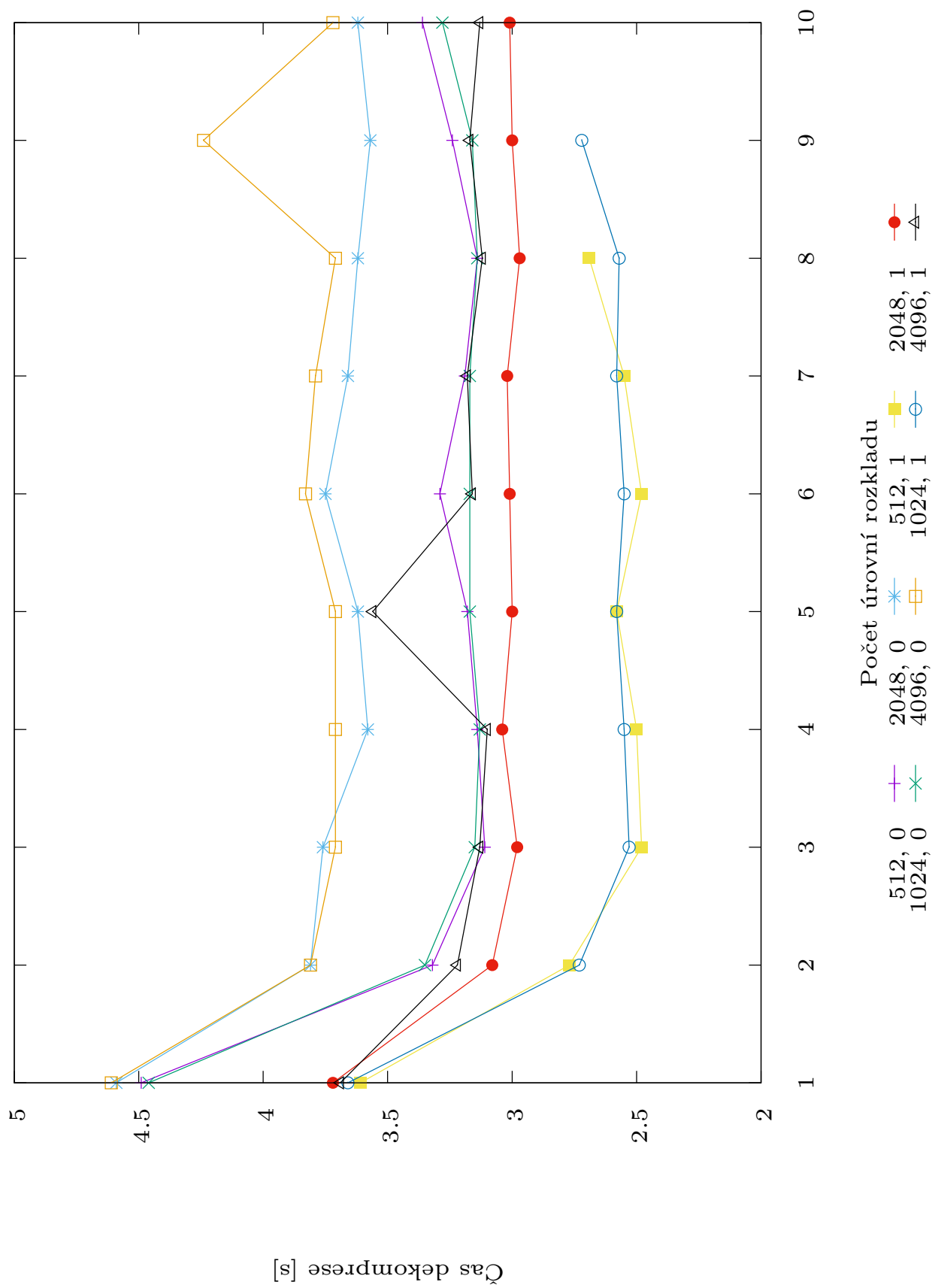
Obrázek 6.17: OpenJPEG - ztrátová komprese, skenované mapy - průměrný čas komprese jednoho obrazu



Obrázek 6.18: OpenJPEG - ztrátová komprese, skenované mapy - průměrné PSNR komprimovaného obrazu



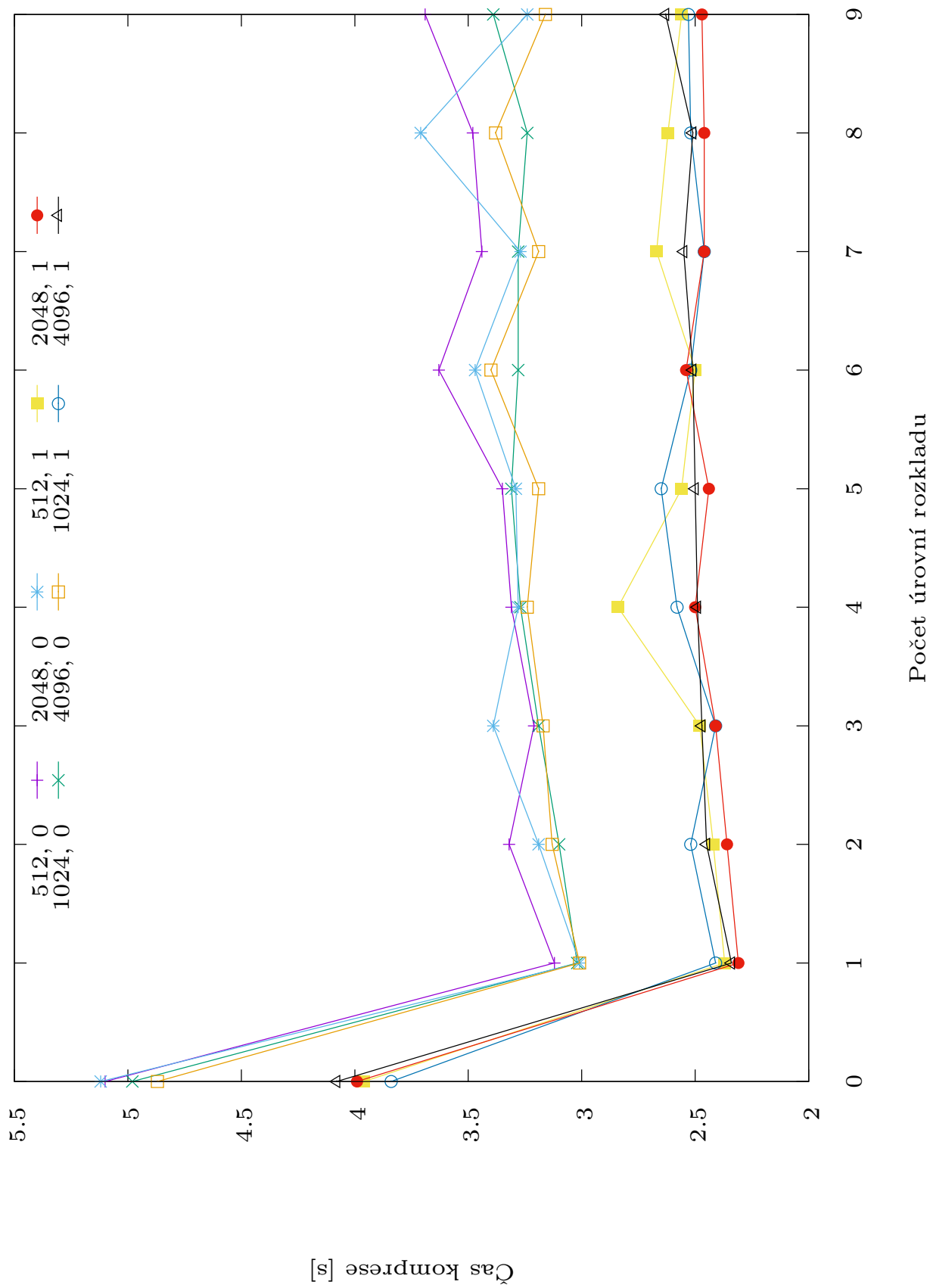
Obrázek 6.19: OpenJPEG - ztrátová komprese, skenované mapy - průměrné PSNR dekomprimovaného obrazu



