Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

Geometrická reprezentace obrazové informace

Plzeň, 2010 Martin Dohnal

Zadání

1. Seznamte se s existujícími technikami pro reprezentaci obrazu trojúhelníkovými sítěmi, které byly vyvinuty na ZČU v rámci projektu GAAV ČR KJB101470701.
2. Seznamte se s metodami komprese trojúhelníkových sítí a metodami pro kompresi obrazu reprezentovaného rastrem pixelů (zejména JPEG2000).
3. Seznamte se s existujícími metodami segmentace obrazu.
4. Navrhněte a implementujte hybridní metodu pro kompresi digitálního obrazu kombinující trojúhelníkové sítě a JPEG2000.
5. Vytvořené programové vybavení otestujte a proveďte zhodnocení.

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 20.5.2010, Martin Dohnal

Geometric representation of image information

During last several years, triangulations have been considered for digital image representation. Triangulations have potentially several advantages, which other representations lack. Mainly, it is simple geometric manipulation with image, for example zooming, rotation and various deformations. Geometric manipulation with an image in triangulation representation does not lead to information loss as in the case of a raster representation. Another advantage is hardware support (provided in graphic cards) of triangle meshes. Because of that, displaying and manipulation with triangulation can be very fast. However, existing representations of digital images using triangulations doesn't offer such quality as leading raster representations like JPEG 2000. It seems that triangulations are suitable only for specific image classes, which leads to a method combining both representations. Aim of this thesis is designing such hybrid method for image representation, while preserving most of advantages of each approach.

Geometrická reprezentace obrazové informace

Již několik let se uvažuje o použití trojúhelníkových sítí pro reprezentaci digitálního obrazu. Trojúhelníkové sítě potenciálně skýtají množství výhod, které ostatní reprezentace postrádají. Jedná se především o snadnou geometrickou manipulaci s obrazem – zvětšování a zmenšování, rotace a různé deformace obrazu. Aplikací těchto operací totiž nedochází ke ztrátě informace, jako je tomu u rastrových reprezentací. Další výhodou je také podpora v hardwaru (zejména v grafických kartách), díky které je zobrazení a manipulace s trojúhelníkovou sítí velice rychlá. Současné reprezentace trojúhelníkovými sítěmi však v obecném případě nedosahují kvalit vedoucích rastrových přístupů jako např. JPEG 2000. Zdá se, že trojúhelníkové sítě jsou vhodné jen pro určité typy obrazu, což vede na metodu kombinující oba přístupy. Cílem práce je tedy navrhnout takovou hybridní metodu reprezentace obrazu, která bude kombinovat co nejvíce výhod obou zmíněných přístupů.

Obsah

1 Úvod....6
2 Digitální obraz....7
2.1 Reprezentace dat....7
2.2 Barevné modely....8
2.3 Kvalita obrazu....11
3 Trojúhelníkové sítě....14
3.1 Triangulace....14
3.2 Projekt na ZČU....17
4 Metody komprese....30
4.1 Trojúhelníkové sítě....30
4.2 Rastrové komprese....34
5 Segmentace obrazu....39
5.1 Rozdělení na kanály....39
5.2 Transformace obrazu....39
5.3 Rozdělení na prostorové oblasti....41
6 Analýza vhodných dat....43
6.1 JPEG 2000 kodek....43
6.2 Standardní sada testovacích obrazů....45
6.3 Speciální obrazy....48
6.4 Shrnutí výsledků....54
7 Úpravy triangulace....55
7.1 Předzpracování....55
7.2 Kvantizace....58
7.3 Odlišná konstrukce triangulace....63
7.4 Shrnutí úprav....65
8 Návrhy hybridních metod....68
8.1 Segmentace na regiony....68
8.2 Segmentace na trojúhelníky....72
8.3 Triangulace jako předzpracování....74
8.4 Rozdílová metoda....76
8.5 Segmentace podle kanálů barveného obrazu....79
9 Závěr....81
Použité zdroje....84

1 Úvod

Po desítky let je přední reprezentací digitálního obrazu rastr pixelů. Obraz je však v této formě velice objemný a tak bylo vytvořeno mnoho kompresních postupů, které tento problém téměř eliminovaly. Rastrová reprezentace ale i přesto skýtá určité nevýhody. Tím, že jsou pixely umístěny na mřížce, dochází při geometrické manipulaci s obrazem (například rotace či zvětšení) ke ztrátě dat, protože každý pixel musí být interpolován.

V poslední době se tak pro reprezentaci digitálního obrazu uvažuje použití trojúhelníkových sítí. Trojúhelníkové sítě mají několik výhod, které ostatním reprezentacím chybí. Při geometrických transformacích obrazu nemusí docházet ke ztrátě dat, protože se pouze přesouvají vrcholy sítě. Díky tomu je tento přístup vhodnější pro zachování hran v obraze. Zvětšení obrazu navíc nezpůsobuje vznik dalších nepřesností, jako je tomu u rastrového obrazu, kde jsou vidět jednotlivé čtvercové pixely. Další důležitou výhodou je hardwarová akcelerace sítí pomocí grafických karet. Trojúhelníkové sítě jsou v počítači nejčastější reprezentací pro trojrozměrné objekty, a tak bylo jejich zobrazování i manipulace s nimi značně urychleny. Interpolace hodnot uvnitř trojúhelníků je pak přímo implementována v hardwaru.

Protože však většina zařízení na snímání obrazu pracuje právě s rastrovou reprezentací, klíčovým problémem je převod obrazu na trojúhelníkovou síť. Pro uchování trojúhelníkové sítě je zapotřebí ukládat kromě hodnot jasů v obraze i pozice vrcholů (případně ještě další informace v závislosti na typu trojúhelníkové sítě), a tak je nutné vybrat ze všech bodů na mřížce jen určitou podmnožinu. Výběr optimální podmnožiny pro požadovanou kvalitu obrazu je však NP-úplný problém a proto bylo v uplynulé době navrženo množství algoritmů, které mají za cíl se optimální množině nejvíce přiblížit.

Jenom zde na univerzitě se reprezentací digitálního obrazu trojúhelníkovou sítí zabývalo již několik bakalářských a diplomových práci. Přes veškerou snahu jsou ale navrhovaná řešení obecně méně kvalitní, než vedoucí rastrové přístupy, jako třeba JPEG 2000. Tento závěr prozatím platí i pro přístupy vytvořené jinde ve světě. Bylo ale dosaženo jistých úspěchů pro specifické třídy obrazových dat. Zdá se tedy, že použití trojúhelníkových sítí je výhodné jen pro určitou podmnožinu obrazových dat, což vede na metodu kombinující obě reprezentace. Výsledkem by ideálně měla být hybridní metoda, která spojuje co nejvíce výhod obou přístupů, což je právě cílem této práce.

V práci jsou nejdříve uvedeny základní charakteristiky digitálního obrazu. Rozvedeny jsou možné reprezentace obrazu a používané barevné modely. Dále jsou shrnuty poznatky z předešlých prací vytvořených na ZČU vztahujících se k tématu trojúhelníkových sítí pro digitální obraz. Poté jsou popsány kompresní metody jak pro trojúhelníkové sítě, tak i pro standardní reprezentaci obrazu rastrem. Tato kapitola se soustředí zejména na formát JPEG 2000, který má být ve výsledné hybridní metodě použit. Teoretickou část práce ukončuje kapitola o segmentaci obrazu. Kapitola stručně uvádí některé techniky umožňující rozdělit obraz na více částí, které budou pro návrh hybridní metody nutné.

Dále následuje praktická část práce, která je složena z několika kapitol. Nejprve je nutné identifikovat, jaká třída dat je výhodná pro triangulaci, či nevýhodná pro JPEG 2000. Následuje několik návrhů, jak triangulaci vytvořenou na ZČU vylepšit pro potřeby hybridní metody. Nakonec následuje samotný návrh hybridních metod a samozřejmě dosažené výsledky. V závěru práce jsou pak tyto výsledky shrnuty a nastíněn směr, jakým by se mohl ubírat další výzkum.

2 Digitální obraz

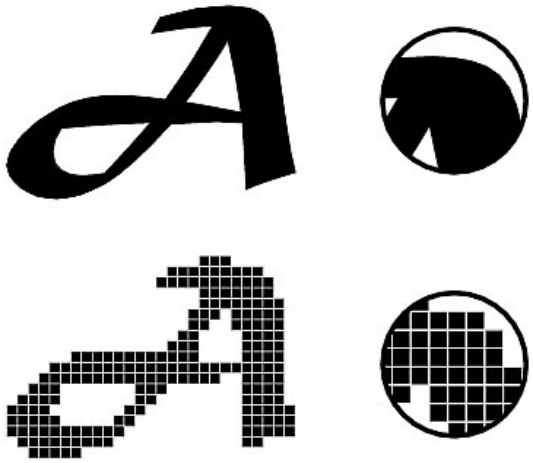
Digitální obraz je dvourozměrný obraz reprezentovaný sledem binárních hodnot. Je to tedy jakýkoliv obraz uchovaný v počítači. Protože obrazy v nekomprimované formě zabírají poměrně hodně místa, bylo za uplynulé desítky let vymyšleno mnoho postupů, jak tuto velikost snížit (bez i se ztrátou kvality obrazu). Při kódování digitálního obrazu jsou klíčové především tyto volby:

- zvolená reprezentace dat,
- použitý barevný model,
- metoda komprese.

2.1 Reprezentace dat

Reprezentace se zpravidla rozdělují pouze na dva typy – rastrové a vektorové. V rastrové reprezentaci je základní jednotkou pixel, který popisuje bod obrazu. Pixely jsou rovnoměrně rozmístěny ve dvourozměrné mřížce (proto rastr). Většina obrazových formátů používá právě tento přístup (JPEG, PNG, BMP, GIF, TIF, JPEG 2000). Výhodou tohoto přístupu je jednoduchost zpracování ze vstupních dat a přímočarý formát dat. Problémem může být následná manipulace s obrazem – například zvětšení, které nutně vede k interpolaci dat, kvůli zvětšení rastru.

Druhou možností je vektorová reprezentace. Obrazová data jsou popisovány geometrickými objekty – křivky, polygony a podobně. Na rozdíl od rastrového formátu, při získávání obrazu z hardware (či převodu z rastrové formy) je nutné tyto základní elementy rozpoznat, tak aby co nejlépe odpovídaly vstupním datům. To je praxi náročná záležitost, a proto se většinou pro reprezentaci obecného obrazu nepoužívají. Vektorové obrazy spíše vznikají jako výstup vektorových editorů, které přímo pracují se základními elementy, nebo se využívají při rozpoznávání objektů z obrazu. Mezi vektorové formáty patří především EPS, EMF a SVG. Velkou výhodou přístupu je velmi jednoduchá a přesná geometrická manipulace – například rotace nebo zvětšení. Protože pracujeme s geometrickými objekty, nedochází při manipulaci k interpolaci dat v důležitých místech obrazu a tak ani ke ztrátě informace.



Obraz 2.1: Vektorový (horní část) a rastrový (dolní část) popis digitálního obrazu.

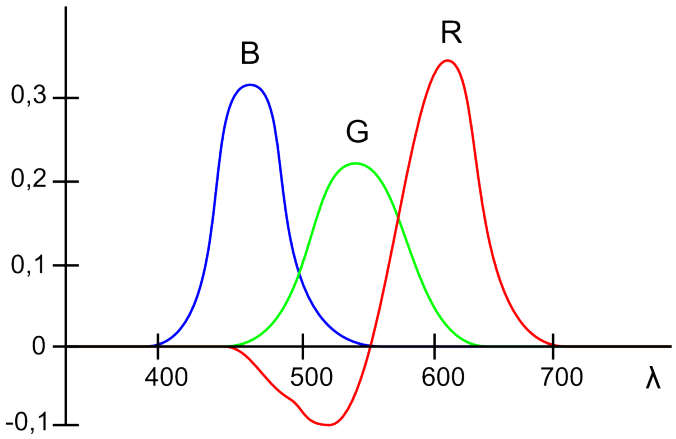
2.2 Barevné modely

Barevné modely určují, jak několika základními barevnými složkami popsat celý barevný prostor. Každou barvu v obraze je možné složit například ze tří základních barev – červená, zelená a modrá (základní barvy je možné volit i jiné). Modely jsou ale zpravidla omezeny pouze na určitou část barevného prostoru. Například zmíněné tři základní barvy odpovídají aditivnímu barevnému modelu RGB, který je nejvíce používaný a taky nejznámější. Existuje ale i mnoho dalších modelů, které používají jiné základní barevné složky, či barvy jinak skládají. Vznikly z několika hlavních důvodů. Například subtraktivní model CMYK se používá v tiskových technologiích. Model HSL nebo HSV je výhodný, pokud chceme snadno měnit zvlášť barevné odstíny obrazu nebo jas, protože jsou tyto dvě složky odděleny. Model YCbCr je vhodný pro digitální přenosy nebo kompresi. Poslední uvedený důvod je v této práci nejdůležitější, a proto následuje stručný přehled barevných modelů, které poskytují dobré kompresní poměry oproti jiným modelům.

2.2.1 RGB model

RGB model je zde uveden spíše kvůli tomu, že je základní. Je ovšem zahrnut i do testování komprese trojúhelníkových sítí s použitím různých barevných modelů (viz [Syk08]), které budou uvedeny dále.

Model má tři základní složky, z nichž každá představuje intenzitu jedné základní barvy modelu (červená, zelená, modrá). Model je aditivní, což znamená, že sčítání jednotlivých složek zvyšuje jas. Existuje několik variant, které se liší v přesných hodnotách základních složek – například sRGB nebo Adobe RGB.



Obraz 2.2: Koeficienty RGB kanálů pro skládání světla s určitou vlnovou délkou (a nutnost záporného koeficientu pro červený kanál).

Nevýhodou modelu je poměrně neintuitivní reprezentace složených barev a celkově uživatelské úpravy obrazu. RGB model navíc nemá na rozdíl od většiny ostatních modelů oddělený jasový kanál. Zejména z tohoto důvodu se dá očekávat horší kompresní poměr. Dalším problémem tohoto modelu je nemožnost popsat některé barvy. Pro jejich popsání by totiž bylo nutné použít záporný R koeficient (viz obr. 2.2). Z praktického hlediska to ale v této práci nemá žádný význam, protože všechny ostatní modely jsou převáděny právě z RGB modelu.

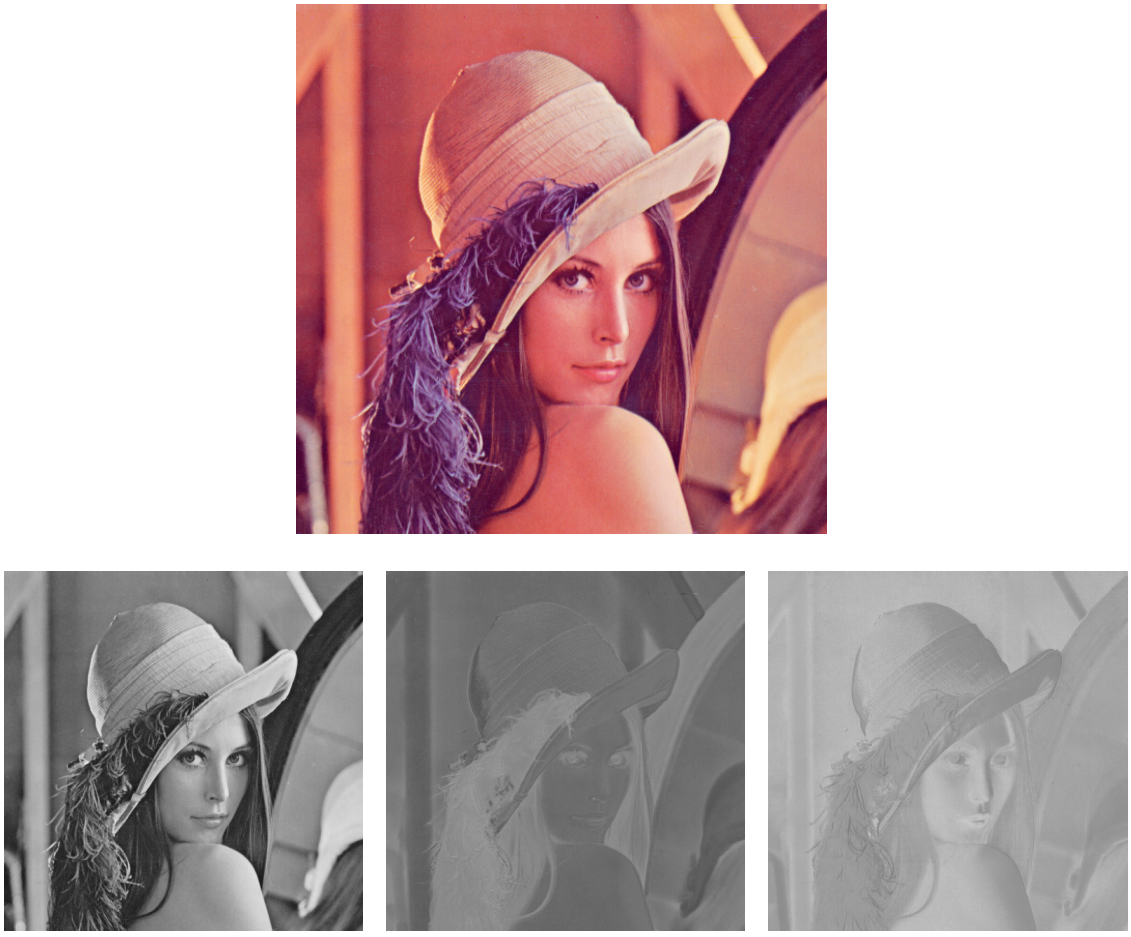
2.2.2 YCbCr model

Tento model má také tři základní složky, nicméně ty neodpovídají základním barvám. Y kanál popisuje jas, zatímco Cb a Cr kanály určují barevný odstín. Model lze jednoduše získat z RGB modelu, podle následujícího vztahu.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.1687 & -0.3313 & 0.5 \\ 0.5 & -0.4187 & -0.0813 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Tento konkrétní převod je použit v JPEG kompresích (viz [JFIF92] a [JPE00]). Jiné aplikace tohoto modelu mají odlišné koeficienty a jiný výstupní rozsah jednotlivých kanálů. V práci se používá právě JPEG převod, protože rozsahy všech kanálů jsou stejné jako rozsahy původního modelu, tj. <0;255>, takže převodem nedochází k přílišné ztrátě informace.

Rozložením obrazu na jednotlivé kanály (viz obr. 2.3) pak vidíme distribuci informace do jednotlivých složek. Zatímco z kanálu Y určujícího jas, je původní obraz bez problému rozpoznatelný, u kanálů Cb a Cr to již může činit potíže. Z rozkladu je vidět, že jasová složka obsahuje mnohem více informace. Pokud se pak tento model komprimuje, využívá se výše zmíněných faktů. Stěžejní je komprese jasové složky. U kanálů popisující odstín lze snížit bitovou hloubku nebo kanály dokonce podvzorkovat (použití například v TV signálu). Model se tedy komprimuje poměrně dobře, což dosvědčuje jeho použití ve formátech JPEG.



Obraz 2.3: Jednotlivé kanály modelu YCbCr. Zleva Y, Cb a Cr.

2.2.3 L*u*v* model

Podobně jako $YCbCr$ model, i zde je oddělena jasová složka od odstínu. $L^*u^*v^*$ model vychází z barevného modelu CIE XYZ, který napravuje nedostatky standardního RGB modelu – nemožnost reprezentovat celé spektrum (pomocí kladných koeficientů). Převody do barevného systému CIE XYZ a $L^*u^*v^*$ jsou čerpány ze [Scha07].

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{0.17697} * \begin{bmatrix} 0.49 & 0.31 & 0.20 \\ 0.17697 & 0.81240 & 0.01063 \\ 0.00 & 0.01 & 0.99 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$L^*u^*v^*$ model upravuje model XYZ tak, aby lépe odpovídal vnímaní jasů lidskému oku. Cílem modelu je, aby jasová složka byla podle lidského vnímání stejnoměrně rozložena po celém rozsahu.

$$L^* = \begin{cases} 116\left(\frac{Y}{Y_r}\right)^{1/3} - 16, & \frac{Y}{Y_r} > \left(\frac{6}{29}\right)^3 \\ \left(\frac{29}{3}\right)^3, & \frac{Y}{Y_r} < \left(\frac{6}{29}\right)^3 \end{cases}$$

$$u^* = 13\,L^*\,(u' - u_r')$$

$$v^* = 13\,L^*\,(v' - v_r')$$

$$u' = \frac{4\mathrm{X}}{X + 15\mathrm{Y} + 3\mathrm{Z}} \quad v' = \frac{9\mathrm{Y}}{X + 15\mathrm{Y} + 3\mathrm{Z}}$$

Kde u_r', v_r' a Y_r jsou referenční hodnoty pro bílou barvu.