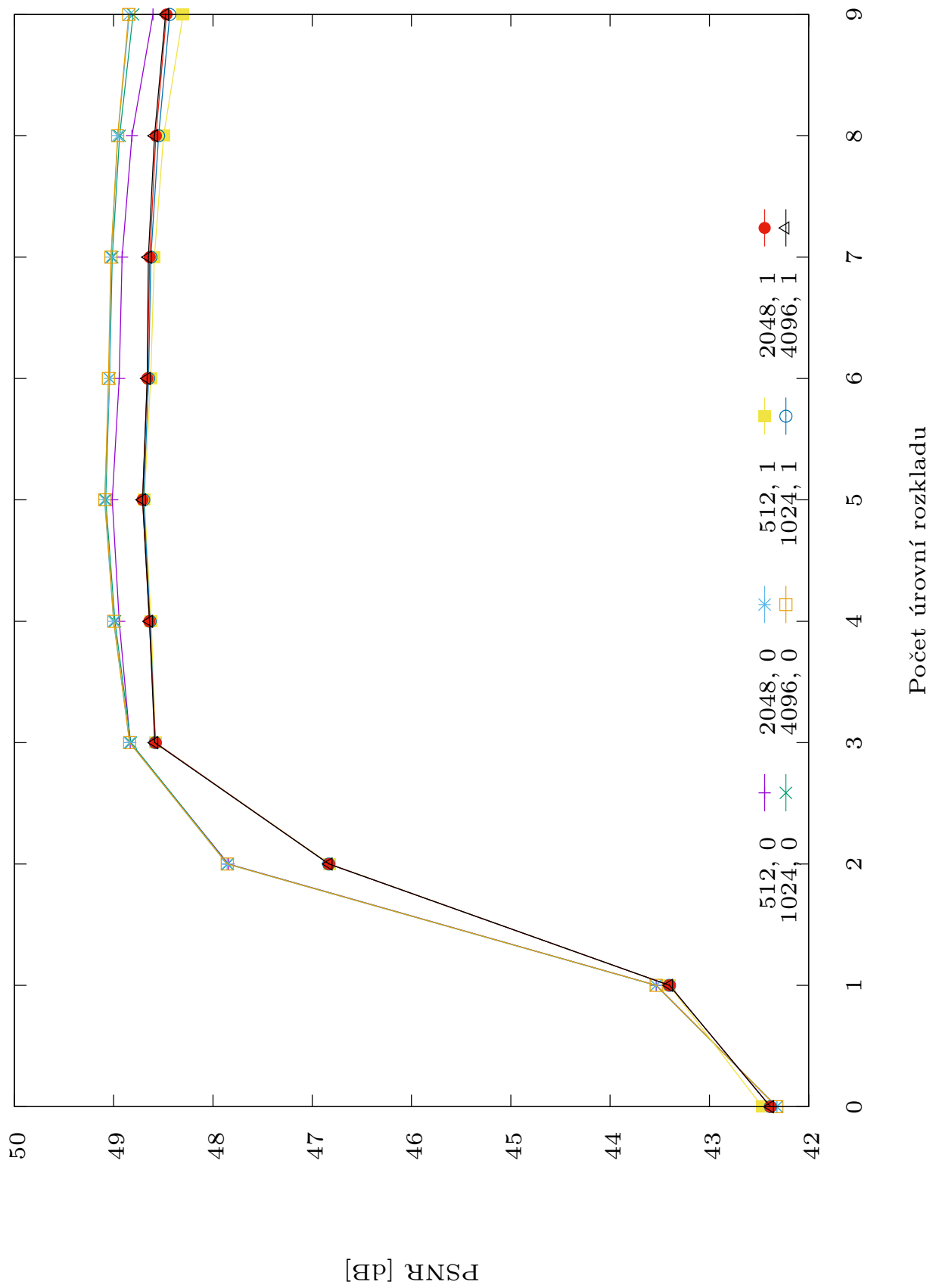
Obrázek 6.20: OpenJPEG - ztrátová komprese, skenované mapy - průměrný čas dekomprese jednoho obrazu

### 6.3.2 Knihovna Kakadu

Pro dataset fotografií se v grafech 6.21, 6.22, 6.23 a 6.24 ukázalo, že nevhodnější je použití módu bypass, jelikož je ve všech směrech rychlejší než běh bez módu. Dále se ukázalo, že velikost dlaždice nemá na výslednou kvalitu vliv.

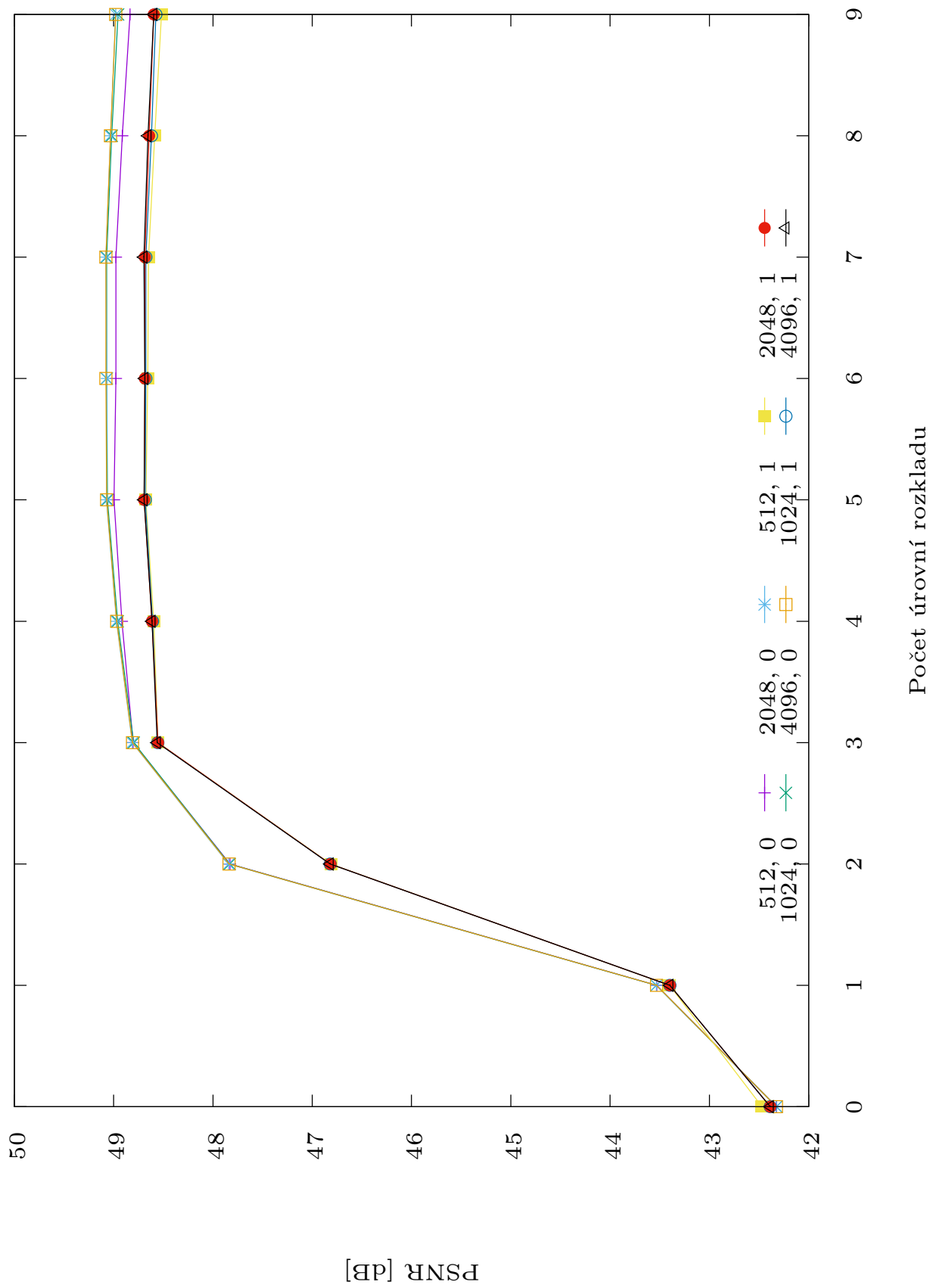


Obrázek 6.21: Kakadu - ztrátová komprese, skenované mapy - průměrný čas komprese jednoho obrázu