Z pohledu komprese by zde neměl být oproti systému $YCbCr$ příliš velký rozdíl. Dalo by se očekávat určité zlepšení vizuální kvality, pokud by byla použita kvantizace hodnot. Snížení bitové hloubky by mohlo být pro lidské oko méně znatelné (oproti stejné operaci v jiných barevných prostorech) vlivem stejnoměrně rozložené jasové složky přes celý rozsah.

2.3 Kvalita obrazu

Kvalitou digitálního obrazu se zde rozumí porovnání nějak upraveného obrazu s originálem, přičemž úpravu v tomto případě představuje právě komprese obrazu. Standardem pro měření kvality obrazu je poměr PSNR (Peak Signal-to-Noise Ratio). Dále se pak ještě začíná používat metoda SSIM (Structural Similarity). Oba přístupy budou dále podrobněji popsány. Je ovšem nutné dodat, že hodnocení je ryze objektivní, a nemusí odpovídat názoru člověka, který obraz vnímá odlišně (i když oba přístupy se snaží lidskému vnímání přiblížit).

2.3.1 PSNR

PNSR metoda je založena na výpočtu průměrné chyby zkoumaného obrazu od originálu. Vznikla zejména pro potřeby analýzy jednorozměrných signálů, takže se používá již poměrně dlouho a je velice rozšířena. V kontextu porovnávaní digitálních obrazů metoda zohledňuje pouze vztah pixelů na odpovídající si pozici – nepočítá tedy s okolím bodu. Pokud by byl například komprimovaný obraz stejný jako originál a pouze posunutý třeba doprava o jeden pixel, PSNR metoda by mohla určit, že jsou obrazy naprosto odlišné. Oproti tomu člověk by si posunu ani nemusel všimnout. Výpočet je založen pouze na součtu odchylek intenzity pixelů.

$$MSE = \frac{1}{m\,n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (X_{i,j} - Y_{i,j})^2$$

Hodnota m a n je šířka a výška obrazu v pixelech, $X_{i,j}$ a $Y_{i,j}$ jsou intenzity pixelů na pozici určené indexy i a j. Sčítají se tedy všechny odchylky v obraze.

$$PSNR\,[dB] = 10 \log_{10} \left(\frac{MAX^2}{MSE} \right)$$

Výsledná hodnota je pak poměr mezi kvadrátem maximální hodnoty intenzity pixelu (bitová hloubka obrazu –1) a vypočtenou sumou odchylek. PSNR poměr je ještě logaritmován pro lepší odlišení nízkých a vysokých hodnot. Typické hodnoty PSNR jsou od 30-50 dB. Zhruba od 40 dB je zkoumaný obraz lidským okem téměř nerozeznatelný od originálu.

Obraz 2.4: Obraz Lena komprimovaný do JPEG 2000 formátu při 25, 30 a 40 dB. Originál je v této velikosti prakticky shodný s obrazem o 40 dB.

2.3.2 SSIM

Metoda SSIM se snaží vylepšit odhad kvality tak, aby lépe odpovídal lidskému vnímání. Na rozdíl od PSNR metody porovnává větší části obrazu, než je pixel. Jedna část je zpravidla čtvercová oblast. Pro porovnání celkové podobnosti obrazů jsou pak výpočty jednotlivých oblastí zprůměrovány.

Kvalita oblasti se počítá následovně:

$$SSIM(x,y)=\frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)}$$

Parametry x a y určují oblasti obou obrazů, μ_x a μ_y jsou střední hodnoty intenzit všech pixelů v odpovídající oblasti, σ_x , σ_y a σ_{xy} jsou rozptyly hodnot v oblastech a c_1 , c_2 jsou konstanty určené podle bitové hloubky obrazu.

$$c_1=(k_1 MAX)^2 \ , \ c_1=(k_2 MAX)^2$$

Střední hodnota indikátoru pro celý obraz je pak:

$$MSSIM=\frac{1}{M}\sum_{j=1}^{M} SSIM(x_j, y_j) \ ,$$

kde M je počet zkoumaných oblastí. Výsledkem indikátoru je hodnota od -1 do 1, přičemž číslo jedna znamená dokonalou shodu s originálem.

Je vidět, že na rozdíl od PSNR metody jsou zde určité možnosti jak výpočet ovlivnit – volba velikosti oblastí, volba počtu oblastí, zda se oblasti překrývají a podobně. Také jsou ve výpočtu použity určité konstanty pro zjištění parametrů c . Tyto konstanty jsou ale jen pouhým doporučením a lze je v metodě volit. Možností je tedy hodně, což znamená, že metodu lze přizpůsobit zpracovávaným datům. Z hlediska porovnávaní tím ale vznikají nepříjemnosti, pokud různé práce používají odlišný typ výpočtu. Proto je důležité všechny volby uvádět a brát je při porovnávání v úvahu.

Obraz 2.5: Obrazy se stejnou hodnotou PSNR. SSIM pro levý obraz (posunutý oproti originálu o jeden pixel) je 0.75 zatímco pro pravý (zkomprimovaný JPEG 2000) je hodnota rovna 0.65.

Pro porovnání na obraze 2.5 bylo počítáno s oblastmi o velikosti 8x8, které se posouvají o jeden pixel přes celý obraz. Konstanty $k_1 = 0.01$ a $k_2 = 0.03$ jsou doporučeny autory metody. Jiných výsledků je možno dosáhnout použitím kruhově omezených oblastí s gaussovskými váhami, jak autoři navrhují.

Ve [Wang04], kde je samotná SSIM metoda navržena, bylo i experimentálně ověřeno, že tato metoda odpovídá lépe lidskému vnímání kvality obrazu. Nevýhodou metody je, že není zatím příliš rozšířena (vznikla až po roce 2000). Pro vyšší kvalitu obrazu navíc neposkytuje dostatečné odlišení – rozdíly SSIM indikátoru se pohybují v řádech setin až tisícin. Pokud je nutné srovnání s výsledky jiných prací, je často nutné přistoupit na PSNR. V poslední době se tak většinou uvádějí obě metody.

3 Trojúhelníkové sítě

Trojúhelníková síť je datová struktura popisující plochu. Skládá se z bodů, které určují vrcholy trojúhelníků a hran, které určují strany trojúhelníků, přičemž trojúhelníky se nesmí překrývat. Je asi zřejmé, že trojúhelníkové sítě jsou v rámci digitálního obrazu vektorovou metodou. Trojúhelníkové sítě tak nabízejí větší flexibilitu při manipulaci s daty, ale za cenu náročnějšího zpracování vstupního obrazu (převodu na trojúhelníkovou síť).

V kontextu této práce bude trojúhelníková síť vytvářena z podmnožiny bodů, které se vyskytují na rastru vstupního obrazu. Možností, jak body pospojovat v trojúhelníkovou síť, je mnoho. Zajímavá je především určitá podmnožina trojúhelníkových sítí zvaná triangulace, jež bude více rozvedena. Následuje popis několika konkrétních typů trojúhelníkových sítí (resp. triangulací), které mají dobré vlastnosti zejména při kompresi. Další kapitola pojednává o projektu vytvořeném na ZČU zabývající se praktickou aplikaci triangulací pro digitální obraz.

3.1 Triangulace

Pokud je dána množina bodů P, triangulaci ve dvourozměrném prostoru lze definovat jako rozdělení prostoru na trojúhelníky s vrcholy P. Přitom je nutné splnit několik podmínek:

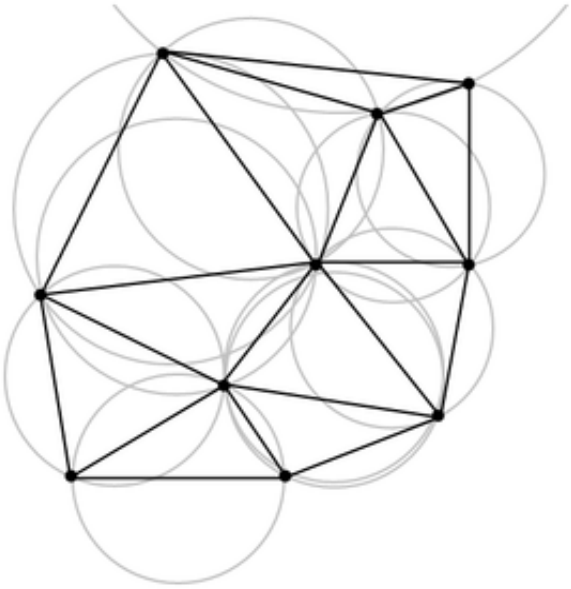
- Vrcholy každého trojúhelníku musí patřit do množiny P.
- Průsečíkem dvou trojúhelníků může být pouze vrchol trojúhelníků nebo hrana.
- Množina trojúhelníků je maximální – nelze přidat žádný další trojúhelník. Triangulace je tedy omezena hranami konvexní obálky množiny bodů P.

Tyto podmínky samy o sobě ale pořád nevedou na jednoznačné řešení. Proto vznikly specifické postupy tvorby triangulace, které upřednostňují určité vlastnosti výsledné trojúhelníkové sítě. Z nich je asi nejznámější Delaunayova triangulace, která maximalizuje minimální úhly trojúhelníků. Jinými slovy – snaží se vytvořit co nejrovnostrannější trojúhelníky. Takové trojúhelníky jsou pro počítačovou grafiku výhodné, protože při práci s nimi nevznikají velké numerické nepřesnosti.

3.1.1 Delaunayova triangulace

Podmínka Delaunayovy triangulace je definována tak, že se uvnitř kružnice opsané každého existujícího trojúhelníku nesmí nacházet žádný vrchol ze vstupní množiny bodů. Pokud je toto pravidlo porušeno, je nutno trojúhelníky zkonstruovat jinak – vrcholy jsou pevně dány, takže se musí vybrat jiné hrany. Pro konstrukci této triangulace existuje mnoho algoritmů, jejichž přehled lze najít na [Wiki10a]. Podrobněji bude popsán pouze algoritmus inkrementální konstrukce (v kapitole 3.2.1), protože je využit v projektu, na který tato práce navazuje.

Delaunayova triangulace je jednoznačná, pokud se na opsané kružnici každého trojúhelníku nachází vždy právě tři vrcholy (viz obr. 3.1). V takovém případě pak k uchování triangulace postačuje pouze samotná množina bodů, a hrany lze pokaždé stejně zrekonstruovat. Díky tomu je tato triangulace velmi výhodná právě pro různé kompresní metody. V případě více vrcholů na kružnici je možné triangulaci sestavit více způsoby, a tak by bylo nutné určit konkrétní triangulaci přidáním dalších informací.



Obraz 3.1: [Wiki10a] Delaunayova triangulace na 10ti vrcholech se zobrazením opsaných kružnic. Uvnitř opsaných kružnic nesmí ležet žádný vrchol.

3.1.2 Triangulace s omezením

Často je důležité, aby v síti byly určité předem definované hrany. K tomu slouží právě triangulace s omezením. Většinou vycházejí z již existujícího typu triangulace (zejména Delaunayovy), jejíž pravidla jsou při konstrukci upravena.

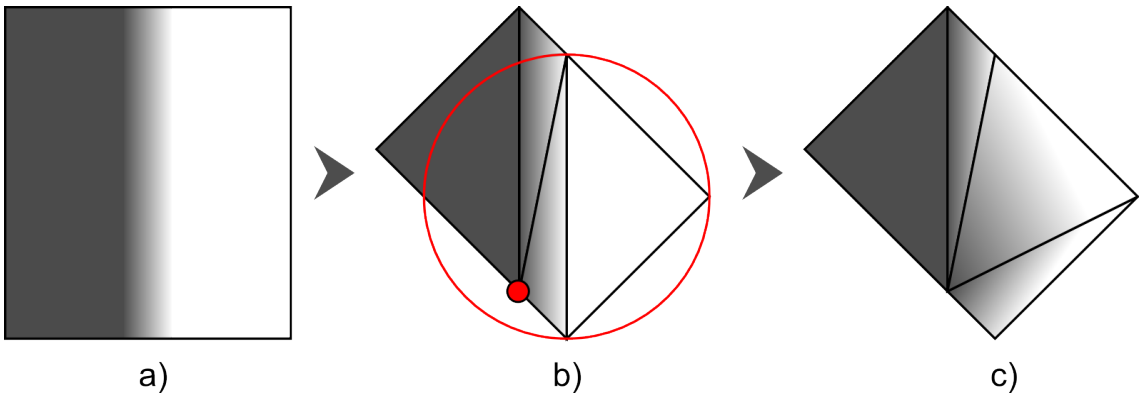
Bylo by možné například detekovat hrany v rastrovém obrazu nějakým dvourozměrným hranovým filtrem a tyto hrany pak vynutit při konstrukci triangulace. Z hranových filtrů získáme ale pouze zvýrazněné pixely, které hrany tvoří. Seskupení takových pixelů musíme nejprve převést na úsečky, které budou tvořit hrany trojúhelníků, což je sám o sobě problematický krok. Protože hrany jsou vynucené, musejí být uchovávány spolu s body triangulace a tím narůstá celková velikost.

3.1.3 Datově závislé triangulace

Datově závislé triangulace berou v potaz při konstrukci navíc hodnoty v daných bodech. V kontextu této práce to je jas bodu v obraze. Otázkou je, jaké by bylo vhodné kritérium, zda je hrana vhodná či nikoliv.

Typickým problémem reprezentace obrazu Delaunayovou triangulací je následující příklad. Pro reprezentaci ostré hrany v obraze jsou potřeba vrcholy na obou stranách hrany. Díky tomu je možné vytvořit úzké trojúhelníky reprezentující ostrou hranu. Pokud je ale zároveň nutné zachovat Delaunayovo kritérium, může se stát, že budou hrany prohozeny nepatřičným způsobem, jak je znázorněno v následujícím obr. 3.2.

Datově závislá triangulace by tento problém mohla řešit, pokud by bylo použito vhodné kritérium (nebo jejich kombinace). Například v uvedeném příkladu by stačilo minimalizovat rozdíl jasů ve vrcholech trojúhelníku. Zmíněný nedostatek lze ovšem odstranit i u Delaunayovy triangulace a to její vhodnou konstrukcí (volbou vrcholů, u kterých nedojde k prohození hran trojúhelníků).



Obraz 3.2: Problém reprezentace ostrých hran Delaunayovou triangulací.

a) hrana v původním obraze; b) triangulace a nesplnění Del. podmínky; c) nutné prohození hran trojúhelníků a poškození hrany v obraze.

3.2 Projekt na ZČU

Projekt GAAV ČR KJB101470701 se zabývá alternativními reprezentacemi obrazové informace. Na projektu se pracovalo od roku 2007 do roku 2009. Na projektu převážně pracovali Ing. Josef Kohout, Ph.D. a Doc. Dr. Ing. Ivana Kolingerová. Některé části projektu pak byly vypracovány jako bakalářské či diplomové práce:

- Interpolační metody pro digitální obrazy reprezentované triangulacemi – Bc. Tomáš Janák;
- Komprese barevných digitálních obrazů s využitím triangulace – Bc. Radek Sýkora;
- Použití triangulací pro reprezentaci videa – Ing. Petr Puncman;
- Přímá manipulace s obrazy reprezentovanými triangulacemi – Bc. Josef Vyškovský.

Projekt je již od počátku omezen výhradně na triangulace zejména kvůli bezproblémovým geometrickým transformacím, podpoře interpolací přímo v hardwaru a mnoha existujícím technikám pro jejich kompresi. Soustředí se zejména na výběr nejvhodnějších postupů pro konstrukci a uložení triangulace vytvořené z podmnožiny bodů rastrového obrazu. Dále se také věnuje použití triangulací pro reprezentaci videa a úpravám obrazů reprezentovaných trojúhelníkovou sítí. Zde budou ale shrnuty pouze části relevantní pro tuto práci.

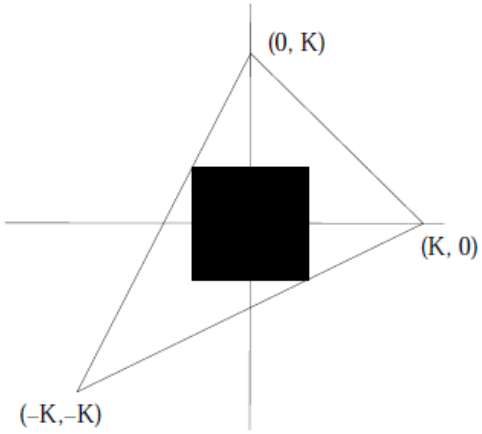
3.2.1 Výběr a konstrukce triangulace

Jako vhodné triangulace autoři uvádí Delaunayovu triangulaci, váženou triangulaci, triangulaci s omezením a datově závislou triangulaci (zmíněné zde v kapitole 3.1). Všechny jsou z hlediska velikosti nutných dat poměrně úsporné, nicméně projekt dále uvažuje pouze Delaunayovu triangulaci. Její konstrukce je nejjednodušší, potřebuje uchovávat nejméně dat (pouze vrcholy trojúhelníkové sítě) a mnoho kompresních algoritmů pracuje právě výlučně s ní.

Aby bylo dosaženo komprese srovnatelné s rastrovými přístupy, musí být vybrána pouze malá podmnožina bodů rastru. Tím je vyvážena nutnost uchovávat u triangulace pozice vrcholů. Protože k hodnocení důležitosti vrcholů, které budou v triangulaci ve finále zachovány se používají lokální kritéria (podrobněji níže), musí probíhat tvorba triangulace postupně. Z toho plyne výběr algoritmu inkrementální konstrukce triangulace.

Inkrementální algoritmus

Inkrementální konstrukce spočívá v postupné tvorbě sítě. Jako počáteční stav sítě musí být zvolena nějaká trojúhelníková síť, která bude fungovat jako konvexní obálka pro všechny vkládané body. Je předem dáno ze struktury digitálního obrazu, že body tvoří obdélníkovou (nebo čtvercovou) oblast. Oblast může být uzavřena do čtvercové konvexní obálky tvořené dvěma trojúhelníky. Problémem ale může být nestabilita při výpočtech, protože každý bod na kraji obrazu musí být spojen hranou s danou konvexní obálkou. To vede k vytvoření dlouhých úzkých trojúhelníků, které zvyšují nestabilitu výpočtu. Oblast je tedy uzavřena do jednoho trojúhelníku (viz následující obraz 3.3), ve kterém dochází k menším numerickým nepřesnostem (viz [Žal03]).



Obraz 3.3: [Koh09] Uzavření datové množiny do trojúhelníku při inicializaci inkrementální triangulace. Koeficient K je závislý na rozměrech obrazu, ze kterého je triangulace vytvářena.

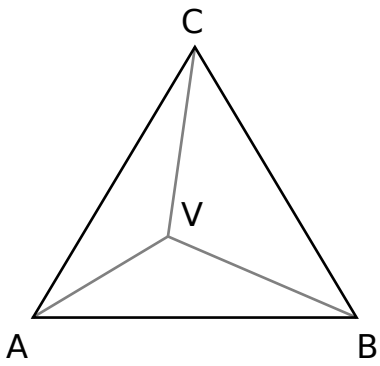
Do sítě se pak opakovaně přidávají vrcholy následným postupem:

1. Lokalizace trojúhelníku obsahujícího nový vrchol **V**.

   Trojúhelník může být nalezen prostým zkoušením všech možností. Vzhledem k tomu, že lokalizace ale probíhá v každém kroku, je nutné použít sofistikovanější přístup. Konkrétně se používá speciální algoritmus procházky „remembering stochastic walk" navržený v [Dev01] (viz [Koh09]), který v nejhorším případě nalezne trojúhelník obsahující zadaný vrchol v čase $O(N)$. V průměru je však složitost téměř $O(\sqrt{N})$. Existují i rychlejší algoritmy se složitostí $O(\log N)$, které ale mají větší paměťové nároky. Porovnání jednotlivých metod lze najít v [Žal03].