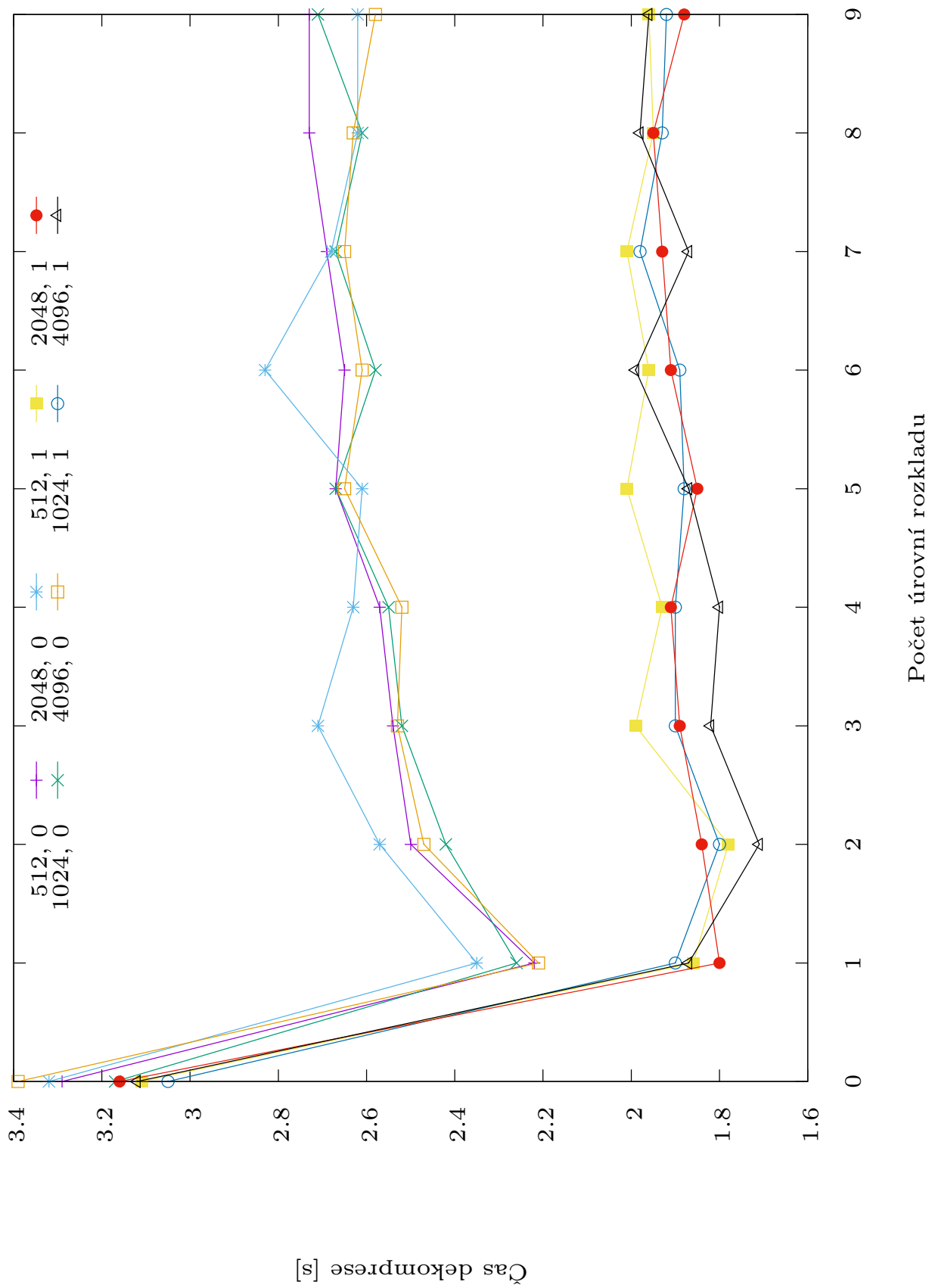


Obrázek 6.22: Kakadu - ztrátová komprese, skenované mapy - průměrné PSNR komprimovaného obrazu





Obrázek 6.23: Kakadu - ztrátová komprese, skenované mapy - průměrné PSNR dekomprimovaného obrazu



Obrázek 6.24: Kakadu - ztrátová komprese, skenované mapy - průměrný čas dekomprese jednoho obrazu