2. Rozdělení trojúhelníku na několik trojúhelníků s vrcholem **V**.

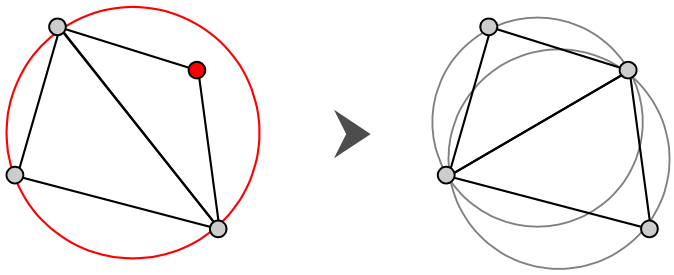
   Poté, co je trojúhelník obsahující nový vrchol lokalizován, musí být rozdělen. Pokud je vrchol uvnitř trojúhelníku, vzniknou místo něj tří trojúhelníky – každý s vrcholem **V** (viz obr. 3.4). Může se ale také stát, že se nový vrchol nachází na hraně. V takovém případě musí být rozdělena hrana a dva trojúhelníky, které mají tuto hranu společnou. Z povahy úlohy nikdy nemůže nastat možnost, že by vrchol byl přidáván na pozici již existujícího vrcholu.



Obraz 3.4: Vložení nového vrcholu V a rozdělení trojúhelníku s vrcholy ABC již existujícího v triangulaci.

3. Oprava sítě tak, aby splňovala Delaunayovu podmínku.

   Pro nové trojúhelníky se musejí vypočítat opsané kružnice a zkontrolovat, zda není porušena podmínka Delaunayovy triangulace (obr. 3.5). Pokud je nutné prohodit hranu, musí být podmínka opět ověřena na upravených trojúhelnících a tak dále. Oprava se tak může rozšířit po celé síti. V průměru je však prohazování hran omezeno jen na malou část sítě. Následuje opět krok číslo **1** – přidání dalšího vrcholu.



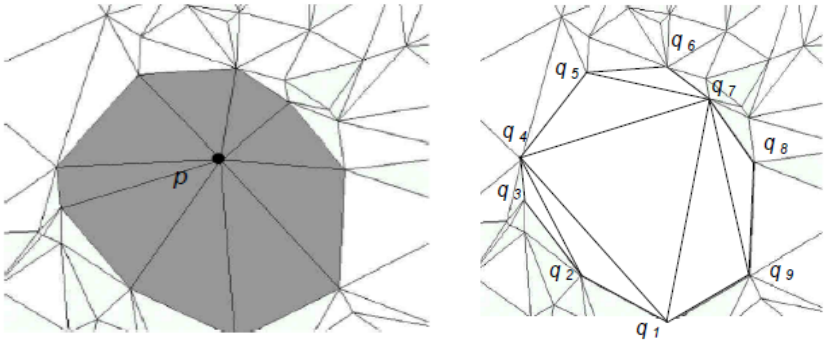
Obraz 3.5: Prohození hran dvou trojúhelníků kvůli Delaunayově podmínce.

4. Dokončení.

   Po přidání všech požadovaných vrcholů, což je většinou určeno splněním nějaké kvalitativní podmínky výsledné sítě, jsou ze sítě odstraněny vrcholy počáteční trojúhelníkové konvexní obálky a trojúhelníky, které tyto vrcholy obsahovaly. Výstupem je tak triangulace pokrývající kompletně celý vstupní obdélníkový obraz.

Decimace

Inkrementální algoritmus může probíhat i opačným směrem. Počáteční trojúhelníková síť je vytvořena ze všech bodů rastru a s každým krokem je odstraněn jeden vrchol. Není zde nutná lokalizace, protože se vždy odstraňuje existující vrchol. Vrcholy jsou zpravidla uchovávány dle nějakého kritéria v prioritní frontě, takže výběr vrcholu je proveden v čase $O(1)$. Komplikací je zde ale vyplnění díry po odstranění vrcholu (viz [Koh09]), zobrazené na následujícím obraze 3.6.



Obraz 3.6: [Koh09] Odstranění vybraného vrcholu a záplatování vzniklé díry.

Ve stávající implementaci projektu se používá právě decimace, protože pro větší množství vrcholů je výsledná triangulace kvalitnější. Inkrementální konstrukce se ukazuje jako vhodnější pro malé triangulace v řádu stovek bodů. Důvodem je ale spíše nízký počet kroků výpočtu, které jsou ke konstrukci třeba, než lepší kvalita obrazu.

3.2.2 Volba důležitých bodů

Klíčovým problémem reprezentace digitálního obrazu triangulací je výběr optimální podmnožiny bodů rastrového obrazu, ze kterého je triangulace tvořena. Cílem je vybrat co nejméně bodů tak, aby se výsledný obraz co nejméně lišil od originálu. Čím více bodů v trojúhelníkové síti bude, tím víc se bude blížit rastrové reprezentaci (jak kvalitou, tak počtem bodů). V důsledku to znamená, že se ztrácí výhody trojúhelníkové sítě jako je např. zachování kvality hran v obraze i pří jeho zvětšení. Navíc bude neúměrně vzrůstat velikost triangulace oproti rastrové reprezentaci kvůli nutnosti uchovávat pozice vrcholů.

Jak tedy vybrat nejdůležitější body? Vybrat optimální množinu je NP-úplný problém. Snažíme se proto množinu jen co nejlépe odhadnout, k čemuž slouží množství navržených heuristik. Nezávisle na typu heuristiky je po celou dobu konstrukce triangulace udržována a aktualizována prioritní fronta s váhami všech vrcholů. V každém kroku je odstraněn vrchol s nejnižší váhou (tedy vrchol jehož odstranění způsobí co nejmenší chybu podle dané heuristiky) a opravena trojúhelníková síť.

Heuristiky lze rozdělit na dvě hlavní kategorie: rastrové a síťové. Rastrové heuristiky počítají důležitost každého bodu pouze podle okolních bodů na rastru. Výhodou je jejich jednoduchost a rychlost, protože je stačí spočítat pouze jednou před konstrukcí triangulace. Tím jsou získány ihned důležitosti všech bodů. Síťové heuristiky oproti tomu vybírají body podle toho, jaká bude výsledná během procesu tvorby triangulace. Jsou tedy pomalejší, protože je nutné opakovaně oblasti triangulovat a přepočítávat vzniklou chybu.

Rastrové heuristiky

- *Random*

  Výběr bodů z rastru je určen čistě náhodně, takže bude mít přibližně rovnoměrné rozdělení po celém obraze. Budou zastoupeny jak hrany obrazu, tak body na plochách obrazu. Protože hrany většinou zabírají proporcionálně mnohem menší část obrazu, bude většina bodů vybrána z méně kontrastních oblastí.

- *Marr-Hildreth*

  Tato heuristika používá pro určení váhy bodu Laplaceův hranový operátor. Operátor slouží k detekci hran, takže body na hranách budou mít mnohem větší váhu.

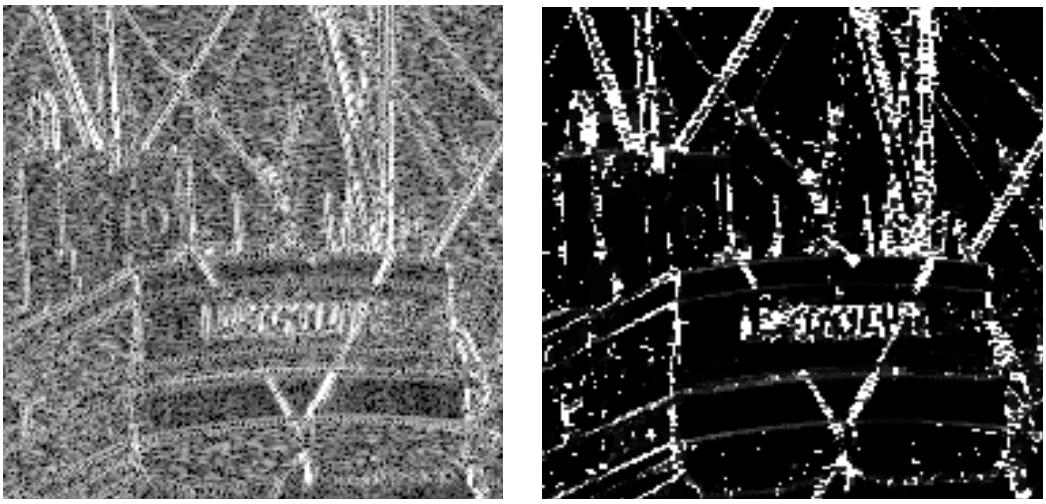
$$v(x,y)=\left|\begin{bmatrix}0 & 1 & 0\\ 1 & -4 & 1\\ 0 & 1 & 0\end{bmatrix} * M(x,y)\right|,$$

  kde $v(x,y)$ je výsledná váha pixelu na pozici x,y a $M(x,y)$ je matice jasů obrazu o velikosti 3x3 se středem na pozici x,y. Problémem heuristiky je přílišný počet bodů s velkou váhou (všechny body na hranách), jak je vidět na obraze 3.7. Pro triangulace s nízkým počtem vrcholů budou všechny vrcholy vytvořené z bodů na hranách, a nezbudou tak žádné vrcholy pro reprezentaci méně kontrastních oblastí.

- *PixSim*

  Metoda určuje váhu bodů na základě počtu podobných bodů v okolí. Bod je podobný, pokud je rozdíl intenzit menší než definovaná tolerance. Množina podobných bodů se dále rozšiřuje v horizontálním a vertikálním směru, dokud je rozdíl intenzit bodů v toleranci. Rozšiřování probíhá ze všech bodů tvořené množiny. Čím je toto okolí

početnější, tím je váha bodu menší. Pro vysokou toleranci je počet bodů s velkou váhou podstatně snížen (viz obraz 3.7). Pro nižší kvalitu triangulace se tak dají očekávat lepší výsledky, než při použití Marr-Hildreth operátoru.



Obraz 3.7: [Koh09] Váhy pixelů vypočtené metodou Marr-Hildreth (obraz vlevo) a metodou PixSim (vpravo) s nastavenou vysokou tolerancí (16).

Síťové heuristiky

Tyto heuristiky jsou aktualizovány během výpočtu triangulace – jsou tedy prováděny opakovaně při každém kroku decimace sítě (ovšem jen pro určitou část obrazu). Oproti rastrovým metodám, které je nutné provést pouze jednou před konstrukcí, trvá kvůli tomu výpočet celé triangulace mnohem déle.

- *DistW*

  Tato heuristika odhaduje důležitost vrcholu v síti podle sousedních vrcholů, se kterými je vrchol spojen hranou. Odhad je určen jako rozdíl jasu vrcholu a váženého průměru jasů jeho sousedů. Váha sousedního vrcholu je vzdálenost od zkoumaného vrcholu. Po odstranění vrcholu musí být heuristika přepočtena i v okolí. Celková váha vrcholu je pak dána vztahem:

$$v(p)=\left| I(p)-\frac{\sum_q I(q).|p-q|}{\sum_q |p-q|} \right| ,$$

  kde $v(p)$ je váha vrcholu na pozici p a $I(p)$ je jas v daném bodě.

- *ErrDist*

  Heuristika je podobná metodě DistW. Rozdíl spočívá v odlišném přepočtu okolí při odstranění vrcholu. K váze vrcholů sousedících s odstraněným vrcholem je přičtena důležitost odstraněného vrcholu. Tím je zajištěna propagace chyby, a tak je zabráněno odstranění mnoha bodů ze stejné oblasti.

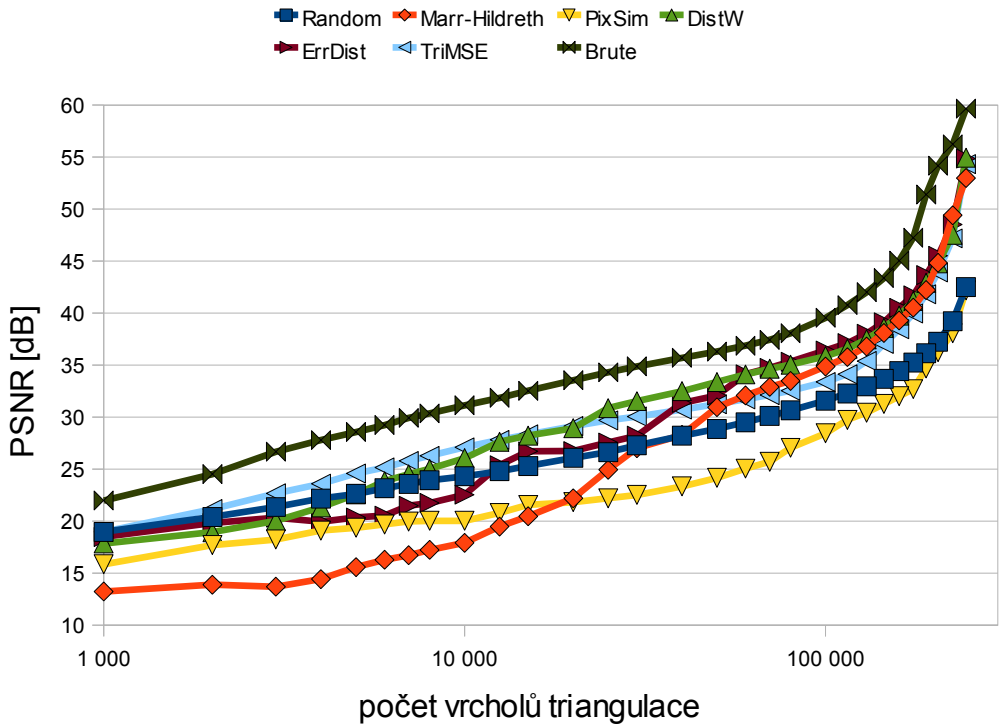
- *Brute*

  Brute metoda odhaduje chybu oblasti vůči původnímu rastrovému obrazu při odstranění každého vrcholu sítě. Pro celou oblast složenou z trojúhelníků, které obsahují zkoumaný vrchol, se počítá střední kvadratická chyba oblasti. Je to součet kvadrátů rozdílů originálních bodů (z rastru) a aproximovaných bodů z trojúhelníkové sítě. Po vyhodnocení pro všechny vrcholy sítě je odstraněn vrchol s nejmenší chybou a následuje další krok. Vyhodnocování chyby bylo prováděno několika odlišnými způsoby (různé použité interpolace trojúhelníků, použití absolutní hodnoty místo kvadrátu rozdílu a podobně), jejichž popis lze nalézt opět v [Koh09].

  Tento přístup je oproti ostatním heuristikám velice pomalý, protože vyhodnocení chyby je časově náročné. Po každém odstranění vrcholu je síť změněna, a tak musí být chyba znovu vypočtena pro okolní oblasti.

Závěr

Nejkvalitnější je dle očekávání heuristika Brute (viz graf 3.1), která je založena na výpočtu chyby aproximací rastru trojúhelníkovou sítí. Metoda je ovšem výpočetně velice náročná. Následují heuristiky DistW a ErrDist, které chybu pouze odhadují. Díky tomu jsou mnohem rychlejší, ale rozdíl v kvalitě oproti Brute metodě se pohybuje okolo 3 dB PSNR, což je značný pokles. Možností by byla akcelerace metody Brute pomocí grafických karet, čímž by bylo pravděpodobně dosaženo dostatečného urychlení výpočtu. Rastrové heuristiky jsou o poznání horší a jejich použití by bylo zdůvodněno pouze jejich nenáročností.



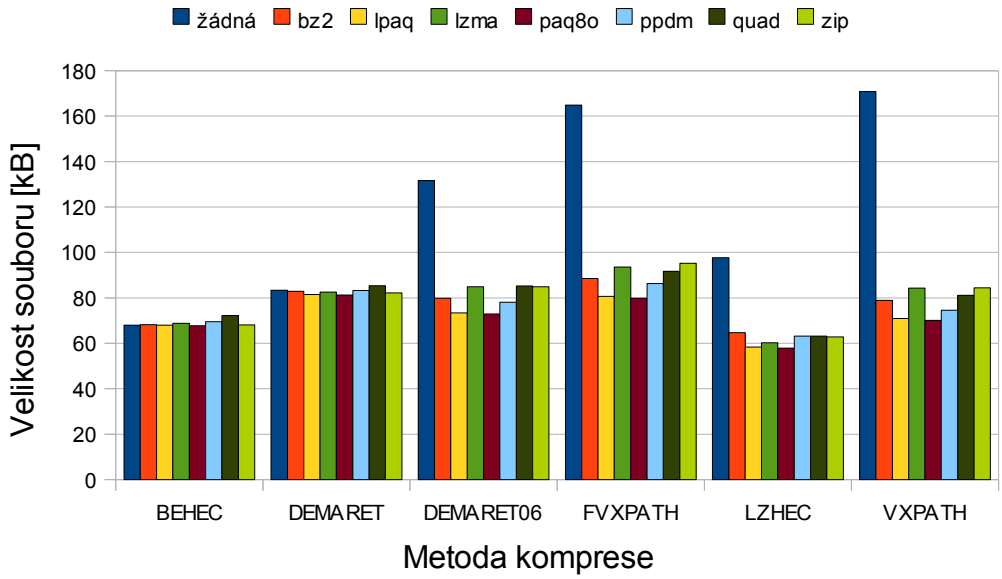
Graf 3.1: Závislost kvality obrazu Lena na počtu vrcholů triangulace pro jednotlivé heuristiky výběru bodů.

Uvedený graf 3.1 zobrazuje kvalitu triangulace v závislosti na počtu vrcholů pro šedotónový obraz Lena. Výsledky pro ostatní testované obrazy jsou obdobné. Je vidět, že metoda Brute je nejvýhodnější pro jakýkoliv počet vrcholů (minimálně o 2 dB PSNR). Při nízkém počtu vrcholů se začíná ztrácet rozdíl ostatních heuristik oproti náhodnému výběru bodů. Nejlepší síťová heuristika Brute je v průměru alespoň o 5 dB kvalitnější než nejlepší rastrová heuristika, což je opravdu výrazný rozdíl. Rastrové heuristiky podávají nekonzistentní výsledky – PixSim je překvapivě horší než náhodný výběr při jakémkoliv počtu vrcholů. Kvalita Mar-Hildreth heuristiky založené na výběru hranových bodů se snižováním počtu vrcholů prudce klesá. Důvodem je přílišný počet bodů na hranách na úkor méně kontrastních oblastí.

3.2.3 Komprese triangulace

K uložení triangulace je zapotřebí uchovávat pozice i hodnoty množiny vrcholů. Rastrová reprezentace oproti tomu potřebuje ukládat pouze hodnoty, protože body jsou umístěny rovnoměrně na mřížce. Z tohoto faktu plyne, že pokud má triangulace dosáhnout podobných kompresních poměrů, je nutné použít sofistikovaný algoritmus pro kompresi pozic vrcholů. I tak bude ovšem komprimovaná triangulace o tuto část větší, a proto také kompresi předchází výše uvedená volba podmnožiny bodů z rastru, které bude triangulace obsahovat.

Většina kompresí je rozdělena na dvě základní fáze: uspořádání dat a kódování. Pro uspořádání jsou aplikovány metody pro Delaunayovu triangulaci, zatímco kódování zajišťují obecné techniky jako třeba LZW či Huffmanovo kódování a jejich kombinace. Podrobnější souhrn kompresních metod je uveden v kapitole 4.1. Zde je pouze uveden závěr plynoucí z testování těchto metod.



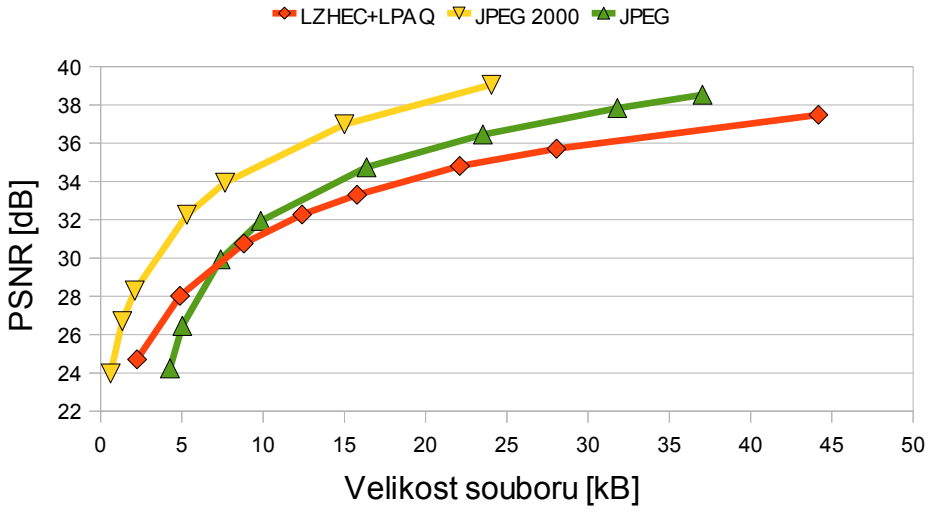
Graf 3.2: Komprese triangulace obrazu Lena o 50 000 vrcholech.