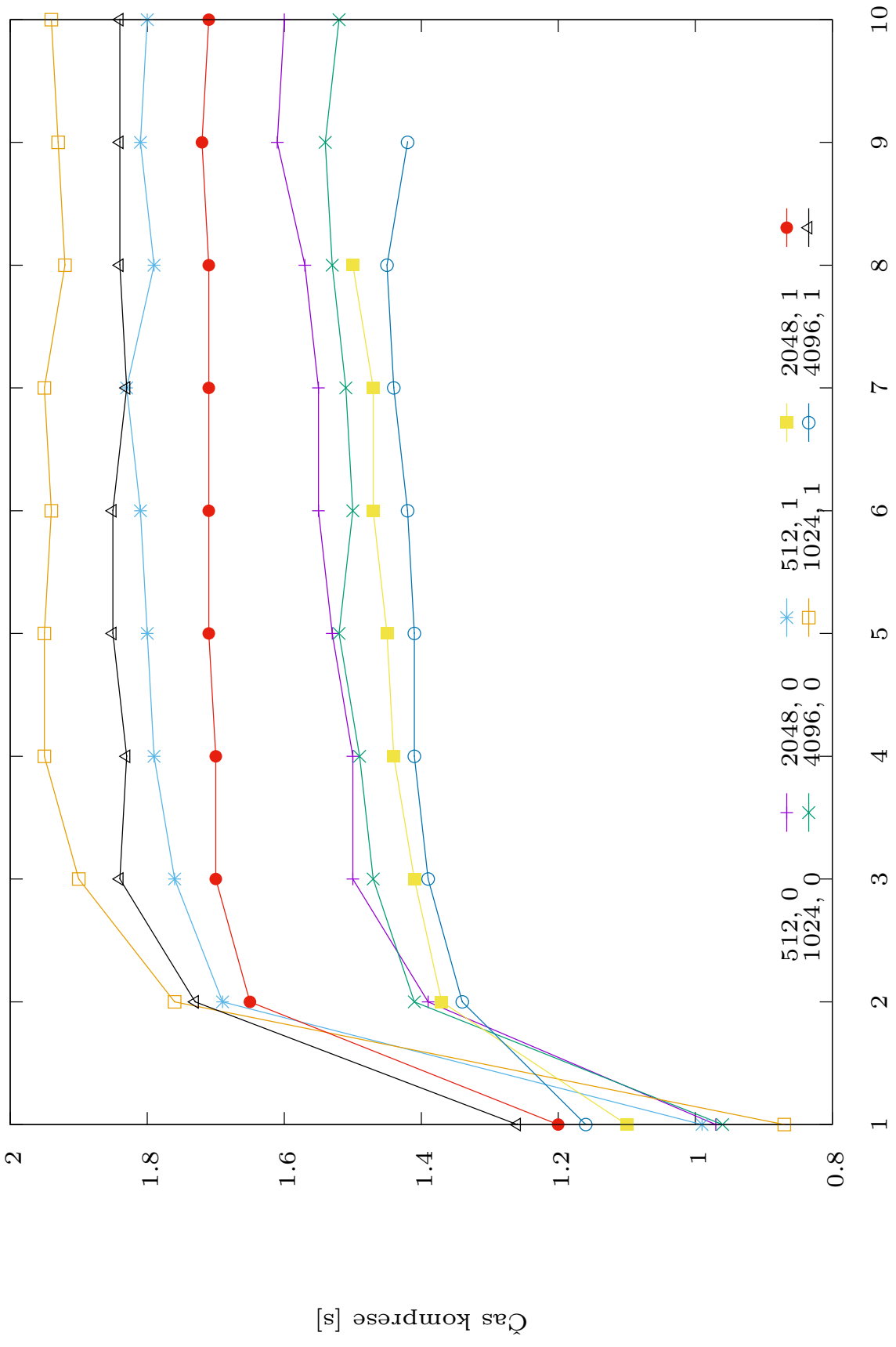
### 6.3.3 Shrnutí

Všeobecně lze říct, že jak komprese, tak dekomprese je s módem bypass rychlejší. U kvality potom záleží na zvolené dlaždici, ani ne tak na módu, nejlépe vycházejí dlaždice  $512 \times 512$  a  $1024 \times 1024$  s módem bypass. PSNR u tohoto datasetu se u knihovny Kakadu pohybovalo kolem 50 dB, u knihovny OpenJPEG kolem 56 dB. Lze tedy doporučit spíše knihovnu Kakadu.

## 6.4 Dataset č.4 - Skenované knihy

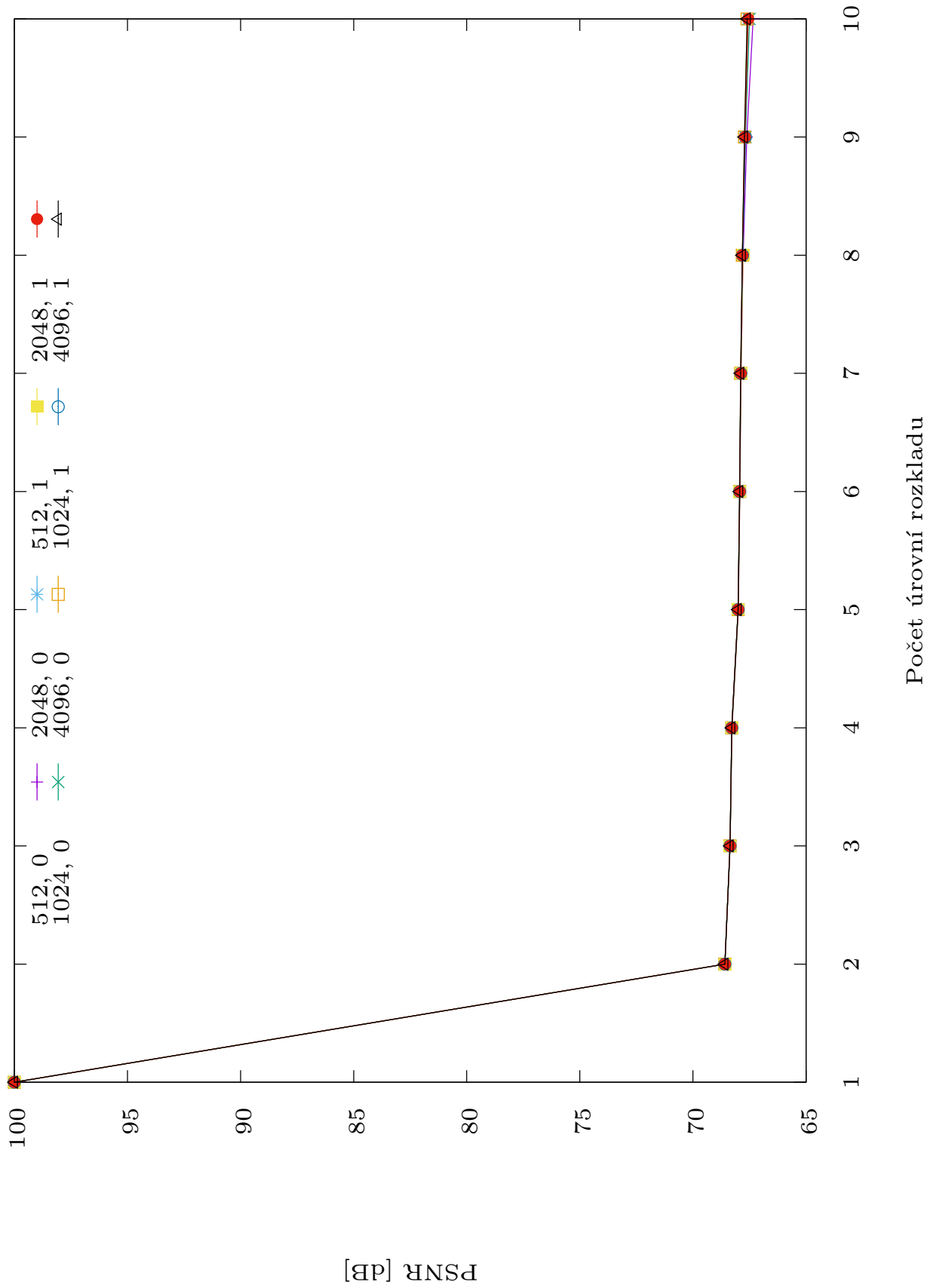
Pro dataset fotografií se v grafech 6.25, 6.26, 6.27 a 6.28 ukázalo, že nejvhodnější je použití módu bypass, jelikož je ve všech směrech rychlejší než běh bez módu. PSNR je ve všech případech shodné. Jako nejvhodnější lze tedy jednoznačně doporučit nastavení dlaždice  $1024 \times 1024$ , mód bypass, počet úrovní rozlišení 5.

### 6.4.1 Knihovna OpenJPEG

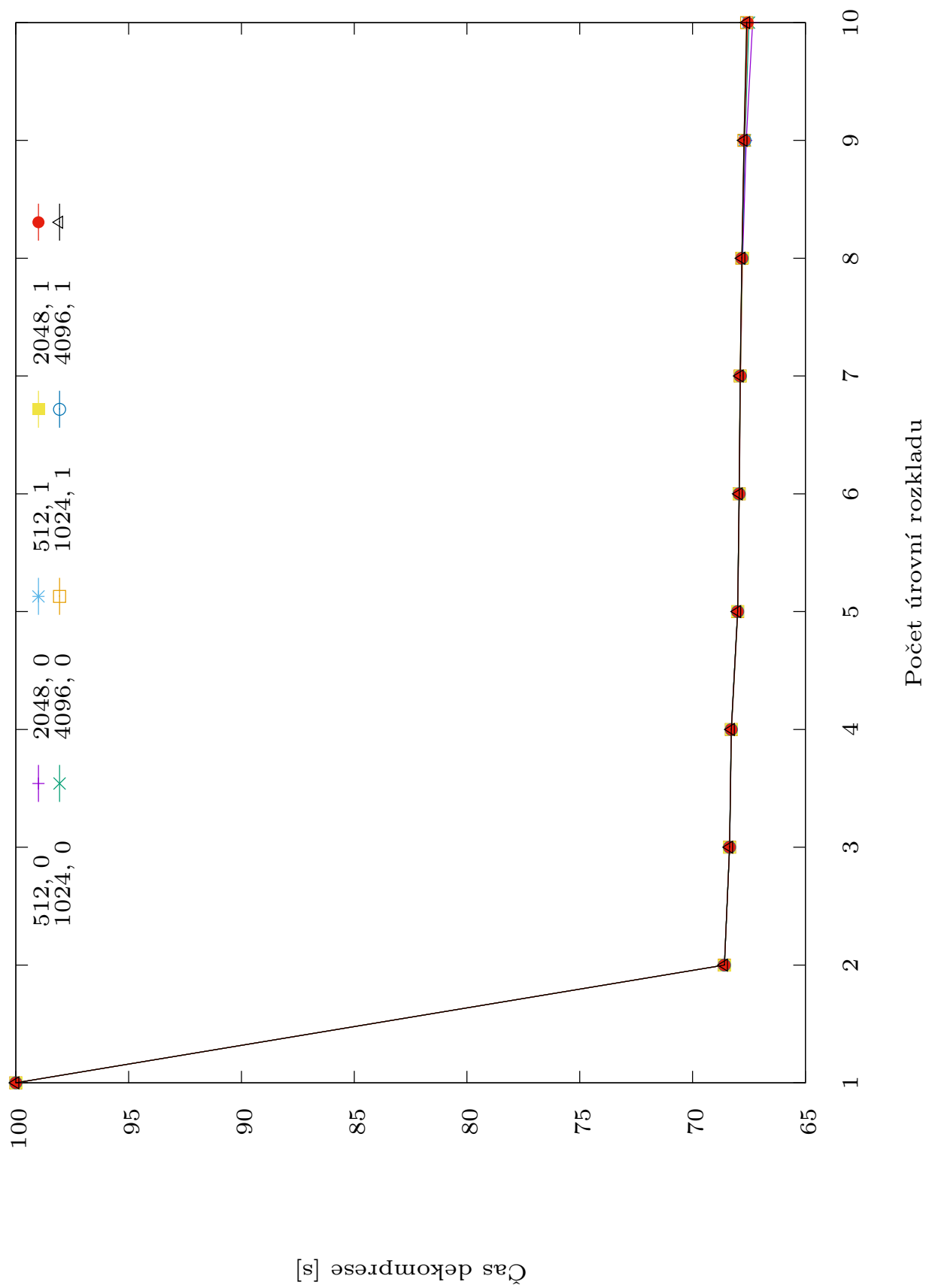


Počet úrovní rozkladu

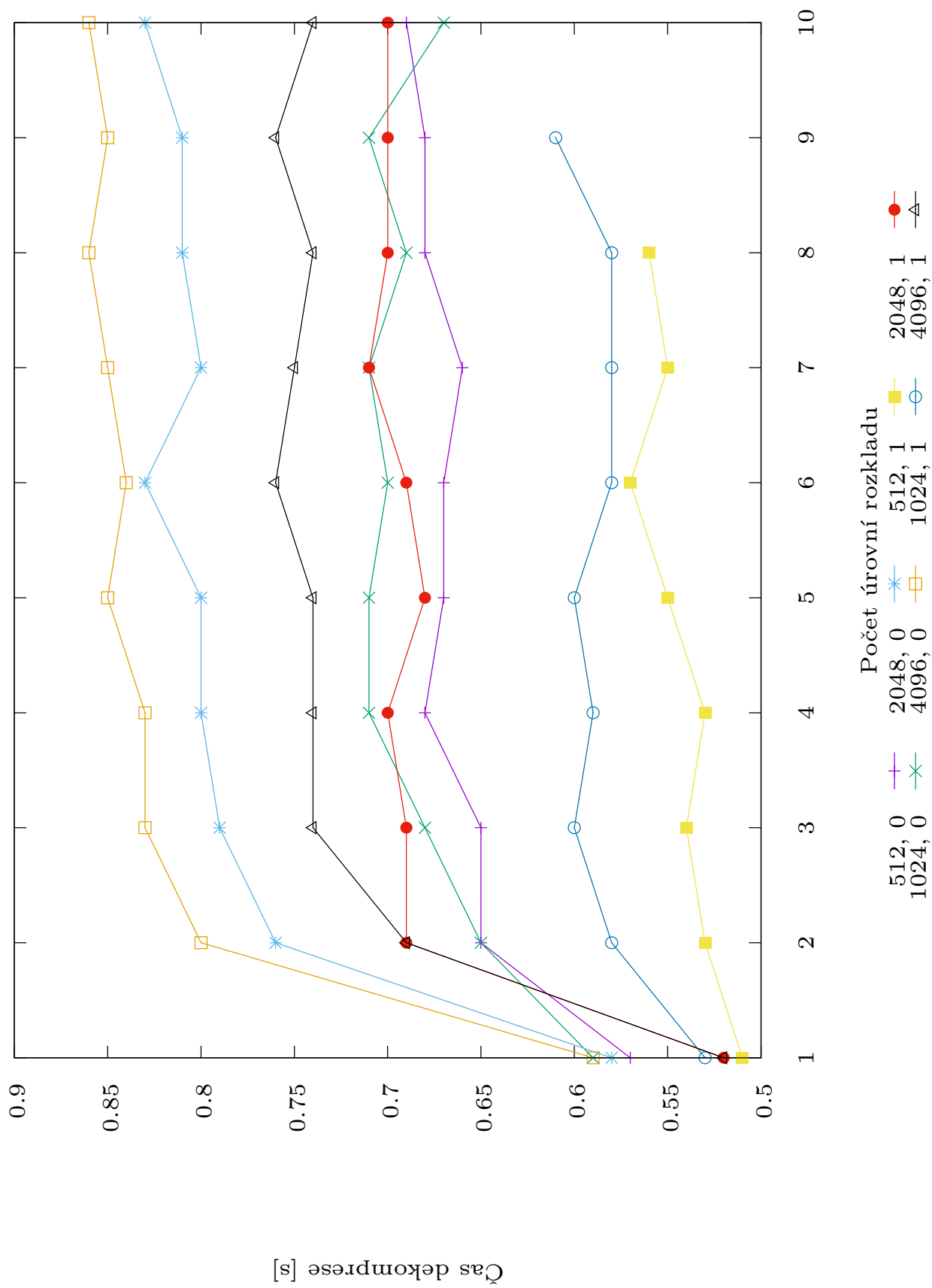
Obrázek 6.25: OpenJPEG - ztrátová komprese, skenované knihy - průměrný čas komprese jednoho obrazu



Obrázek 6.26: OpenJPEG - ztrátová komprese, skenované knihy - průměrné PSNR komprimovaného obrazu



Obrázek 6.27: OpenJPEG - ztrátová komprese, skenované knihy - průměrné PSNR dekomprimovaného obrazu