Z grafu 3.2 je patrné, že nejlepší kompresní poměr má kombinace metody LZHEC (varianta průchodu vrcholů po Hilbertově křivce) a paq komprese (jak lpaq, tak paq8o). Následuje BEHEC (opět průchod po Hilbertově křivce). Metoda již používá Huffmanovu kompresi, takže další aplikace jiných kompresních přístupu už velikost výsledného souboru

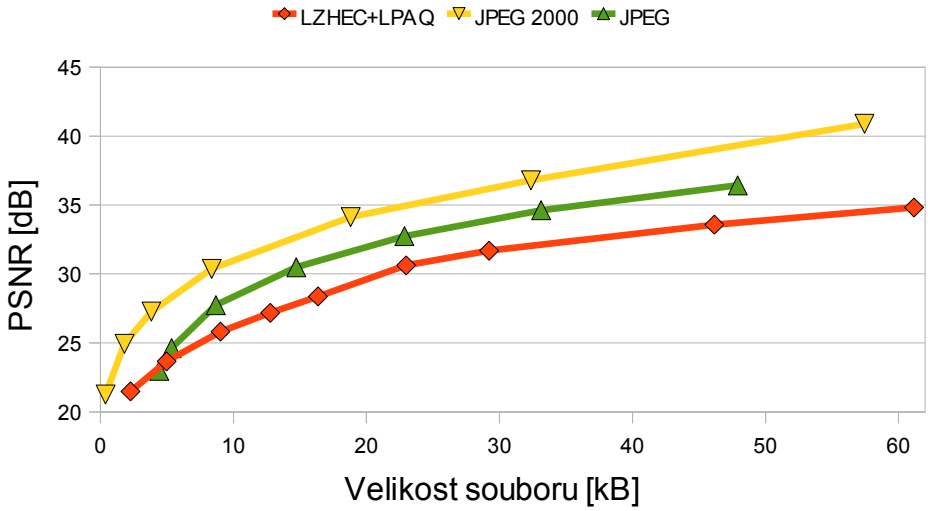
nesnižuje. Metoda VXPATH (Vertex Path) nedosahuje tak dobrých výsledků, jak by se dalo podle její náročnosti očekávat. Rychlejší varianta metody (Fast Vertex Path) je pak ještě horši.

Z hlediska následných kompresí je nejlepší v každém případě lpaq nebo paq8o kodér. Protože paq8 kodér je ale přibližně 35krát pomalejší než lpaq varianta (viz [LPAQ10]), vychází nejvýhodněji lpaq.

Při porovnání se standardními rastrovými formáty je zřejmé, že ani nejlepší nalezená kombinace není schopná konkurovat JPEG 2000 formátu (viz následující grafy 3.3 a 3.4). I samotný formát JPEG je pořád celkově výhodnější. Pro kvalitu pod přibližně 30 dB PSNR je sice triangulace v případě obrazu Lena výhodnější, ale takto nízká kvalita se pro reprezentaci obrazu využívá zřídka. Pro jiné obrazy je triangulace většinou horší do úrovně 25 dB. Nutno ovšem dodat, že v případě triangulací nebyla použita kvantizace hodnot, která by mohla výsledky ještě ovlivnit. Nelze ovšem očekávat zlepšení na úroveň formátu JPEG 2000.



Graf 3.3: Kvalita komprese obrazu Lena vzhledem k velikosti souboru pro jednotlivé komprese. LZHEC+LPAQ označuje nejlepší kompresní metodu triangulace.



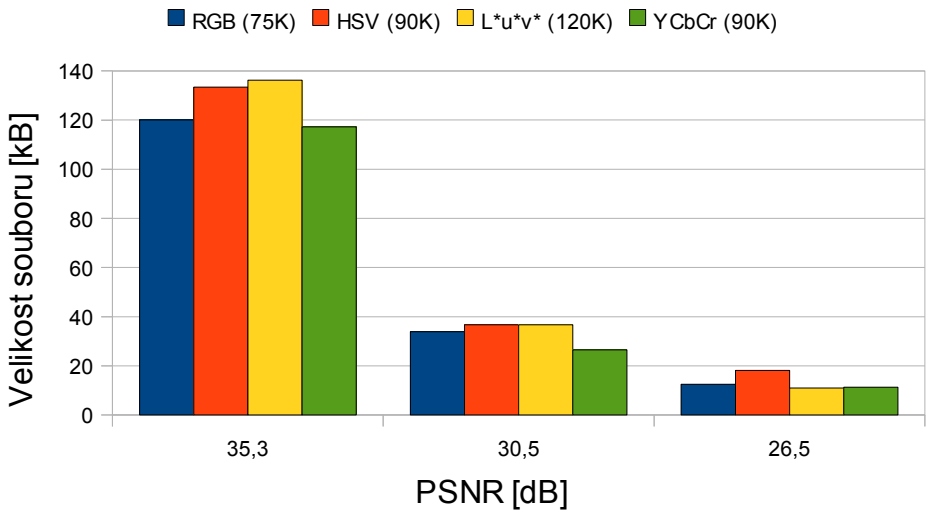
Graf 3.4: Kvalita komprese obrazu Fishingboat vzhledem k velikosti souboru pro jednotlivé komprese. LZHEC+LPAQ označuje nejlepší kompresní metodu triangulace.

3.2.4 Barevný obraz

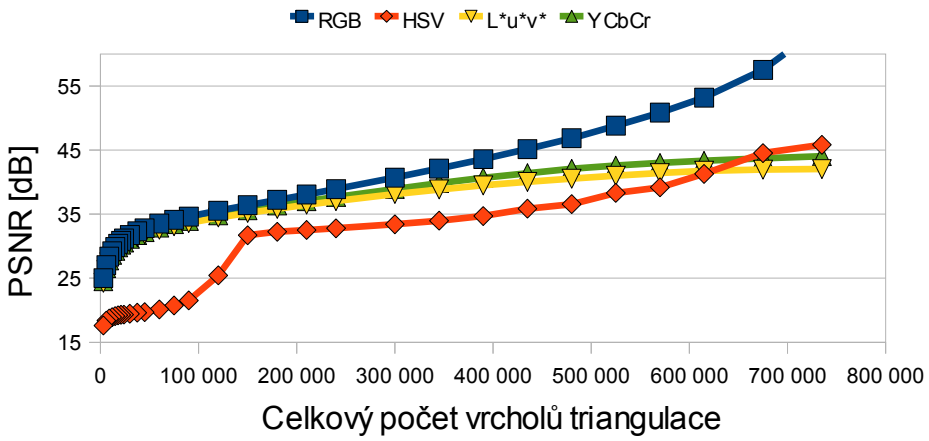
Do této části projektu se pracovalo pouze s jedním barevným kanálem, respektive s odstíny šedi. Pokud budeme uvažovat barevné obrazy s více kanály, je nutné promyslet, jak bude triangulace vytvářena. Triangulace pro jednotlivé kanály jsou totiž odlišné, přičemž míra odlišnosti závisí mimo jiné na zvoleném barevném modelu. Dále jsou také otestovány dvě možnosti, jak tyto triangulace uchovávat – odděleně, nebo spojené do jedné triangulace (tzv. kotriangulace). Barevné systémy, které jsou v projektu testovány již byly popsány v kapitole 2.2. Zde jsou opět jen uvedeny výsledky.

Oddělené triangulace

Nejdříve jsou testovány velikosti komprimovaných souborů pro různé barevné systémy. Z grafu 3.5 je vidět, že nejvýhodnější je použití YCbCr modelu – při stejné kvalitě je komprimovaná triangulace v průměru nejmenší.



Graf 3.5: Kvalita komprese obrazu Fruits oddělenými triangulacemi pro každý kanál v různých barevných modelech.



Graf 3.6: Kvalita triangulovaného obrazu Lena v závislosti na počtu vrcholů pro různé barevné modely.

Zajímavá je závislost kvality obrazu při velkém počtu bodů na daném barevném modelu (graf 3.6). Zde je vidět, že pro hodně kvalitní triangulaci je nejvýhodnější použit RGB model. Rozdíly jsou ale znatelné až pro PSNR větší než 40 dB. Vzhledem k tomu, že triangulace jsou vhodné zejména pro malý počet vrcholů z původního obrazu, nepovažuji tuto vlastnost RGB modelu v kontextu práce za příliš důležitou.

Kotriangulace

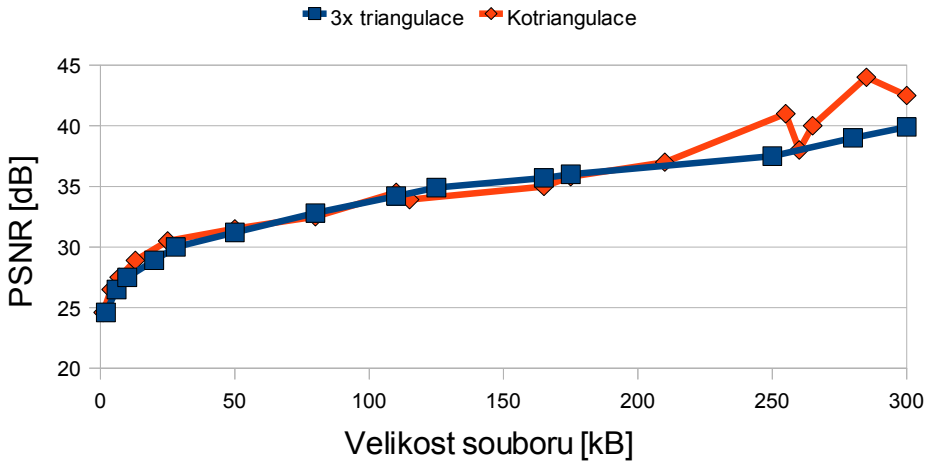
Spojením triangulací všech barevných kanálů vznikne takzvaná kotriangulace. V ideálním případě by měla stejně vrcholů jako triangulace jednoho kanálu. To by znamenalo, že důležité vrcholy jsou ve všech kanálech stejné. V praxi je tento výsledek nereálný. Naopak nejhorší případ – všechny vrcholy jsou unikátní a vrcholy kotriangulace jsou součtem všech vrcholů jednotlivých triangulací – má ke skutečnosti daleko blíže. Výsledky jsou ovšem opět závislé na barevném systému.

Chování kotriangulací pro různé barevné systémy je obdobné jako chování oddělených triangulací (viz graf 3.5). Systém HSV je pro kotriangulace velice nevhodný. Nejlepší výsledky podává systém YCbCr a následuje RGB model.

Závěr

Při porovnání jednotlivých triangulací s kotriangulací (následující graf 3.7) se překvapivě ukazuje, že rozdíl kvality je velice malý a vzhledem k nutnosti spojování triangulací do kotriangulace, je výhodnější spíše ukládat triangulaci každého kanálu zvlášť. Je to způsobeno tím, že při kotriangulaci se ukládají u každého vrcholu jasy všech kanálů. Tedy i ty, které byly při konstrukci kotriangulace pouze interpolovány. Kotriangulace by tedy mohla dosahovat lepších výsledků, pokud by se u vrcholu kódovaly jen hodnoty původního kanálu a ostatní kanály by byly označeny k interpolaci při rekonstrukci.

Zde by dle mého názoru ještě bylo zajímavé vyzkoušet uchování triangulace jasové složky (kanál Y) a provést kotriangulaci složek odstínu (Cb a Cr). Kotriangulace by mohla být hrubější, protože lidské oko je na změnu odstínu méně citlivé než na změnu jasu. Výsledek by pak ale bylo lepší porovnávat podle jiných kritérií, než PSNR nebo SSIM, protože ani jedno neuvažuje žádné váhy jednotlivých kanálů.



Graf 3.7: Závislost kvality oddělených triangulací a kotriangulace na velikosti komprimovaného souboru pro obraz Lena.

3.2.5 Interpolace dat

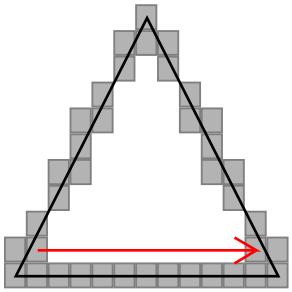
Výsledný obraz rekonstruovaný z triangulace je ovlivněn tím, jak se interpolují pixely v jednotlivých trojúhelnících sítě. Nejjednodušší metodou je konstantní interpolace, kdy celý trojúhelník vyplňuje stejná hodnota získaná jako průměr jasů vrcholů. Lepší možností je bilineární interpolace, kdy jsou hodnoty počítány podle vzdálenosti od jednotlivých vrcholů. Výhodou této dnes standardní interpolace je její jednoduchost a rychlost při poměrně dobré kvalitě. Interpolaci navíc přímo podporují grafické karty.

Kromě těchto základních možností se testovalo několik složitějších přístupů (podrobně vysvětleny v [Jan09]):

- Zinkiewiczova interpolace;
- Bézierova a Coonsova interpolace na trojúhelníkových plátech;
- Natural neighbor interpolace;
- Interpolace pomocí Voronoiova diagramu.

Bilineární interpolace

Bilineární interpolaci je možné provádět několika způsoby. Standardním přístupem je tzv. scanline interpolace. Spočívá v lineární interpolaci pixelů na hranách trojúhelníku. Následně je vnitřek trojúhelníku lineárně interpolován po řádcích na základě hodnot pixelů dvou hran, které se nacházejí na stejném řádku.



Obraz 3.8: Bilineární scanline interpolace - interpolace uvnitř trojúhelníku za použití již interpolovaných pixelů na hranách.

Barycentrické souřadnice

Druhou možností je využití barycentrických souřadnic pro trojúhelník. Místo dvou kartézských souřadnic v rovině (X,Y) jsou použity tři barycentrické souřadnice, které se vztahují k jednotlivým vrcholům trojúhelníku (viz následující obr. 3.9).

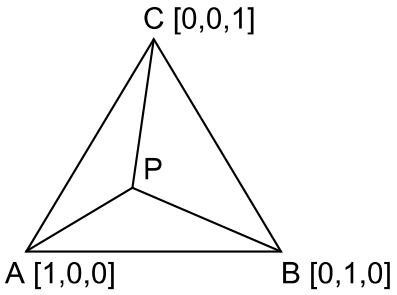
Barycentrické souřadnice α_A, α_B, α_C bodu P vzhledem k trojúhelníku ABC jsou:

$$\alpha_A = \frac{O(PBC)}{O(ABC)}, \quad \alpha_B = \frac{O(PCA)}{O(ABC)}, \quad \alpha_c = \frac{O(PAB)}{O(ABC)},$$

kde $O(XYZ)$ je obsah trojúhelníku s vrcholy XYZ.

Pro barycentrické souřadnice platí, že $\alpha_A + \alpha_B + \alpha_C = 1$. Interpolovaný jas $I(P)$ v daném bodě P, vzhledem k danému trojúhelníku ABC je potom:

$$I(P) = \alpha_A I(A) + \alpha_B I(B) + \alpha_C I(C).$$



Obraz 3.9: Barycentrické souřadnice jednotlivých vrcholů trojúhelníku ABC.

Zienkiewiczova interpolace

Tato interpolace používá k výpočtu hodnot gradientní vektory vrcholů trojúhelníku. Gradientní vektor vychází z normály vrcholu, která je určena jako součet povrchových normál okolních trojúhelníků poměrný k velikosti trojúhelníků. Výsledný dvourozměrný vektor je pak ještě vydělen intenzitou, která je s vrcholem spjatá. Hodnoty v trojúhelníku jsou pak podle těchto gradientů kubicky interpolovány (detailní vysvětlení i implementaci lze najít v [Jan09]).

$$n(v)=\sum_t \frac{A_t . n_t}{A},\ grad=\left(-\frac{n_x}{I(v)},-\frac{n_y}{I(v)}\right)$$

$n(v)$ je normála ve vrcholu v, který patří do trojúhelníků t. A_t je pak obsah trojúhelníku t a n_t je jeho normála. A je celkový obsah všech trojúhelníků, které mají vrchol v. $I(v)$ je intenzita jasu v daném vrcholu.

Bézierova interpolace

Interpolace probíhá na bézierových plochách. Standardně jsou plochy určeny 4x4 koeficienty. V případě triangulace je třeba interpolaci provádět na trojúhelníkových oblastech složených ze čtyř trojúhelníků triangulace. Podrobnější popis opět viz [Jan09].

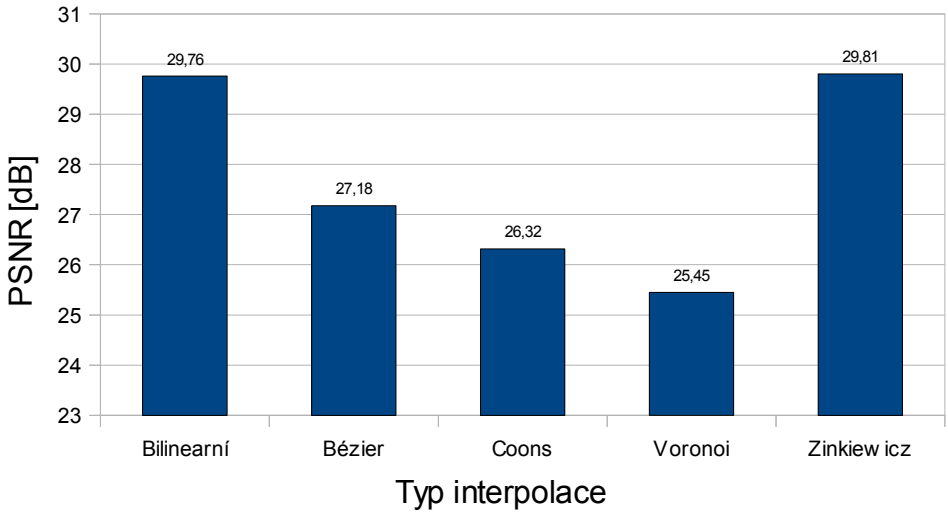
Závěr

V grafu 3.8 jsou uvedeny PSNR hodnoty pro triangulaci obrazu Lena o 5 000 vrcholech. Výsledky jsou obdobné pro ostatní šedotónové i barevné testované obrazy. Ukázalo se, že nejkvalitnější je Zienkiewiczova interpolace, druhá je standardní bilineární interpolace. Z grafu je ovšem vidět, že rozdíl je poměrně nepatrný. Cenou za použití Zienkiewiczovy interpolace je dvojnásobný čas výpočtu, který je navíc porovnávaný oproti softwarové implementaci bilineární interpolace. Čas se ale pohybuje v řádech setin sekund, takže zpomalení je nepatrné.

Interpolace založené na křivkách jsou zde méně vhodné, protože jejich použitím dochází k nežádoucímu vyhlazení hran v obraze. Navíc zavádějí do obrazu určité artefakty, které není jednoduché odstranit (viz [Jan09]). Interpolace jsou k tomu ještě časově náročné (více než 100x pomalejší než bilineární interpolace), což je činí takřka nepoužitelnými.

Uvedené interpolace byly testovány až na hotových triangulacích. To znamená, že nebyly brány v úvahu při samotné konstrukci a nelze tak očekávat přílišné zlepšení kvality. Zahrnutí interpolací do konstrukce (resp. do výpočtu chyby trojúhelníků) by dle mého názoru mohlo mít na kvalitu zásadní vliv. Problémem je ale opět výpočetní složitost – interpolace jsou

prováděny (kromě bilineární interpolace) na více trojúhelnících najednou. Při odstranění bodu, které nastává v každém kroku konstrukce, je tím pádem nutné přepočítat celou okolní oblast.



Graf 3.8: Kvalita triangulace s pěti tisíci vrcholy obrazu Lena podle typu použité interpolace.

3.2.6 Výsledky projektu

Při použití nejlepších metod z jednotlivých kapitol:

- heuristika Brute pro odhad důležitosti vrcholů,
- komprese sítě metodou LZHEC a následovaná lpaq kodérem,
- Zienkiewiczova interpolace triangulace,

reprezentace digitálních obrazů pomocí triangulace zaostává oproti vedoucím rastrovým formátům. U každé části lze ovšem najít ještě další možnosti k vyzkoušení: tvorba triangulace s omezením, nalezení nových heuristik pro odhad důležitých bodů, použití kotriangulací jen pro určité barevné kanály, zahrnutí komplikovanějších interpolací do výpočtu triangulace, vyzkoušení dalších metod komprese trojúhelníkové sítě (zejména pro CDT a DDT triangulace).

Ukazuje se, že triangulace jsou vhodnější pro vysoké komprese (a nízkou kvalitu obrazu). V určitých speciálních případech (obrazy s nízkým šumem a málo hranami) je pak navržená metoda konkurenceschopná formátu JPEG. Pro obecné použití je však nedostačující.