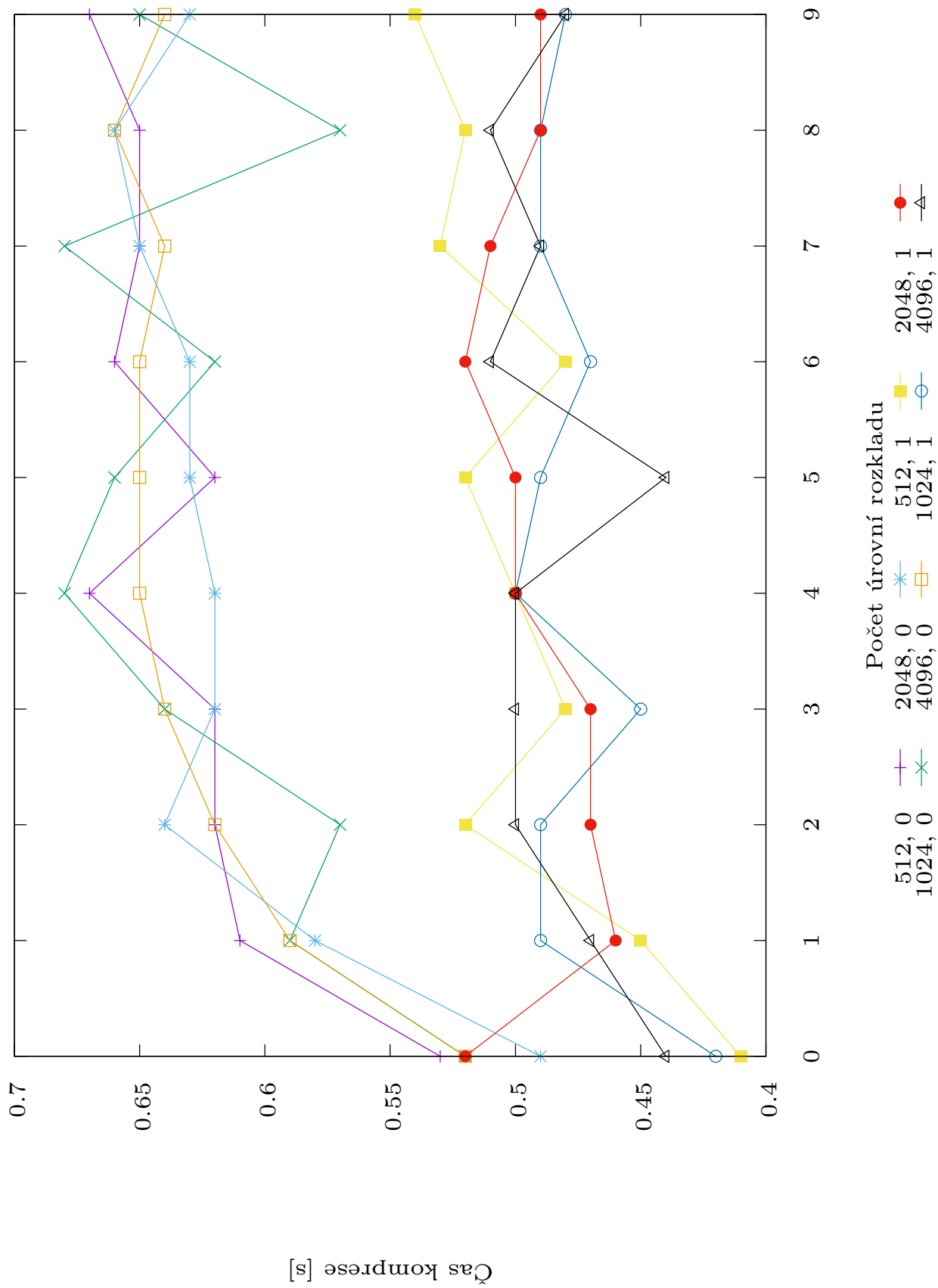


Obrázek 6.28: OpenJPEG - ztrátová komprese, skenované knihy - průměrný čas dekomprese jednoho obrazu

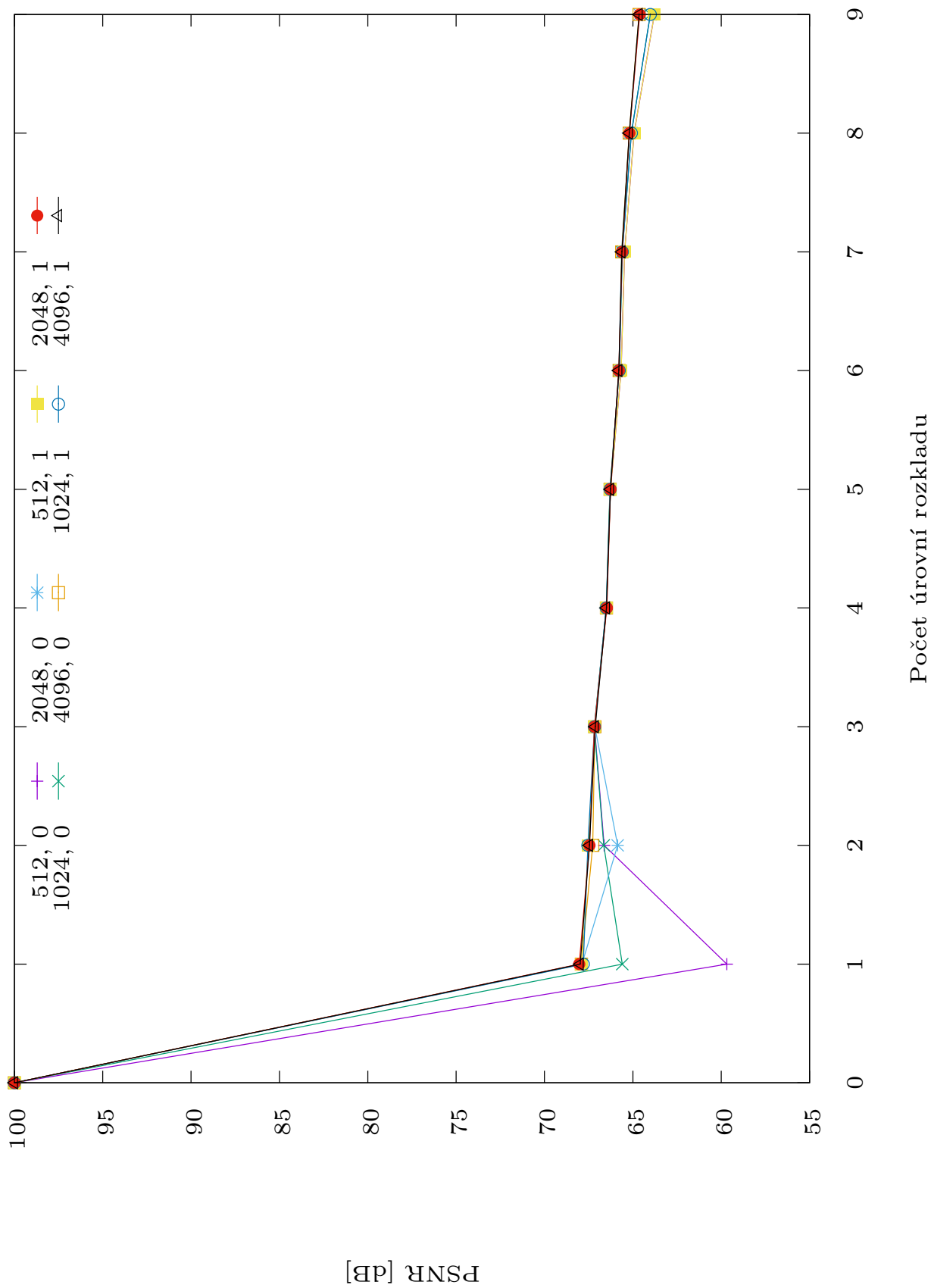
### 6.4.2 Knihovna Kakadu

Pro dataset fotografií se v grafech 6.29, 6.30, 6.31 a 6.32 ukázalo, že nejvhodnější je použití módu bypass, jelikož je ve všech směrech rychlejší než běh bez módu. Dále se ukázalo, že je na nízkých úrovních rozkladu jak kvalitativně, tak rychlostně horší použití knihovny bez módu. U vyšších úrovní rozkladu, ale nemá velikost dlaždice na výslednou kvalitu vliv.