Cílem této práce je navázat na tento projekt a pokusit se navrhnout metodu kombinující geometrickou reprezentaci pomocí triangulací (ideálně z popsaného projektu) a rastrovou reprezentaci v podobě formátu JPEG 2000.

4 Metody komprese

Digitální obraz je možné komprimovat mnoha způsoby. Klíčová je reprezentace, ve které se obraz nachází – například vektorová reprezentace polygony, nebo dvojrozměrná mřížka (dále rastr). Podle dané reprezentace se pak zpravidla vybírá vhodná kompresní metoda. Zde budou uvedeny pouze komprese trojúhelníkových sítí a rastru pixelů, protože ostatní reprezentace nejsou pro tuto práci relevantní.

Zpravidla se dá komprese rozdělit na dvě části – úprava uspořádání dat a kódování těchto dat. Uspořádání je zde nutné vždy, protože se data musí linearizovat do jednorozměrných proudů a dále do souborů. Nejjednodušší uspořádání 2D rastru je uložení po řádkách. Zdaleka však nemusí být nejvhodnější pro kompresi. Čím lépe jsou data uspořádána, tím menší je jejich náhodnost (uspořádání snižuje entropii). Znamená to, že data se dají lépe předvídat, čímž je pak ve fázi kódování dosaženo lepšího kompresního poměru.

4.1 Trojúhelníkové sítě

Trojúhelníkové sítě musí na rozdíl od rastrových reprezentací uchovávat i strukturu sítě (v rastru je implicitně určena 2D mřížkou). Kvůli tomu je potřeba pro jejich uložení více dat, což představuje značnou nevýhodu. Pokud by se v triangulaci nacházely všechny body rastru, byl by nekomprimovaný soubor 3krát větší (o souřadnice vrcholů). Proto se předpokládá, že síť bude složena jen z malé podmnožiny bodů rastru (např. 10% bodů) – viz kapitola 3.2.2 v této práci.

Navíc musí být v obecném případě uchovány kromě vybraných bodů také hrany trojúhelníků. V takovém případě již geometrická reprezentace přestává konkurovat rastrovým přístupům kvůli velkému množství dat. Proto se používají zejména takové trojúhelníkové sítě, kde není nutné hrany uchovávat, nebo stačí pouze jejich malá podmnožina (viz kapitola 3.1 - Triangulace). Metody komprese, které budou v této práci popisovány, fungují většinou právě pro tento typ triangulací (zejména pro Delaunayovu triangulaci).

Metody se zaměřují hlavně na to, jak triangulaci co nejlépe uspořádat do jednorozměrného proudu dat. Většinou dochází v proudu k oddělení jednotlivých složek souřadnic vrcholů a jasových hodnot. V souboru jsou pak data uložena například následovně: x-ové souřadnice vrcholů; y-ové souřadnice vrcholů; jasové hodnoty. Následné kódování se většinou příliš neliší. Převážně se jedná o volbu mezi RLE (Run Length Encoding), slovníkovými metodami (LZW), Huffmanovým kódováním, aritmetickým kódováním a jejich kombinací.

Návrh a testování jednotlivých metod bylo provedeno v projektu GAAV ČR KJB101470701, na který tato práce volně navazuje. Zde je uvedeno shrnutí většiny z těchto metod. Podrobnější popis lze nalézt v [Koh09]. Výsledky metod jsou uvedeny v kapitole 3.2, která je celá věnována zmiňovanému projektu.

4.1.1 Jednoduchá uspořádání

Pokud budou vrcholy uloženy v náhodném pořadí, například po nějakém jednoduchém průchodu sítí, bude uspořádání velice neurčité, což se promítne ve špatné kompresi. Stačí ovšem vrcholy třeba horizontálně (a sekundárně vertikálně) seřadit, a situace bude naprosto odlišná. Takové seřazení si můžeme bez problémů dovolit pro Delaunayovu triangulaci, protože není závislá na topologii.

Při kódování pozic jako rozdílů hodnot po sobě jdoucích vrcholů bude většina rozdílů horizontálních souřadnic při rozumně velké triangulaci (5000 a více bodů) rovna nule. Rozdíly y-ových hodnot pak budou řadou neklesajících posloupností. Komprese takovýchto posloupností je pak velice účinná. Problémem zůstávají hodnoty jasu ve vrcholech – ty jsou v takovémto seřazení (i v původním) značně neuspořádané.

4.1.2 Vertex Path / Fast Vertex Path

Metoda prochází vrcholy trojúhelníkové síť tak, aby minimalizovala rozdíl pozic po sobě jdoucích vrcholů. K tomu používá Minkovského vzdálenost $d(p,q)$, kde p a q jsou vrcholy triangulace.

$$d(p,q)=|(p_x-q_x)|+|(p_y-q_y)|$$

Metoda prochází vrcholy nezávisle na hranách trojúhelníků. Pro nalezení nejbližšího vrcholu je v této základní variantě nutné prohledat celou síť. Průchod je tedy velice pomalý (se složitostí $O(n^2)$), ale dá se očekávat dobrá komprese.

Rychlejší variantou této metody je Fast Vertex Path, která hledá nejbližší vrchol pouze z množiny vrcholů sousedících přes hranu. Může se stát, že budou již všechny sousední vrcholy navštívené, a pak se metoda v následujícím kroku chová stejně jako původní Vertex Path (tj. hledá nejbližší vrchol bez ohledu na topologii trojúhelníkové sítě). V průměru je však metoda mnohem rychlejší. Průchod už ovšem není optimální, a tak je komprese horší.

Ukládání x-ové a y-ové souřadnice pak probíhá odděleně. Hodnoty jasu ve vrcholech jsou kódovány buď stejným způsobem (tj. rozdíl jasu od předešlého), anebo jsou ukládány přímo. Protože rozdíl jasů není průchodem nijak minimalizován, nemusí být ukládání rozdílu výhodnější. Při osmibitovém jasu může být rozsah rozdílu až <-255; 255>, což je dvojnásobný rozsah oproti přímému uložení jasu <0; 255>.

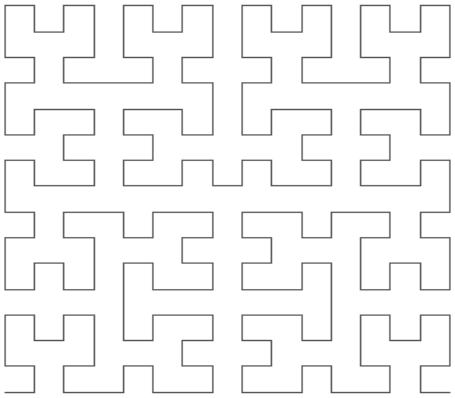
4.1.3 Průchod přes Hilbertovu křivku

Souřadnice trojúhelníkové sítě jsou v tomto případě kódovány jako jednorozměrná vzdálenost na Hilbertově křivce (obr. 4.1). Díky tomu je potřeba ukládat místo dvou souřadnic pouze jednu. Na druhou stranu lze očekávat, že vzdálenosti budou mít větší rozsah. Výhodou přístupu je jeho nízká složitost $O(n)$. Vrcholy jsou totiž procházeny v předem určeném pořadí a pouze jednou.

Komplikací jsou obrazy o jiném než čtvercovém rozměru. V takovém případě budou části křivky mimo obraz. Na těchto částech nebudou ležet žádné vrcholy, ale pořád je nutné počítat s celkovou vzdáleností.

Následná komprese má (v projektu, na který tato práce navazuje – [Koh09]) dvě variace:

- Použití adaptivní bitové šířky pro uchování vzdáleností a následná Huffmanova komprese (varianta BEHEC). Huffmanova komprese je provedena i pro hodnoty jasů.
- Bitová šířka je rozdělena na dvě části: 8 bitů pro vzdálenosti menší než 128, a 16 bitů pro větší vzdálenosti (varianta LZHEC). Předpokládá se, že většina hodnot se bude nacházet v rozmezí 0-128. Hodnoty jasů jsou zachovány na původních osmi bitech. Další komprese je ponechána externím metodám.



Obraz 4.1: Hilbertova křivka stupně 4

4.1.4 Bitová mapa (metoda Korilla)

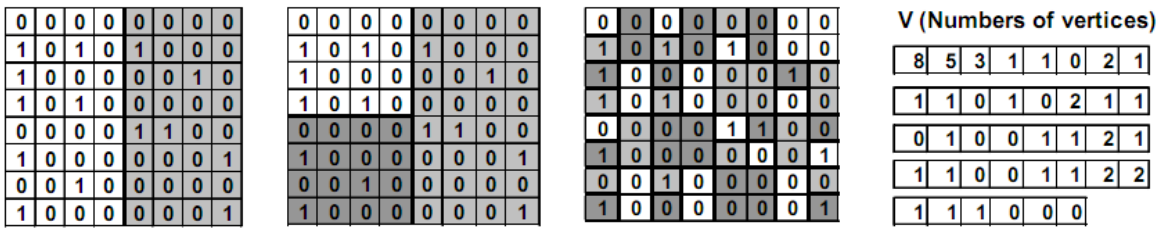
Tato metoda interpretuje triangulaci jako bitovou mapu jejích vrcholů. Vrcholy již z principu triangulace (rastrového obrazu) leží na mřížce. Ke každé pozici na mřížce je přiřazena jedna bitová hodnota, podle toho, zda je na pozici vrchol nebo ne. Data jsou serializovaná do sledu bytů, složených z osmi pozic. Pro triangulace s nízkým počtem vrcholů pak bude většina bytů rovných nule. Následuje aplikace RLE či Huffmanovy komprese.

Pro triangulace s málo vrcholy lze použít účinnější variantu, která neskládá hodnoty do bytů, ale používá Huffmanovu kompresi na různě dlouhé symboly. V tomto případě různě dlouhé řetězce nul.

4.1.5 Demaret

Metoda byla navržena v článku [Dem06]. Pozice vrcholů jsou popsány pomocí bitové mapy – viz předešlá metoda. Následuje rozdělování jednotlivých obdélníkových segmentů bitové mapy, dokud není obdélníková oblast prázdná, nebo má velikost jeden pixel. Při každém rozdělení se na výstup se zapisuje počet pixelů v levé či horní rozdělené oblasti. Rozděluje se vždy oblast s největším počtem vrcholů (znázorněno na obr. 4.2).

Následuje Huffmanova komprese takto uložených počtů vrcholů a Huffmanova komprese jasů v jednotlivých vrcholech.

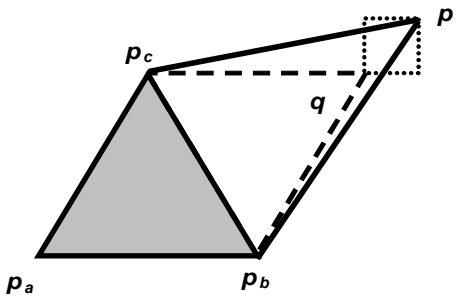


Obraz 4.2: [Koh09] Ukázka kompresní metody z [Dem06]. Postupné dělení oblastí a zápis na výstup. Zleva: první dělení segmentu, druhé dělení segmentu, konečné rozdělení, výstupní pole uchovávající souřadnice vrcholů v popsaném formátu.

4.1.6 Edgebreaker

Edgebreaker je metodou pro kompresi obecných trojúhelníkových sítí, navrženou v [Ros98]. Vhledem k tomu, že není omezena jen na Delaunayovu triangulaci, tak ukládá kromě vrcholů i hrany. Protože je však tato komprese velice kvalitní, byla vyzkoušena i pro Delaunayovu triangulaci. Metoda vyžaduje pouze jeden průchod, a tak má složitost $O(n)$.

Metoda kóduje odděleně topologii sítě a pozice vrcholů. Průchodem sítě uloží topologii jako sled příznaků, které vypovídají o návaznosti dalšího trojúhelníku. Příznaků je pouze pět, a aplikací Huffmanovy komprese je k uložení topologie zapotřebí přibližně 1,5 bitu na trojúhelník. Pozice vrcholů jsou pak odhadovány prediktorem založeným na konstrukci paralelogramu (viz následující obr. 4.3). Ukládá se pak pouze rozdíl mezi skutečnou a předpovězenou hodnotou. Čím je tedy síť pravidelnější, tím lepší kompresi lze očekávat.



Obraz 4.3: [Koh09] Edgebreaker - predikce pozice vrcholu p podle trojúhelníku p_a, p_b, p_c. Vrchol q je odhadnutá pozice.

4.2 Rastrové komprese

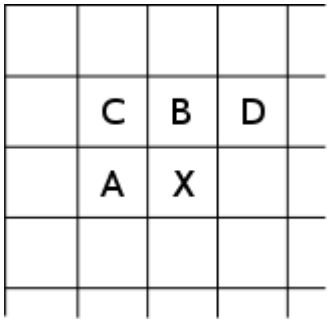
Nejjednodušší metody využívají již existujících kompresí, které byly vytvořeny zpravidla pro lineární (jednorozměrná) data. Rastr je například po řádkách linearizován a pak je aplikována nějaká jednodušší metoda jako třeba RLE či LZW. Tento přístup používá například bezztrátový formát TIF. Linearizací po řádcích je ovšem ztracena informace ve dvourozměrném okolí bodů, která by mohla být s výhodou využita pro účinější kompresi. Proto například JPEG kodek transformuje obraz po dvourozměrných částech (čtvercích), do jiného spektra, ve kterém je linearizace prováděna odlišně.

4.2.1 PNG

PNG je bezztrátový rastrový formát prosazován zejména jako náhrada formátu GIF pro prostředí webu (viz [PNG10]). Důležité je v tomto kontextu zejména uchování průhlednosti a možnost použití palety barev. Kódování je rozloženo do dvou fází – předzpracování a tzv. Deflate komprese.

Předzpracování

Předzpracování slouží lepšímu přizpůsobení dat pro následnou kompresi. Většinou jde o nějakou jednoduchou metodu pro snížení entropie. V tomto případě se metoda pokouší co nejpřesněji předpovědět hodnotu (předzpracování probíhá na úrovni bytů, ne pixelů) v rastru na základě okolí. Protože je nutné data kvůli kompresi linearizovat, je okolí voleno pouze z již zpracovaných částí rastru. Hodnota se pak nahradí rozdílem originální hodnoty a předpovídané hodnoty. Předpovídaná hodnota může být určena několika způsoby (viz obr. 4.4) – jen podle jedné hodnoty (A, B), průměrem z A, B a C, jednou z hodnot A, B nebo C určené podle toho, která má blíže k výrazu A+B-C. Typ předpovědi je pak pro každý řádek v rastru vybrán heuristikou.



Obraz 4.4: [Wiki10b] Pixely (A, B, C, D) používané k předpovědi hodnoty pixelu X.

Deflate komprese

Kompresi zajišťuje metoda Deflate, která je kombinací dvou známých metod – LZ77 kódování a následné Huffmanovy komprese.

- LZ77 je slovníková metoda (jednodušší verze LZW kódování). Jejím cílem je nahradit řetězce dat odkazy na jejich předešlý výskyt. Odkaz samozřejmě nesmí být delší než nahrazovaný řetězec, jinak by metoda neměla smysl. Z toho plyne zejména to, že odkazy mohou sahat jen do určité vzdálenosti od stávajícího výskytu (např. 1kB).

- Huffmanovo kódování pak analyzuje výskyt jednotlivých symbolů v datech a podle četnosti výskytů je kóduje do bitových posloupností, které jsou jednoznačně odlišitelné podle prefixu. Symbol, který se vyskytuje nejčastěji, může pak být reprezentován pouze jedním bitem.

Pro pochopení, proč se používá kombinace metod použijeme jednoduchou ukázku – řetězec o stovkách nul. Každá nula je uchována jako znak a zabírá tedy jeden byte. Pokud by se použilo pouze Huffmanovo kódování (pro symboly o jednom bytu), každá nula by byla reprezentována jedním bitem – dosáhneme tak osmkrát menší velikost. Pokud však předtím aplikujeme LZ77 metodu, dlouhé řetězce nul budou reprezentovány jen několika znaky (odkazem). V tomto případě si to lze zjednodušeně představit jako použití RLE kódování. Velikost výstupu tím omezíme na několik málo bytů.

4.2.2 JPEG

JPEG je v dnešní době nejrozšířenější formát pro uchování digitálního obrazu. JPEG sám o sobě je ztrátový formát. Existují ovšem i bezeztrátové varianty, které ale nejsou součástí původní specifikace tohoto formátu (a lze se s nimi setkat jen velice zřídka). Kódování je opět rozděleno do několika fází – předzpracování, transformace signálu a samotná komprese. Každá fáze zde zavádí do obrazu nějakou informační ztrátu. Podrobné specifikace formátu lze nalézt v [JPE93].

Předzpracování

Barevný obraz je převeden do barevného modelu YCbCr (model je popsán v kapitole 2.2.2). Dochází tak k oddělení jasové části a částí popisujících odstín. Lidské oko je mnohem citlivější na změny jasu než změny odstínu – proto jsou tyto dva kanály podvzorkovány. Na každé čtyři pixely v obraze připadá pouze jedna hodnota kanálu Cb a Cr. Podobný proces probíhá například při kódování televizního signálu. Dochází zde tedy ke ztrátě značného množství informace, které ovšem nelze lidským okem příliš zachytit.

Transformace signálu

Po předzpracování je obraz rozdělen na bloky o velikosti 8x8 pixelů. Pro každý blok je provedena individuálně diskrétní kosinová transformace (pro každý barevný kanál zvlášť) podle následujícího vzorce:

$$G_{u,v} = \alpha(u)\alpha(v)\sum_{x=0}^{7}\sum_{y=0}^{7} p_{x,y}\cos\left[\frac{\pi}{8}(x+0.5)u\right]\cos\left[\frac{\pi}{8}(y+0.5)v\right],$$

kde x a y určují pozici pixelu v bloku, $p_{x,y}$ je hodnota pixelu, u a v jsou výsledné souřadnice v bloku a funkce $\alpha()$ určují váhu koeficientu dle jeho pozice.

Transformace nahradí intenzity pixelů koeficienty různých kosinových signálů. Levý horní koeficient je stejnosměrná složka (kosinová funkce byla v tomto případě konstantní) a reprezentuje průměrnou hodnotu intenzity bloku (viz obr. 4.5). Tento koeficient je pak kódován odděleně od ostatních.

$$\begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ -4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix} \rightarrow \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Obraz 4.5: Nahrazení intenzit pixelů koeficienty kosinových signálů a následné vydělení kvantizační maticí.

Matice koeficientů je pak podělena definovanými kvantizačními koeficienty, které jsou pro každou pozici v matici odlišné. Výpočet sníží většinu koeficientů v pravém dolním na nulu.

Komprese

Koeficienty jsou dále linearizovány po diagonálách – zhruba druhou část signálu tak tvoří souvislý řetězec nul. Tento linearizovaný signál je pak zakódován pomocí metody RLE a následně Huffmanova kódování. Tento krok je tedy velice podobný jako u PNG formátu (kde je důvod této kombinace vysvětlen – viz kapitola 4.2.1).

4.2.3 JPEG 2000

V dnešní době má formát JPEG 2000 asi nejlepší poměr kvality a výkonu (tj. rychlost komprese a dekomprese, velikost výstupu) pro obecný digitální obraz. Protože navrhovaná hybridní metoda má tento kodek z části využívat, bude zde podrobněji vysvětlen. Popsán bude pouze proces kódování (dekódování je inverzní) a výstupní formát dat. Podrobné informace lze nalézt ve specifikaci formátu uvedené v [JPE00].

Proces kódování lze rozložit podobně jako u formátu JPEG na tří hlavní části – předzpracování, transformace a komprese. Ve všech fázích se vyskytují určité rozdíly podle toho, zda má být kódování ztrátové či bezztrátové. Popis se soustředí především na ztrátovou variantu, protože triangulace (a tak pravděpodobně i navrhovaná hybridní metoda) není konkurenceschopná pro vysoké datové toky.

Předzpracování

Ve fázi předzpracování jsou barevné komponenty převedeny do jiného modelu. Při ztrátové verzi je to již zmíněný barevný model YCbCr (viz kapitola 2.2.2), protože chromatické složky obsahují méně informací, a tak je lze lépe komprimovat.

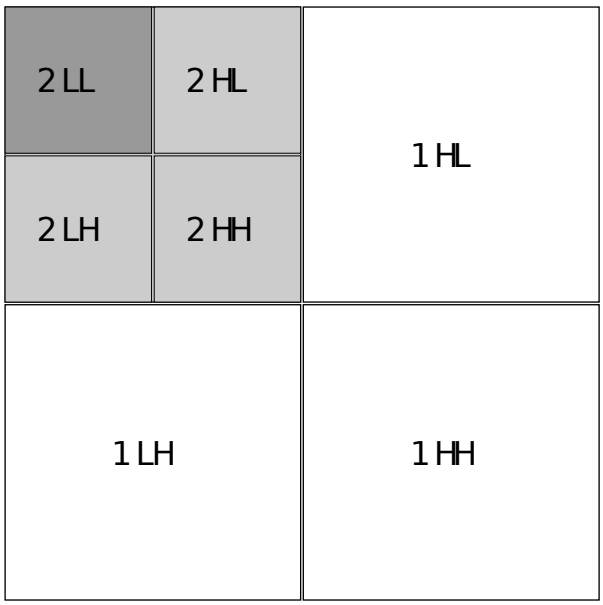
Dále je obraz rozdělen na tzv. dlaždice (tiles). S každou dlaždicí (většinou čtvercového tvaru) je nadále zacházeno individuálně. Díky tomu je pak formát odolnější proti chybám a je také možné použít pro každou dlaždici různý stupeň komprese. Prakticky je tato vlastnost ve velké míře opomíjena, a většinou se tak používá pouze jedna dlaždice, která popisuje celý obraz.

Waveletová transformace

V další fázi jsou obrazová data převedena diskrétní waveletovou transformací na odlišnou reprezentaci. Tato transformace zde bude podrobněji popsána, protože je charakteristická výlučně pro formát JPEG 2000.

Waveletová transformace spočívá v rozdělení signálu na více částí pomocí waveletových funkcí. Signál je rozdělen do určitého počtu frekvenčních pásem, které je pak každé kódováno odděleně. Waveletové funkce jsou v podstatě filtry, které signál upravují. Funkce jsou jednorozměrné – jsou aplikovány pouze na řádek nebo sloupec části obrazu.

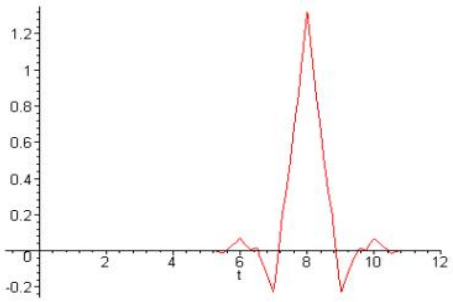
Jedna transformace rozdělí obraz na čtyři části – každá část je kombinací dolní nebo horní pásmové propusti (waveletového filtru) na vodorovné a svislé ose. Další transformace pak vždy probíhají na části obrazu vytvořené dolními propustmi (viz obr. 4.6). Každá část je zmenšena na polovinu, protože popisuje obraz s polovičním pásmem. Počet transformací je volitelný, většinou se jich používá přibližně pět.



Obraz 4.6: Postupná dekompozice obrazu waveletovou transformací na více úrovní (zde pouze dvě) a pásem (vždy čtyři v každé úrovni).

Wavelet

JPEG 2000 používá CDF (Cohen-Daubechies-Feauveau) wavelety. Konkrétní typ se liší podle toho, zda je kódování nastaveno jako ztrátové nebo ne. Pro ztrátové kódování se používá CDF 7/9. Čísla 7 a 9 udávají počet koeficientů filtrů pro dolní a horní propusti. Pro bezeztrátové kódování je užitý wavelet CDF 5/3. Rozdíl spočívá v použití celočíselných koeficientů, které eliminují nutnost zaokrouhlování výsledků. Díky tomu není žádná informace ztracena, a obraz lze zpětně rekonstruovat s naprostou přesností.



Obraz 4.7: [Abo01] Koeficienty waveletového filtru CDF 7/9

Samotná aplikace filtru je předmětem zpracování signálů a spočívá v jednoduché konvoluci vstupního signálu (řádku nebo sloupce obrazu) s daným filtrem.

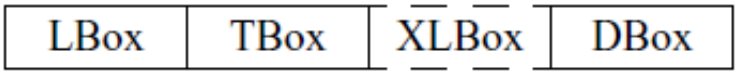
Kódování

Při ztrátovém kódování jsou hodnoty nejprve kvantizovány. Zde nastává další ztráta kvality. Koeficienty každého pásma (např. 1HL) jsou pak specificky uspořádány do obdélníkové struktury. Poté jsou procházeny jednotlivé bitové roviny (první je nejvyšší bit všech koeficientů) a u každého koeficientu je určena jeho důležitost. Ta je pak propagována i do jeho okolí. Podrobnější informace nejsou pro tuto práci důležité, a lze je získat z [JPE00]. Nakonec je určen nutný počet bitových rovin, a každá je zvlášť zakódována aritmetickým kódováním.

Výstupní formát

Protože bude pravděpodobně nutné pracovat s jednotlivými částmi JPEG 2000 souboru, následuje ještě popis tohoto formátu z hlediska uspořádání dat v souboru.

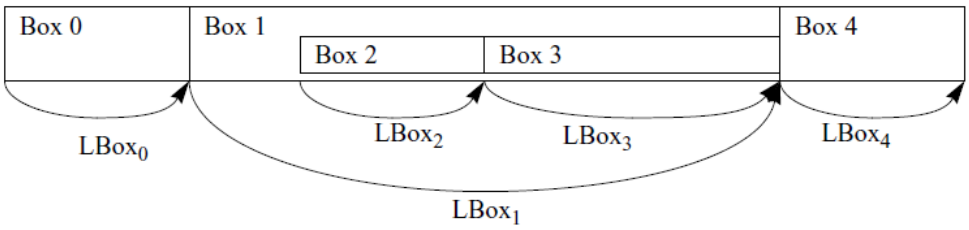
Celý soubor je rozdělen na bloky, které mají danou strukturu:



- LBox – velikost bloku (32 bitů).
- TBox – typ bloku (32 bitů).
- XLBox – velikost bloku (64 bitů).

  Tato hodnota je přítomna pouze pokud je hodnota LBox rovna jedné. Umožňuje popsat delší blok.
- DBox – data bloku.

  Struktura dat je závislá na konkrétním typu bloku. Blok může obsahovat i další bloky. Soubor je tedy složen z hierarchické stromové struktury bloků (viz následující obraz 4.8).



Obraz 4.8: [JPE00] Struktura JPEG 2000 souboru.

Důležitou vlastností je oddělená definice „codestream“ bloku, který obsahuje samotná komprimovaná data. Většina JPEG 2000 kodeků pak umožňuje ukládat pouze tento blok, bez hlaviček specifických pro standardní JP2 formát. To má sice svá úskalí, jako například nepřítomnost informací o barevných komponentách, nicméně tím lze ušetřit až 1kB dat. Navrhovaná hybridní metoda pak může ukládat jednotlivé části obrazu do těchto bloků, a informace o barvě uchovávat pouze jednou globálně.

5 Segmentace obrazu

Protože navrhovaná metoda má využívat jak trojúhelníkovou síť, tak rastrovou reprezentaci obrazových dat (v zastoupení formátu JPEG 2000), musí být vstupní obraz rozdělen na dvě části. Jedna by pak měla být vhodná pro triangulaci a druhá pro rastrovou kompresi JPEG 2000. O tom, které části jsou pro danou reprezentaci výhodnější, se bude pojednávat převážně až v následující kapitole. Zde jsou popsány základní možnosti, jak obraz dělit.

Pod pojmem segmentace obrazu se většinou rozumí rozdělení obrazu na množiny pixelů, které určují jednotlivé segmenty. To je ovšem pouze rozdělení 2D signálu do prostorových oblastí. Zde budou popsány i další možnosti jak obraz dělit – např. převodem do jiných prostorů, aplikací filtrů a podobně.

5.1 Rozdělení na kanály

Jedním z předem daných segmentací je rozdělení podle barevných kanálů v obraze (samozřejmě pouze pro barevné digitální obrazy). Oddělení například červeného kanálu v RGB modelu a jeho reprezentace triangulací nedává příliš smysl. Pokud se ale zaměříme na jiné barevné modely, může být rozdělení smysluplné. YCbCr model odděluje jasovou složky od složek odstínu. Jasová složka přitom obsahuje mnohem více informace. Protože triangulace jsou vhodné především pro obrazy s málo informacemi, bylo by možné je použít pro reprezentaci kanálů s odstínem.

5.2 Transformace obrazu

Transformací signálu je možné převést obraz do jiného prostoru a dále ho rozdělit na dvě části podle nějaké podmínky. Tento přístup používá jak formát JPEG, tak JPEG 2000. Oba však transformují obraz jinak a hlavně na více částí (viz kapitola 4.2.2 a 4.2.3). Triangulace i JPEG 2000 by pak kódovaly obraz stejně velký jako originál, jen s různými informacemi.

5.2.1 Frekvenční filtry

Jednorozměrný signál lze rozdělit na více částí podle frekvence. Typickým příkladem je filtrace signálu horní či dolní propustí. Po filtraci dolní propustí bychom získali obraz s pouze nízkými frekvencemi, který by mohl být vhodný pro kódování triangulací. Zbytek spektra, tedy obraz s rozdílem originálu oproti filtrovanému obrazu, bude představovat vysoko-frekvenční informace, které by mohl lepé komprimovat JPEG 2000.

Aplikace filtru na signál je většinou provedena konvolucí:

$$g(x) = \int_{-\infty}^{\infty} f(t)\,h(x-t)\,dt,$$

kde $g(x)$ je výstupní hodnota v bodě x, $f(t)$ je hodnota vstupního signálu v bodě t a $h(x-t)$ je hodnota filtru v bodě $(x-t)$.

Protože je ale signál dvourozměrný a pohybujeme se diskrétně po pixelech obrazu, jejichž poloha je omezena velikostí obrazu, aplikace filtru probíhá dle následujícího vztahu.

$$g(x,y)=\sum_{k=x-r}^{x+r}\sum_{l=y-r}^{y+r} f(k,l)h(x-k,y-l),$$

kde $g(x,y)$ je výstupní hodnota pixelu na pozici x,y, $f(k,l)$ je hodnota pixelu ve vstupním obraze na pozici k,l a $h(x-k,y-l)$ je hodnota filtru na pozici v parametrech. Poloměr filtru označuje parametr r.

Jako dolní propust pak může sloužit třeba filtr počítající průměrnou hodnotu v oblasti (viz obraz 5.1). Poloměrem filtru je v podstatě určena hraniční frekvence. Existují i sofistikovanější filtry, které velikost filtru během výpočtu mění, nebo nejdou přímo zapsat konvolucí. Aplikace filtru však není příliš odlišná.

$$H=\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

Obraz 5.1: Filtr o velikosti 3x3 počítající průměrnou hodnotu z okolí.

5.2.2 Další možnosti

Mezi další možné transformace patří diskrétní Fourierova transformace, diskrétní kosinová transformace (použitá v JPEG) či waveletová transformace využitá v JPEG 2000. Segmentace po použití těchto transformací se však ukazují jako nevhodné.

- Fourierova transformace nabízí rozdělení obrazu přímo na dvě části – obraz s amplitudou a obraz s fází. V obou obrazech se však vyskytují zejména vysoké frekvence, které jsou pro komprese nevhodné.
- Výstupem diskrétní kosinové transformace je také vysokofrekvenční obraz, podobně jako u Fourierovy transformace. Pokud by byla transformace prováděna na malých blocích (tak jak to provádí JPEG), dal by se obraz rozložit na jednotlivé frekvence. Např. pro bloky 8x8 by byl originální obraz rozložen na 64 osmkrát menších obrazů reprezentujících každý určitou frekvenci. Situace je však podobná. Pouze pro nulovou frekvenci by obraz obsahoval málo vysokofrekvenčních dat a byl by vhodný pro reprezentaci triangulací. Rozdělení by pak nebylo příliš odlišné od použití jednoduchého frekvenčního filtru popsaného výše.
- Waveletová transformace je pro segmentaci také nevýhodná. V tomto případě jsou nízkofrekvenční data reprezentovány jedním obrazem o velmi malých rozměrech. JPEG 2000 kodek většinou vytváří při kompresi přibližně pět úrovní waveletových transformací (viz kapitola 4.2.3). Obraz s nízkými frekvencemi proto bude mít rozměry pouhých několik pixelů, a tak nemá smysl měnit jeho reprezentaci.

5.3 Rozdělení na prostorové oblasti

Tento přístup je zpravidla ztotožňován s pojmem segmentace obrazu. Pixely v obraze jsou podle nějakého pravidla rozděleny do dvou a více množin. Jedním takovým pravidlem může být třeba rozdělení pixelů dle nějakého prahu. Pokud je jas pixelu menší než prahová hodnota, patří pixel do skupiny A, jinak patří do skupiny B. Je jasné, že takto jednoduchý postup bude pro řešení segmentace k ničemu. Ani triangulace, ani JPEG 2000, nepopisuje rozdílně horní nebo dolní rozsah jasů obrazu.

Metody můžeme pro naše potřeby rozdělit na dvě základní skupiny:

1. Množiny jsou uzavřené a tvoří spojité oblasti;
2. Množiny se protínají a nemají jednoznačnou hranici.

Pravděpodobně bude výhodnější uplatnit metody, které tvoří spojité oblasti. Díky spojitosti je možné vytvořit hranice a tím odlišit, která část obrazu bude zpracovávána triangulací nebo JPEG 2000 kompresí. JPEG 2000 ovšem pracuje pouze s obdélníkovými oblastmi, a tak je nutné nějak doplnit okolí oddělené oblasti.

Kromě dvou částí obrazu je navíc nutné uložit i hranice oblastí. To jsou data navíc, která musejí být ve výsledku kompenzována lepším kompresním poměrem oblastí reprezentovaných triangulací. Klíčová bude taky reprezentace hranic – pravděpodobně bude nutné hranice popisovat jinak, než prostou množinou pixelů (například kódováním počáteční pozice a pak pouze směrů, nahrazením parametrickými křivkami, sledem úseček apod.), aby se daly co nejlépe komprimovat.

5.3.1 Region growing

Převážná většina metod pro segmentaci na spojité oblasti je založena na algoritmech pro zvětšování oblastí. Oblast je zpočátku velká pouze jeden pixel a postupně se rozšiřuje do okolí. Metody jsou často závislé na výběru počátečních jedno-pixelových oblastí. Rozšiřování oblasti je omezeno určitou podmínkou, která porovnává vlastnosti celé oblasti vůči zkoumanému pixelu. Pokud jsou tak vybrány jiné počáteční oblasti, budou vlastnosti oblasti při růstu odlišné, a výsledek může dopadnout jinak.

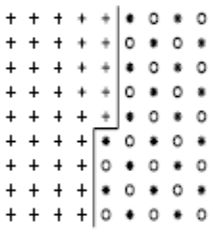
Jednoduchou podmínkou pro růst oblasti je třeba podobnost jasu oblasti. Při rozšiřování oblastí se často narazí na jinou oblast, která má podobný průměrný jas a dojde k jejich spojení. Před výpočtem se většinou provádí kvantizace jasů na méně úrovní, aby nedošlo k rozdělení na velké množství několikapixelových oblastí.

5.3.2 JSEG

Metoda navržená Y. Deng (viz [Deng99]) bere při segmentaci v úvahu nejen jas a barvu, ale také texturu oblasti. Algoritmus je rozdělen do dvou fází:

1. Rozdělení pixelů do tříd. Barvy (odstíny) obrazu jsou kvantizovány. Před kvantizací je provedena filtrace, která z obrazu odstraní šum, výstupem filtrace jsou navíc váhy jednotlivých pixelů určující jak moc je hodnota pixelu důležitá z hlediska lidského vnímání. Hodnoty jsou kvantizovány na základě těchto vah a barev jednotlivých pixelů v barevném prostoru L*u*v*, který popisuje jasovou složku vizuálně rovnoměrně po celém rozsahu (viz kapitola 2.2.3). Výstupem jsou pak třídy pixelů, každá představující jednu kvantizovanou hodnotu.

2. Následuje odhad oblastí na základě rastru s mapou tříd (viz obr 5.2). Odhadují se jak oblasti s pouze jednou třídou, tak oblasti, kde jsou třídy nějakým způsobem promíchány. Algoritmus lze nalézt v [Deng99].



Obraz 5.2: [Deng99] Rozdělení na dvě třídy podle odhadu textury.

Tím, že jsou oblasti rozděleny dle textury je možné odlišit oblasti obrazu, ve kterých se intenzita jasu (či barva) příliš nemění. Takové části by mohly být vhodné pro triangulace, protože bude k popsání oblasti minimum vrcholů. Otázkou zůstává, zda nebude kompresní poměr pro JPEG 2000 v těchto oblastech ještě lepší.

Problémem metody, z pohledu této práce, je především segmentace gradientních oblastí. Dá se očekávat, že triangulace by mohly být úspěšnější zejména na takových částech obrazu. Gradientní oblasti jsou ovšem metodou rozdrobeny do více částí kvůli způsobu kvantizace.

5.3.3 SRM

Metoda navržená R. Nockem ([Nock04]) „Statistical region merging“ je založená na algoritmu region growing, popsaném výše. Tvorba regionů probíhá následovně:

1. Každý pixel tvoří ze začátku jednu oblast.
2. Pro každou dvojici sousedních pixelů (ve čtyř-okolí) je vyhodnocena funkce podobnosti *f()*. Všechny dvojce jsou pak seřazeny podle hodnoty funkce.
3. Pro každý region jsou pak testovány okolní regiony a po případě spojeny, pokud mají průměr, směrodatnou odchylku a další statistické veličiny (vypočtené z hodnot funkce *f()* vyhodnocené v kroku **2**) dostatečně podobné. Podmínka podobnosti je poměrně komplikovaná – detailní vysvětlení lze nalézt v [Nock04].

Kritická je v algoritmu volba funkce podobnosti *f()*. V článku byly testovány dva poměrně jednoduché přístupy: rozdíl absolutních hodnot jasů dvojice pixelů a výpočet gradientu získaného z hranového Sobelova operátoru. Ukazuje se, že gradient je ve většině případů lepší.

Výhodou této metody oproti předchozí je právě lepší segmentace oblastí s plynulými jasovými přechody. Takovéto oblasti budou pravděpodobně pro triangulaci vhodné.

6 Analýza vhodných dat

Návrh hybridní metody začíná analýzou vhodných dat pro jednotlivé přístupy, ve které jsou provedeny srovnání obou kompresí (JPEG 2000 a triangulace z projektu na ZČU) pro typově různé obrazy. Z kapitoly by mělo být patrné, jaké části obrazu by se daly výhodně reprezentovat triangulací.

Triangulace je vytvářena (a komprimována) programem vyvíjeným v rámci projektu na ZČU. Vybrány byly nejúspěšnější přístupy popsány v kapitole 3.2 (tj. heuristika Brute pro výběr vhodné podmnožiny vrcholů, pro kompresi triangulace metoda LZHEC, která prochází vrcholy triangulace po Hilbertově křivce, následovaná lpaq kompresí a Zienkiewiczova interpolace pro vizualizaci triangulace).

JPEG 2000 kompresi zastupuje OpenJPEG kodek. Kvalitu komprese lze nastavit přímo podle indikátoru PSNR, a tak lze snadno porovnávat výsledky obou typů kompresí. Samozřejmě uvedené nastavení slouží pouze jako odhad, není tedy zajištěno, že je nastavená kvalita dodržena. Po kompresi je tedy vždy vypočteno skutečné PSNR.

6.1 JPEG 2000 kodek

Pro testování byl použit kodek OpenJPEG (viz [OJP09]). Jediná další zdarma dostupná knihovna je Jasper. Mezi kodeky nejsou žádné větší rozdíly a OpenJPEG byl vybrán protože se s knihovnou poměrně snadno pracuje, stále se vyvíjí a má okolo sebe aktivní komunitu. Oproti Jasper knihovně je sice pomalejší, ale zase pokrývá větší část standardu JPEG 2000.

Samotná referenční příručka popisující JPEG 2000 má dvanáct rozsáhlých částí. Zmíněné knihovny implementují kompletně pouze první část, která se zabývá kódováním nepohyblivého rastrového obrazu. OpenJPEG knihovna se navíc postupně rozšiřuje o druhou část specifikace, která zmiňuje další možnosti vylepšení kódování.

Implementace

Protože jsem při integraci knihovny narazil na několik problémů, bude v této kapitole popsána práce s knihovnou a nastavení, která byla používána. Knihovna je napsaná v jazyce C, zatímco implementace jednotlivých metod v této práci je v jazyce C#. Z toho důvodu byl vytvořen wrapper v jazyce Visual C++, který slouží jako prostředník mezi těmito jazyky. Samotný wrapper je velice jednoduchý a slouží v podstatě k převodu datových struktur mezi managed pamětí jazyka C# a unmanaged pamětí jazyka C a k volání procedur knihovny.