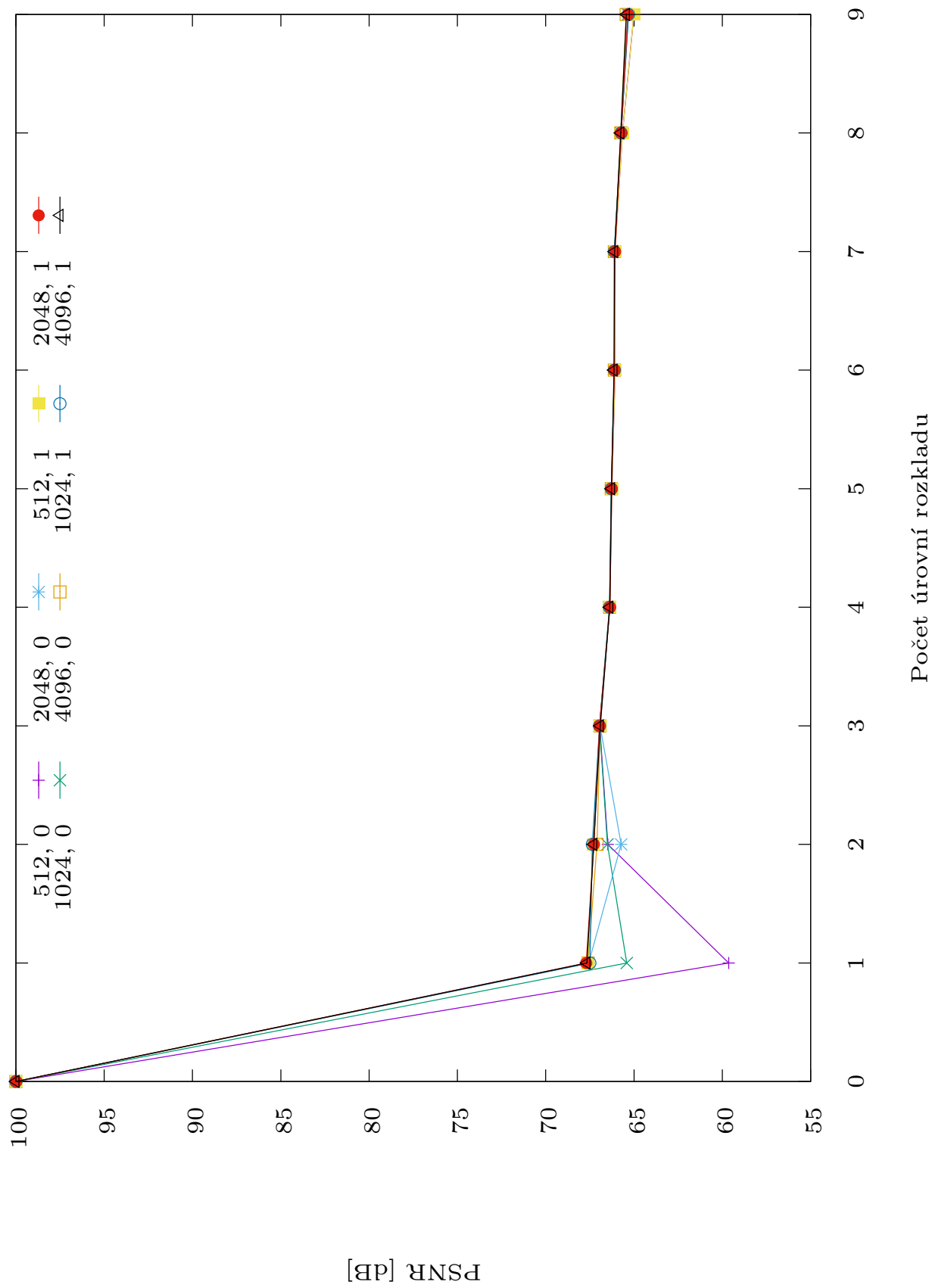




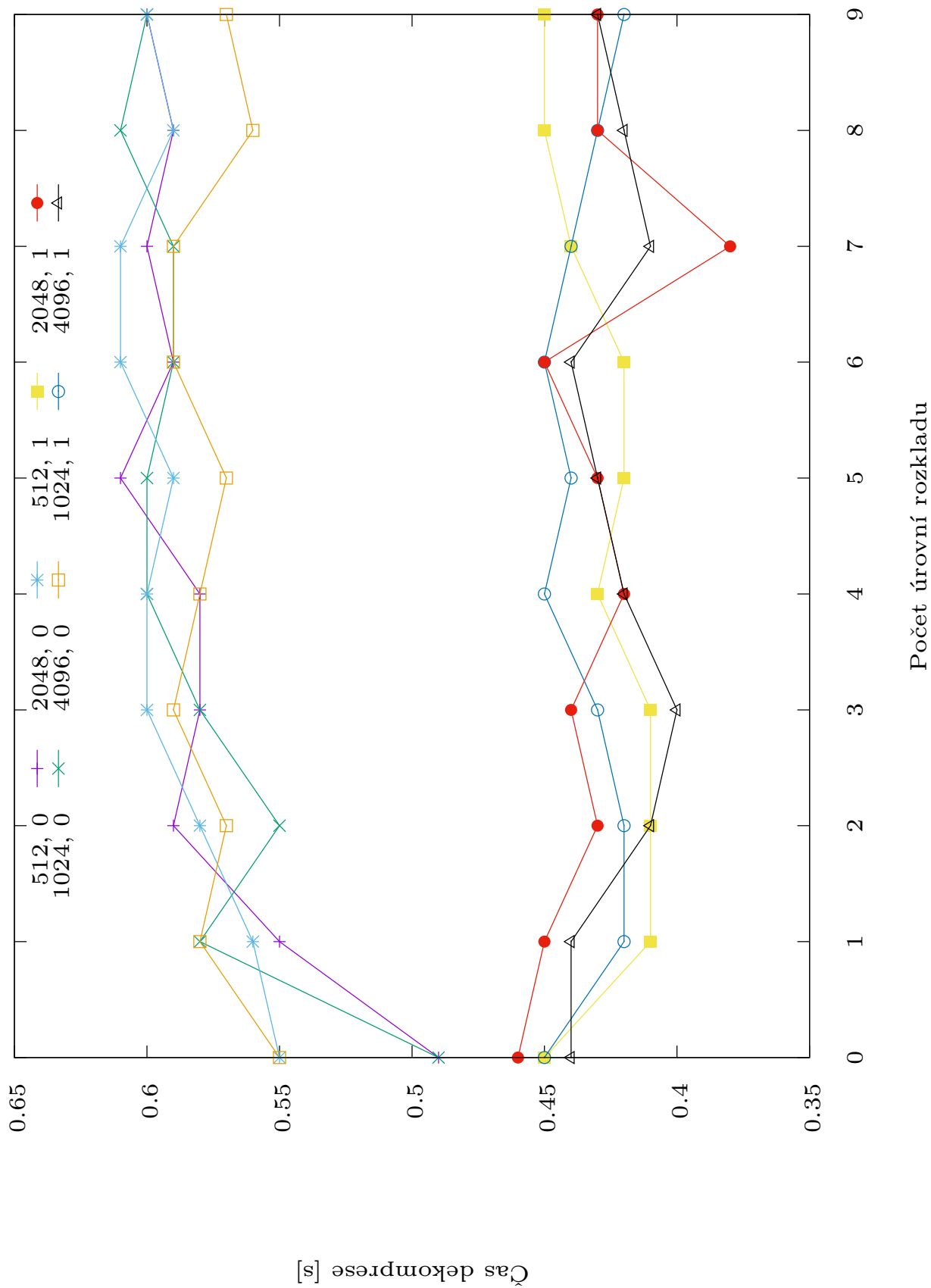
Obrázek 6.29: Kakadu - ztrátová komprese, skenované knihy - průměrný čas komprese jednoho obrazu



Obrázek 6.30: Kakadu - ztrátová komprese, skenované knihy - průměrné PSNR komprimovaného obrazu



Obrázek 6.31: Kakadu - ztrátová komprese, skenované knihy - průměrné PSNR dekomprimovaného obrazu



Obrázek 6.32: Kakadu - ztrátová komprese, skenované knihy - průměrný čas dekomprese jednoho obrazu

### 6.4.3 Shrnutí

Všeobecně lze říct, že jak komprese, tak dekomprese je s módem bypass rychlejší. U kvality potom nějak moc nezáleží ani na zvolené dlaždici, ani na módu, nejhůř se u obou knihoven jeví nepoužití módu při nízkém počtu úrovní rozkladu. PSNR u tohoto datasetu se u knihovny Kakadu pohybovalo kolem 70 dB, u knihovny OpenJPEG kolem 70 dB.