Jedinou komplikovanější částí je samotná práce s knihovnou. Pro kompresi i dekompresi je zapotřebí postupně volat několik procedur, jejichž pořadí a parametry následují.

1. Před kompresí i dekompresí je nutné nastavit procedury událostí (struktura *opj_event_mgr_t*). Tím je určeno, která procedura se zavolá například při chybě komprese nebo při varování. V běžné praxi většinou tyto události nenastávají. Toto nastavení je nutné, a nelze bez něj pokračovat v kompresi ani dekompresi.
2. Následuje nastavení parametrů komprese (*opj_cparameters_t*) nebo dekomprese (*opj_dparameters_t*). Pro dekompresi je nutné pouze určit formát vstupu (v případě této práce je to buď JP2, nebo J2K – rozdíl viz kapitola 4.2.3). Pro kompresi je nutné nastavit zejména tyto parametry:

- *tcp_distoratio* – určuje požadované výsledné PSNR obrazu v dB. V případě bezztrátové komprese je nutné nastavit tuto hodnotu na nulu.
- *irreversible* – nastavuje ztrátovou kompresi. Obraz je konvertován do YCbCr modelu (pokud byl barevný) a používá se jiný typ waveletu, než při bezztrátové kompresi (viz opět kapitola 4.2.3).
- *tcp_numlayers* – umožňuje nastavit progresivní uložení. Dekódování pak probíhá postupně po jednotlivých vrstvách, což může být výhodné pro internetové prohlížeče. Komprimovaný soubor je pak ale samozřejmě větší. V případě této práce je použita pouze jedna vrstva.
- *tcp_mct* – Indikuje, zda má obraz více kanálů („mct“ odpovídá multiple color transform).

3. Pro kompresi je nutné obrazová data převést do formátu struktury *opj_image_t* (popř. z tohoto formátu při dekompresi). Strukturu lze vytvořit procedurou *opj_image_create*. Ke každému kanálu obrazu je nutné nastavit jeho rozměry, počet bitů na hodnotu a případně podvzorkování. Komponenty obrazu je nutné uložit odděleně – každý má ve struktuře své pole.
4. Dalším krokem je vytvoření struktury *c_info* nebo *d_info*, která sdružuje všechny nastavené informace (v případě komprese nastavený formát výstupu, strukturu uchovávající obraz, strukturu s nastavením komprese a odkaz na samotný enkodér).
5. Následuje už samotná komprese či dekomprese voláním procedury *opj_encode* nebo *opj_decode*. Výstupem po kompresi je pole bytů, po dekompresi popsaná struktura *opj_image_t*.

Výstupní data

Pokud je obraz kódován do J2K formátu, postrádá některé hlavičky, čímž lze uspořit místo. Komprimovaný formát se skládá z jednotlivých bloků (informační, datové a podobně), jak bylo popsáno na konci kapitoly 4.2.3. Na počátku souboru se nacházejí informace o velikosti obrazu a typu komprese. Tato část zabírá i v úsporném J2K formátu přibližně sto bytů. Taková velikost se může zdát nepodstatná, ale pokud budou kódovány z obrazu postupně jednotlivé segmenty, budou těchto hlaviček až stovky. V takovém případě bude mít jejich součet klidně i několik kB, což už může rozhodovat o úspěchu či neúspěchu hybridní metody. Proto byla do implementace některých metod zahrnuto oříznutí této části souboru. Samozřejmě je u těchto metod nutné znát přinejmenším velikost kódované oblasti, aby mohla být při dekompresi obrazu tato část doplněna (typ komprese a zbytek hlavičky bude pro všechny oblasti stejný).

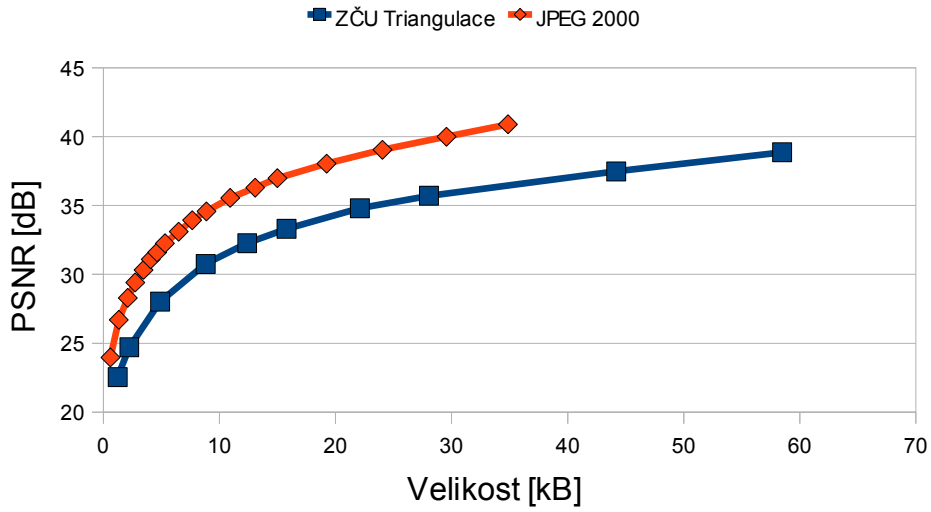
6.2 Standardní sada testovacích obrazů

Obraz 6.1: Standardní testovací obrazy: Lena, Peppers,Baboon, Fishingboat.

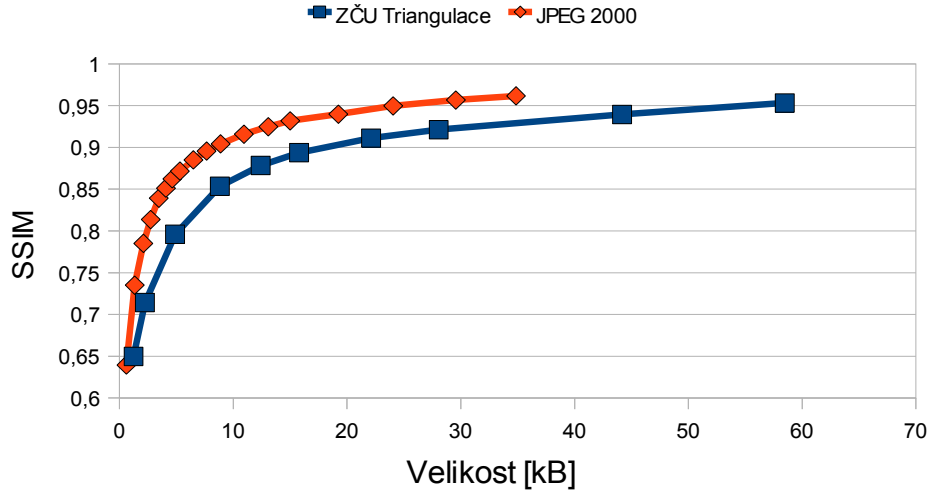
Obraz Lena je kombinací nízkofrekvenčních oblastí pozadí a vysokofrekvenčních oblastí části klobouku. Obraz Peppers má téměř jednobarevné oblasti s ostrými hranami. Baboon je ukázkou čistě vysokofrekvenčního obrazu a Fishingboat obsahuje ostré jednopixelové přechody (lana lodí). Všechny obrazy jsou v odstínech šedi a o velikosti 512x512 pixelů, bez jakékoliv předešlé komprese.

Pro všechny testované obrazy jsou výsledky podobné – JPEG 2000 komprese poskytuje kvalitnější výstup při libovolné velikosti výstupního souboru (viz následující grafy 6.1 a 6.2). Ostatně tento výsledek byl uveden již v závěru projektu, na který tato práce navazuje. Dále jsou uvedeny pouze výsledky pro obraz Lena. Kvalita je měřena jak v PSNR, tak i v SSIM (obě metody jsou popsány v kapitole 2.3). Indikátor SSIM je počítán se standardními koeficienty zvolenými autory metody uvedené v kapitole 2.3.2. Výpočet je stejný pro celou tuto práci.

Kvalita komprese se většinou pohybuje v mezích 20 dB až 40 dB indikátoru PSNR. Při 20 dB je obraz již velice špatně rozeznatelný a při 40 dB je poměrně těžké obraz odlišit od originálu.



Graf 6.1: Srovnání kvality obrazu Lena komprimovaného JPEG 2000 a triangulací podle indikátoru PSNR.



Graf 6.2: Srovnání kvality obrazu Lena komprimovaného JPEG 2000 a triangulací podle indikátoru SSIM.

6.2.1 Degradace kvality

Obě metody vykazují při degradaci kvality odlišné vlastnosti. Na obrazu zkomprimovaném JPEG 2000 jsou viditelné čtvercové artefakty podobné obyčejné JPEG kompresi. Vznikají na hranách obrazu kvůli použité waveletové funkci, která kvůli své složitosti (viz kapitola 4.2.3) potřebuje na popsaní ostrých přechodů více koeficientů.

Triangulace oproti tomu dokáže ostré přechody zachovávat lépe – povšimněte si zejména pravého horního rohu obrazu ve srovnání 6.2. Plochy více ztrácejí detaily, což je způsobeno nedostatečným počtem vrcholů. Tyto vlastnosti jsou ovšem charakteristické ne obecně pro triangulaci, nýbrž pro tuto konkretní metodu výpočtu. Samotná detekce a výběr bodů na hranách je obtížná problematika. Např. použitím nějakého hranového detektoru získáme sice body na hranách, jenže je jich mnohem víc než je nutné, čímž se zhorší výsledná kvalita. Jak vybrat vhodné body v rozumném čase zůstává otázkou.

Obraz 6.2: Výřez obrazu Lena. Vlevo JPEG 2000 komprese (2,1 kB), vpravo Triangulace (2500 vrcholů, 4,8 kB). Oba zkomprimované obrazy mají shodné PSNR 28 dB.

Z těchto odlišností lze usoudit, že triangulací by se daly lépe reprezentovat ostré hrany, zatímco JPEG 2000 komprese postihne zejména detaily obrazu. Kombinace ale přesto nemusí být výhodná – stačí si všimnout velikostí obou porovnávaných komprimovaných obrazů. JPEG 2000 má 2,1 kB zatímco triangulace 4,8 kB. Kompresi triangulace lze pravděpodobně ještě vylepšit, nicméně nedá se již očekávat dvojnásobné zmenšení. Pokud porovnáme obě komprese při stejné velikosti souboru (viz obr. 6.3), je JPEG 2000 výrazně kvalitnější. Pořád si lze ovšem všimnout čtvercových artefaktů na ostrých hranách (opět pravý horní roh). Zmíněná ostrá hrana u triangulace ovšem také nemusí být přesná. Když bude v jednom směru hrana nepatrně posunuta, PNSR této části obrazu může vyjít hůře než u JPEG 2000 komprese.

Obraz 6.3: Obraz Lena. Oba komprimované soubory mají přibližně stejnou velikost 5 kB. JPEG 2000 vlevo má PSNR 32 dB, triangulace 28 dB.

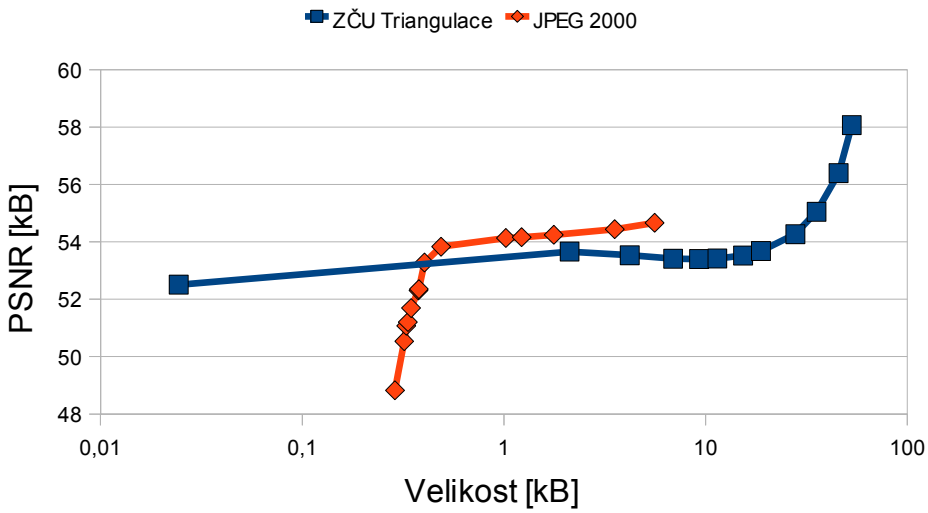
6.3 Speciální obrazy

Vzhledem k tomu, že je potřeba nějak obě reprezentace zkombinovat (pravděpodobně segmentací na regiony), je nutné nalézt nějaký typ obrazu, pro který je triangulace výhodnější. Proto následuje testování kompresí na méně běžných obrazech, jako jsou vektorová data, či obrazy s pouze nízkými frekvencemi.

Jasové přechody

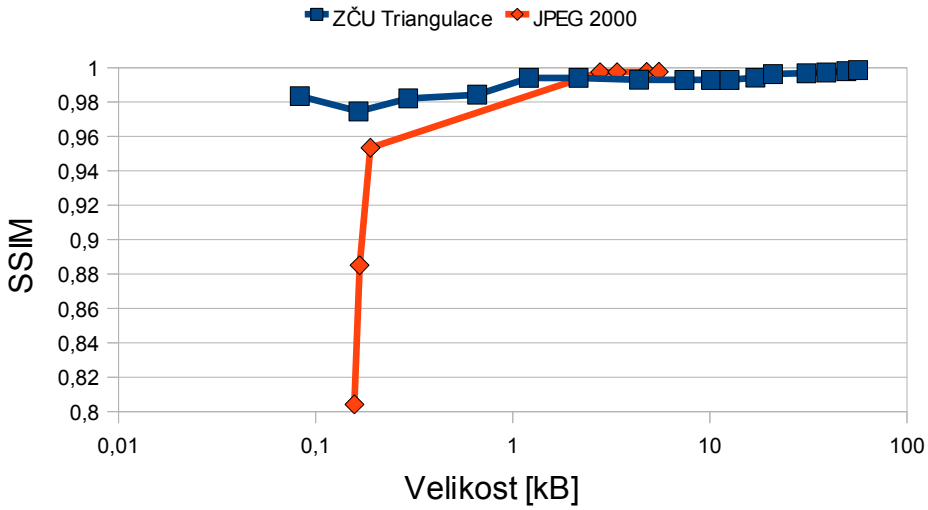
Vzhledem k charakteru těchto dat se dá očekávat, že bude triangulace v těchto případech mnohem výhodnější. Pokud je ovšem cílem práce reprezentovat hybridní metodou obecné obrazy, je tato výhoda diskutabilní.

Obraz 6.4: Vektorové obrazy jasových přechodů.



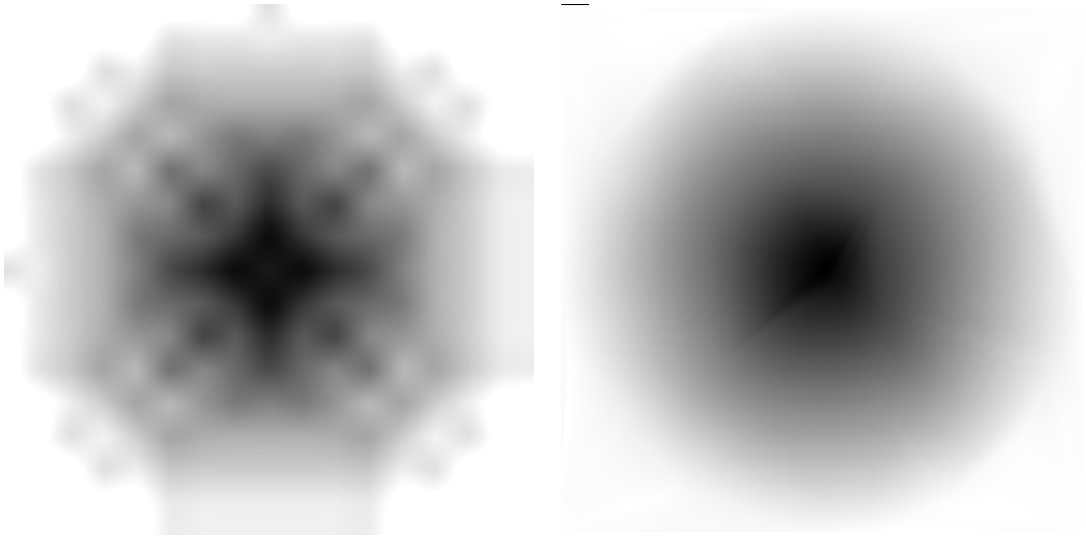
Graf 6.3: Srovnání kvality komprimovaného diagonálního jasového přechodu v závislosti na velikosti podle indikátoru PSNR.

Pro diagonální i horizontální gradient je pod 54 dB PSNR výhodnější triangulace (graf 6.3). Při dalším snižování vrcholů téměř nedochází ke ztrátě kvality. Výsledek je celkem samozřejmý – oba obrazy lze triangulací popsat pouze čtyřmi vrcholy. SSIM indikátor je v tomto případě poměrně nevhodný, protože rozdíly v kvalitě obou obrazů se pohybují v tisícinách.



Graf 6.4: Srovnání kvality komprimovaného radiálního jasového přechodu v závislosti na velikosti podle indikátoru SSIM.

Výsledek pro radiální přechod (graf 6.4) je podobný jako pro diagonální a horizontální gradient – při menších kvalitách výstupu je triangulace menší. Navíc jsou na následujícím obraze 6.5 vidět nedostatky indikátoru PSNR. Při shodném PSNR je výstup JPEG 2000 komprese mnohem horší, než výstup triangulace – viz srovnání níže. U triangulovaného obrazu je navíc vidět nepřesnost výpočtu v podobě černé horizontální úsečky v levém horním rohu. Té by šlo zabránit zvýšením přesnosti v programu triangulace. Právě tato malá část obrazu zásadně ovlivní výsledné PSNR, takže vychází pro oba obrazy stejně. Proto je zde uveden graf pouze SSIM indikátoru.



Obraz 6.5: Radiální gradient při shodném PSNR 30 dB. Vlevo JPEG 2000 komprese, vpravo triangulace.

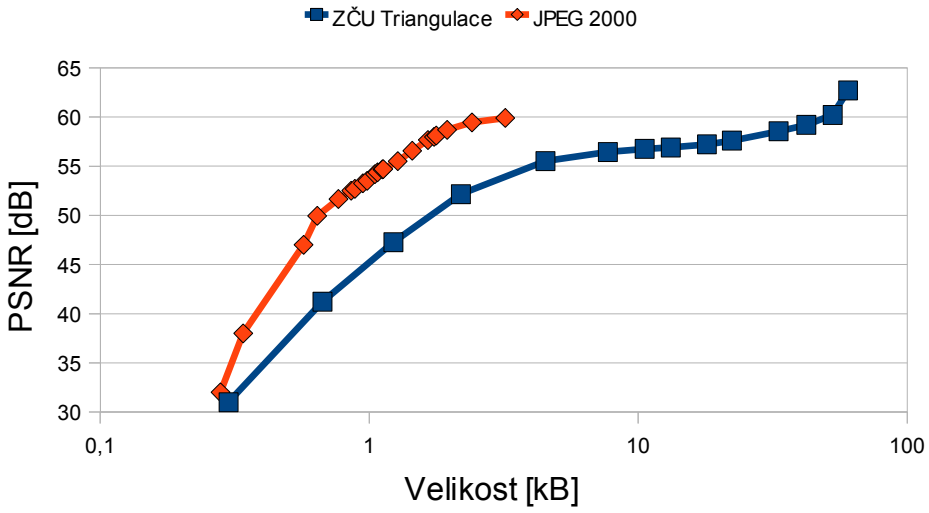
Je vidět, že na těchto obrazech je triangulace vskutku výhodnější. Nutno ovšem podotknout, že obrazy komprimované oběma metodami zabírají velmi málo místa (pod 1 kB). Triangulace tedy může být přibližně až 3krát výhodnější, ale ve výsledku je to rozdíl v řádu desítek bytů. Úspěšně by šla triangulace použít, pokud by obrazy byly velmi rozměrné

(například tisíce pixelů). Ale vzhledem k tomu, že jsou to vektorová data, není důvod, proč nezůstat u jejich původní reprezentace (pokud není obraz převeden z rastru) jako třeba SVG (viz kapitola 2.1). V běžných obrazech, jako jsou fotografie, se takovéto přechody vyskytovat sice mohou, ale je to spíše zřídka a často bývají tyto oblasti navíc zašuměné.