# Kapitola 7

## Závěr

Vlastní práce se skládá ze tří hlavních částí. V kapitole *Kompresa obrazu a JPEG 2000* byly vysvětleny některé základní pojmy v oblasti komprese obrazu a podrobněji zde byla probrána funkcionality formátu JPEG 2000. V kapitole *Implementace* byla popsána práce s jednotlivými knihovnamy a také princip fungování zvoleného frameworku pro analýzu nastavení jednotlivých parametrů formátu JPEG 2000.

Na začátku práce bylo v plánu zachování jednotlivých komprimovaných i dekomprimovaných obrazů k jejich následnému subjektivnímu posouzení. Od této varianty bylo však vzhledem k nárokům na úložný prostor upuštěno a jednotlivé obrazy se tak po zpracování objektivní metodou PSNR ihned zahazují. Kvůli komerční licenci knihovny Kakadu a nemožnosti nahlížení do kódu a debugování nebyla také realizována velice zajímavá možnost zkoumání tzv. mezirozkladů, kdy je obraz při kompresi rozdělen do několika vrstev kvality a při zpětné dekomprimaci obrazu se výsledná kvalita odvíjí od počtu dekomprimovaných vrstev. Tato část zde nechává tedy možnost pro další zpracování, za předpokladu plné verze knihovny Kakadu.

V kapitole *Analýza nastavení formátu JPEG 2000 pro bezztrátovou kompresi* byly do podrobnosti rozebrány jednotlivé datasety. Datasety byly celkem čtyři - 6 Mpx fotografie, skenované časopisy [1], skenované mapy [5] a skenované knihy [1]. Byla tak splněna podmínka nejen na různorodost datasetů, ale také na formát jednotlivých obrazů. Všechny použité obrazy byly ve formátu RGB24, nekomprimovaný TIFF. V kapitole *Analýza nastavení formátu JPEG 2000 pro ztrátovou kompresi* byly použity stejné datasety jako pro bezztrátovou kompresi. Analýza jednotlivých nastavení byla velice časově náročná pro některé datasety skript běžel několik dní (týdnů).

# Literatura

- [1] artists, V.: *Archive.org*. [Online; navštíveno 04.05.2018].  
URL [www.archive.org](http://www.archive.org)
- [2] authors, V.: *Komprese dat*. Listopad 2017, [Online; navštíveno 22.01.2018].  
URL [https://cs.wikipedia.org/wiki/Komprese\\_dat](https://cs.wikipedia.org/wiki/Komprese_dat)
- [3] Beichel, R. R.: *Image Data Compression*. The University of Iowa, Říjen 2000, [Online; navštíveno 20.01.2018].  
URL <http://user.engineering.uiowa.edu/~dip/lecture/DataCompression.html>
- [4] David S. Taubman, M. W. M.: *JPEG2000 Image Compression Fundamentals, Standards and Practice*. Springer, 2002, ISBN 978-1-4613-5245-7.
- [5] of Japan, G. I. A.: *Antarctic Geospatial Data*. [Online; navštíveno 04.05.2018].  
URL <http://www.gsi.go.jp/antarctic/index-e.html>
- [6] de Louvain, U.: *OpenJPEG*. 2018, [Online; navštíveno 04.05.2018].  
URL <http://www.openjpeg.org/>
- [7] Otemuyiwa, P.: *The Great JPEG 2000 Debate: Analyzing the Pros and Cons to Widespread Adoption Applications*. Duben 2017, [Online; navštíveno 21.01.2018].  
URL [https://cloudinary.com/blog/the\\_great\\_jpeg\\_2000\\_debate\\_analyzing\\_the\\_pros\\_and\\_cons\\_to\\_widespread\\_adoption](https://cloudinary.com/blog/the_great_jpeg_2000_debate_analyzing_the_pros_and_cons_to_widespread_adoption)
- [8] Sayood, K.: *Introduction to Data Compression*. Elsevier Inc., 2006, ISBN 978-0-12-620862-7.
- [9] Taubman, D.: *Kakadu Survey Documentation*. 2005, [Online; navštíveno 04.05.2018].  
URL <http://kakadusoftware.com/wp-content/uploads/2014/06/Kakadu.pdf>
- [10] Taubman, D.: *Kakadu software*. 2014, [Online; navštíveno 04.05.2018].  
URL <http://kakadusoftware.com>
- [11] Taubman, D.: *Kakadu's Distortion-Length Slopes*. 2014, [Online; navštíveno 04.05.2018].  
URL <http://kakadusoftware.com/wp-content/uploads/2014/06/kakadu-log-slope-definition.pdf>
- [12] Taubman, D.: *Usage Examples for the Demonstration Applications Supplied with Kakadu V7.10.2*. 2018, [Online; navštíveno 04.05.2018].  
URL [http://kakadusoftware.com/wp-content/uploads/2014/06/Usage\\_Examples2.txt](http://kakadusoftware.com/wp-content/uploads/2014/06/Usage_Examples2.txt)

- [13] Tinku Acharya, P.-S. T.: *JPEG2000 standard for image compression : concepts, algorithms and VLSI architectures* . John Wiley & Sons, Inc., 2005, ISBN 0-471-48422-9.
- [14] Urbánek, P.: *Komprese obrazu pomocí vlnkové transformace*. 2013, [Online; navštíveno 04.05.2018].  
URL [https://www.vutbr.cz/www\\_base/zav\\_prace\\_soubor\\_verejne.php?file\\_id=118893](https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=118893)
- [15] Vychodil, B.: *Standardy pro obrazová data*. 2012, [Online; navštíveno 04.05.2018].  
URL <http://www.ndk.cz/standardy-digitalizace/standardy-pro-obrazova-data>
- [16] Šimek, J.: *Využití pokročilých objektivních kritérií hodnocení při kompresi obrazu*. 2010, [Online; navštíveno 04.05.2018].  
URL [https://www.vutbr.cz/www\\_base/zav\\_prace\\_soubor\\_verejne.php?file\\_id=28290](https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=28290)



# Přílohy

# Příloha A

## Manuál

Skript je napsaný v jazyce Python3. Před spuštěním je nutné nainstalovat všechny požadavky ze souboru README.

Skript se spouští pouze s jedním parametrem, a to složkou se soubory, které mají být zpracovány. Soubory musí být ve formátu uncompressed TIFF a nesmí přesáhnout rozlišení 80 Mpx.

Jednotlivé parametry pro nastavení formátu JPEG 2000 se dají měnit v horní části skriptu. Skript obsahuje tři třídy:

- OpenJpeg – obsahuje 2 metody – compress a decompress
- Kakadu – obsahuje 2 metody – compress a decompress
- HelpingFunctions – obsahuje pomocné metody - prepare\_folder, count\_bpp, write\_to\_csv, parse\_folder, parse\_file, postprocess\_image, delete\_dir a delete\_image

### Popis volitelných nastavení:

- q\_layers – počet vrstev kvality
- tile\_collection – kolekce dlaždic
- codeblocks – velikost code-blocks
- p\_order – nastavení pořadí zpracování (progression order)
- x\_axis – offset na x ose
- y\_axis – offset na y ose
- modes – slovník módů
- reversible – explicitní argument pro Kakadu ('yes'|'no')
- irreversible – explicitní přepínač pro OpenJPEG (0|1)
- num\_threads – počet vláken (jen pro Kakadu)
- compression\_ratio\_kkd – kompresní poměr pro Kakadu (1:8 = 3)
- compression\_ratio\_opj – kompresní poměr pro OpenJPEG (1:8 = 8)

**Příklad spuštění:** python3 JPEG2000\_analyzator.py -i <soubor>