Nízkofrekvenční data

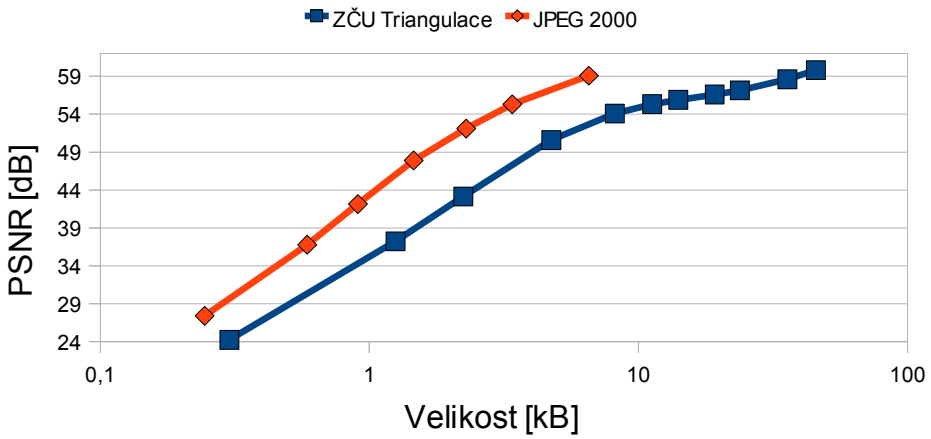
Následuje testování obrazů s pouze nízkofrekvenčními daty neobsahující žádné ostré hrany. Pokud by byla v tomto testu triangulace výhodnější, dala by se použít ke kompresi pozadí či rozmazaných regionů. Podobně jako u gradientních vektorových obrazů bude však tento typ také zabírat velmi málo místa i při poměrně dobré kvalitě. Hlavní částí (z pohledu velikosti komprimovaného souboru) budou pro běžné obrazy opět spíše vysokofrekvenční data.

Obraz 6.6: Nízkofrekvenční obrazy. Vlevo vygenerovaný obraz Clouds, vpravo gaussovsky rozostřený obraz Lena (poloměr rozostření 20 pixelů).



Graf 6.5: Srovnání kvality komprimovaného obrazu Clouds v závislosti na velikosti podle indikátoru PSNR. Pod kvalitu 30 dB PSNR je velikost komprimovaného souboru tak malá, že nemá smysl komprese dále porovnávat.

Pro obraz Clouds je JPEG 2000 stále výhodnější (graf 6.5). Rozdíly jsou při standardní kvalitě (40 dB a níže) nepatrné, protože velikost obou komprimovaných souborů je menší než 1 kB. Graf indikátoru SSIM je obdobný. Pro velmi vysokou kvalitu obrazu (nad 55 dB) přestává triangulace JPEG 2000 kompresi konkurovat. Nad 40 dB je však komprimovaný obraz těžko rozeznatelný od originálu, a tak chování kompresí v této oblasti není tak důležité.



Graf 6.6: Srovnání kvality obrazu Lena rozostřeného gaussovským filtrem o poloměru 20.

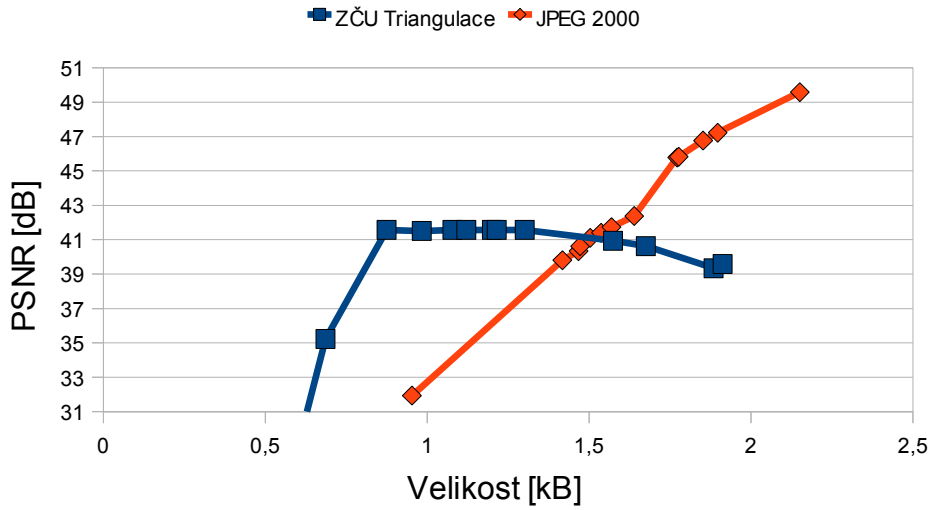
Rozostřený obraz Lena se chová podobně (graf 6.6). JPEG 2000 opět vychází lépe po celém rozsahu PSNR. SSIM indikátor vykazuje stejný průběh.

Z výsledků je patrné, že komprese triangulací je výhodnější pouze pro velmi nízké frekvence (což představují plynulé jasové přechody). Pro ostatní frekvence v obraze je lepší použít JPEG 2000.

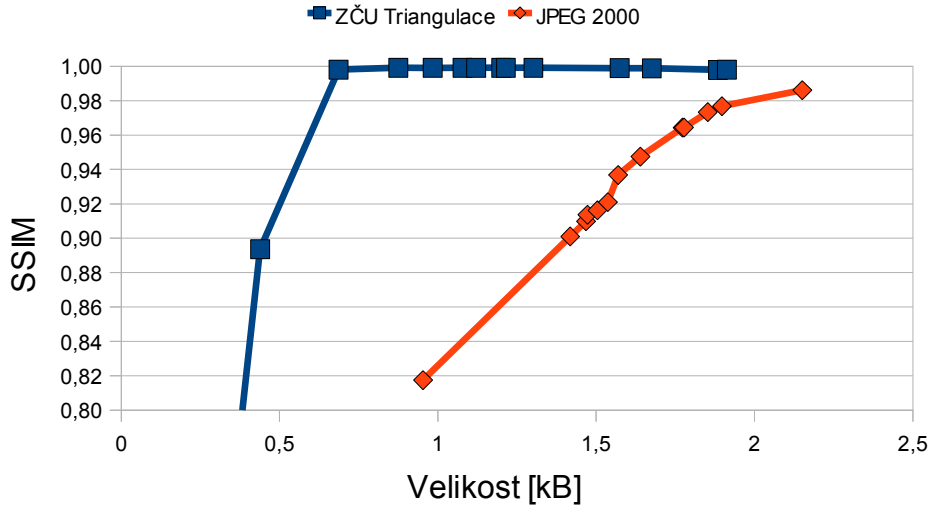
Ostré hrany

Komprese ostrých hran pomocí triangulace je závislá především na správném odhadu vhodných bodů. Při dobrém odhadu by mohla být komprese triangulací lepší. Jsou zde testovány dva obrazy – šachovnice, závislá pouze na výběru bodů a selektivně rozmazaný obraz Lena, ve kterém jsou ponechány ostré hrany. V druhém případě je testován obraz Lena rozmazaný selektivním gaussovským filtrem pro zachování ostrých hran (obr. 6.7).

Obraz 6.7: Obrazy s ostrými hranami. Vlevo šachovnice (černo-bílá bez odstínu šedi), vpravo obraz Lena rozmazaný selektivním gaussovským filtrem o poloměru 10.



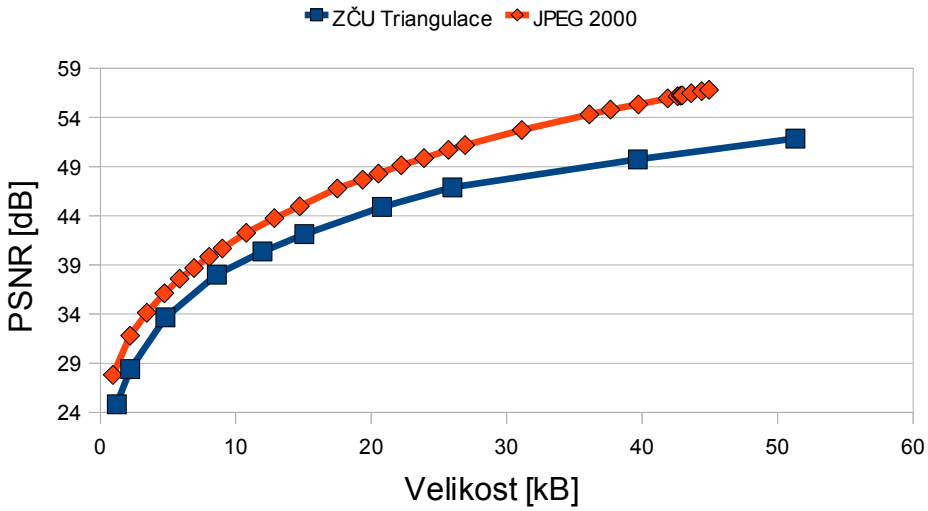
Graf 6.7: Srovnání kvality obrazu šachovnice z závislosti na velikosti souboru podle indikátoru PSNR.



Graf 6.8: Srovnání kvality obrazu šachovnice z závislosti na velikosti souboru podle indikátoru SSIM.

V případě šachovnice je pro malé velikosti souboru triangulace lepší (grafy 6.7 a 6.8). Je to dáno především vektorovým typem dat, pro který je triangulace rozhodně vhodná. Zajímavé je srovnání indikátoru PSNR a SSIM. Zatímco PSNR u triangulace je při větších souborech horší než u JPEG 2000, SSIM hodnota triangulace je lepší přes celý rozsah grafu. To je způsobeno zejména artefakty vytvořenými při malých triangulacích. Protože pro reprezentaci ostrých hran je nutné mít několik vrcholů triangulace přímo u sebe, je pravděpodobnější, že vznikne nepřesnost při testování Delaunayovy podmínky a dojde k prohození hran, které má zásadní vliv na kvalitu (viz obr. 6.8). Vrcholy jsou navíc ve stávající implementaci rozmístěny na mřížce, a tak se může přímo na opsané kružnici vyskytovat jeden další vrchol. Tento nedostatek lze eliminovat drobným náhodným posunem jednotlivých vrcholů triangulace. Z pohledu PSNR je takový artefakt velice zásadní. SSIM oproti tomu určuje podobností jednotlivých oblastí a tak je vidět, že kvalita JPEG 2000 obrazu zde bude nižší.

Obraz 6.8: Komprese šachovnice. Vlevo triangulace s PSNR 31 dB, SSIM 0,99; vpravo JPEG 2000 s PSNR 32 dB a SSIM 0,81.



Graf 6.9: Srovnání kvality obrazu Lena rozostřeného selektivním gaussovským filtrem pro zachování ostrých hran (poloměr masky 10).

U rozmazaného obrazu Lena se zachováním poměrně ostrých hran se při nižších hodnotách PSNR velikosti výstupu obou kompresí přibližují, nicméně JPEG 2000 je opět lepší (graf 6.9). Rozsah grafu je ale od 24 dB do 59 dB a minimální rozdíl v kvalitě je při stejné velikosti souboru 3 dB. Průběh SSIM indikátoru je obdobný. Situace je v tomto případě tedy lepší než při kompresi originálního obrazu Lena, kde byl rozdíl v kvalitě přibližně 5 dB PSNR.

6.4 Shrnutí výsledků

- Pro standardní testované obrazy Lena, Baboon, Fishingboat, Peppers není triangulace (ani pří výběru nejlepších metod) pro JPEG 2000 konkurenci.
- JPEG 2000 je výhodnější i pro nízkofrekvenční data, kde není přítomen téměř žádný šum.
- Ukazuje se, že triangulace dokáže lépe reprezentovat pouze ostré hrany v obraze a plynulé jasové přechody. Problémem je ale výběr vhodných bodů na hranách, protože občas dochází k nesplnění Delaunayovy podmínky v triangulaci, a následnou opravou sítě dojde k poškození hrany. Gradientní přechody se zase v běžných obrazech (fotografiích) vyskytují spíše okrajově (např. část oblohy při fotce krajiny). Navíc tyto oblasti po kompresi zabírají velmi malou část výstupu, a tak i když by se zde dalo ušetřit, nebude úspora moc znatelná.

Protože výsledky nejsou pravděpodobně dostačující k vytvoření úspěšné hybridní metody, byla část práce věnována dalším pokusům s triangulacemi. Vzhledem k obsáhlosti celého projektu s triangulacemi jsem se zaměřil především na úpravu již existujícího postupu vyvinutého v rámci projektu GAAV ČR KJB101470701 zde na univerzitě.

7 Úpravy triangulace

V této kapitole jsou uvedeny úpravy tvorby triangulace, které by teoreticky mohly výsledky uvedené v předešlé analýze vhodných obrazových dat vylepšit a to jak kvalitativně při stejném počtu vrcholů, tak i zvýšením kompresního poměru. Úpravy jsou zaměřeny především na triangulace s nízkým počtem vrcholů, protože v této oblasti nejsou rozdíly mezi triangulací a rastrovou kompresí tak výrazné.

7.1 Předzpracování

Standardně nepoužívá triangulační algoritmus z projektu na ZČU žádné předzpracovaní. Počáteční trojúhelníková síť vytvořená ze vstupního obrazu obsahuje všechny pixely obrazu a váhy vrcholů jsou počítány na základě výpočtu chyby interpolovaných trojúhelníků oproti originálnímu obrazu (podrobněji popsáno v kapitole 3.2.2, konkrétně metoda „Brute“). Pro triangulace s nízkým počtem vrcholů má však metoda jisté nedostatky – v takovém případě jde totiž zejména o reprezentaci hran v obraze. Podobná úvaha byla uvedena již v projektu na který tato práce navazuje – např. použití hranových operátorů pro určení důležitosti jednotlivých vrcholů. Tyto přístupy byly ovšem kvalitativně mnohem horší než stávající metoda „Brute“. Nebylo však vyzkoušeno ještě několik možností, které by mohly kvalitu pro malé triangulace ovlivnit.

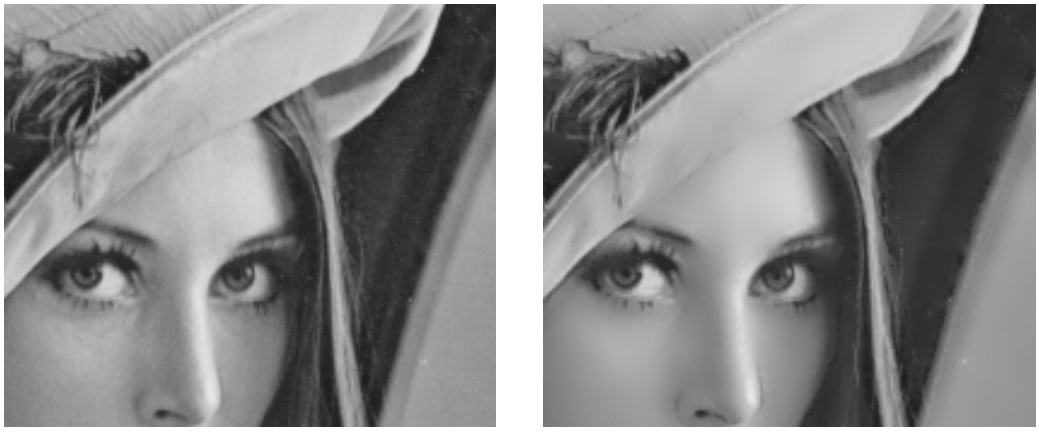
7.1.1 Redukce šumu

Pokud je v obraze hodně šumu, může se často stát, že chyba při odstranění vrcholu ze zašuměné oblasti bude větší, než chyba při odstranění vrcholu na hraně. Z toho důvodu jsem se pokusil redukovat v obraze šum, ale při zachování ostrých hran.

Jednoduché filtry na rozmazání obrazu sice šum zredukují, ale rozmažou nevybíravě i hrany. Proto jsem navrhl vlastní filtr:

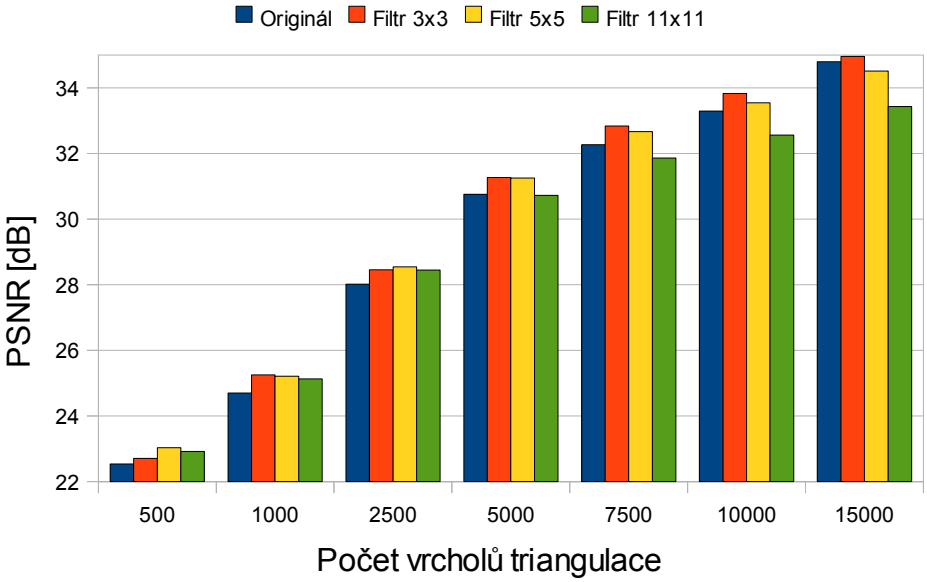
$I(x,y) = D\,I(x,y) + (1-D)\,O(x,y)$, kde $D = k\,|O(x,y) - I(x,y)|$.

$I(x,y)$ je jas v bodě na pozici x, y , $O(x,y)$ je průměrný jas čtvercového okolí. D je tak rozdíl jasu bodu od průměru okolí. Čím je rozdíl větší, tím je větší pravděpodobnost, že bod je na hraně.

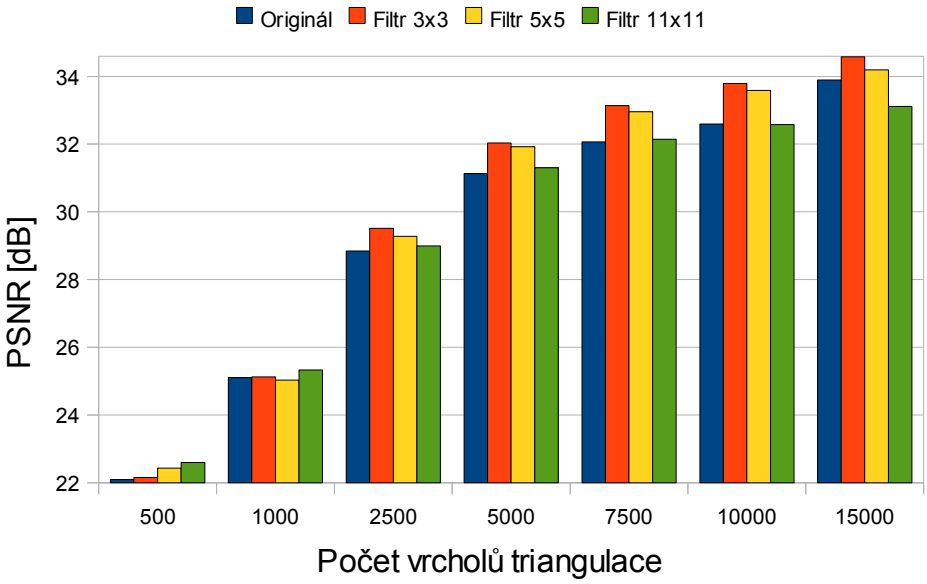


Obraz 7.1: Filtr pro redukci šumu (detail obrazu Lena). Vlevo originální obraz, vpravo filtrovaný obraz s velikostí masky 11x11.

Filtr byl testován pro velikosti masky 3x3 až 11x11. Z filtrovaného obrazu byla vytvořena triangulace o 500 až 15 000 vrcholech, rekonstruován komprimovaný obraz a porovnán s originálem (tj. nefiltrovaným původním obrazem).



Graf 7.1: Vliv filtrace šumu na kvalitu výstupu pro obraz Lena.



Graf 7.2: Vliv filtrace šumu na kvalitu výstupu pro obraz Peppers.

Pro triangulace s 1 000 a více vrcholy je filtrace s maskou 3x3 lepší než přímá triangulace originálu (viz grafy 7.1 a 7.2). Přínos v kvalitě není ovšem příliš patrný (nárůst PSNR průměrně 0,4 dB) a tak je otázkou zda má smysl tímto předzpracováním dále komplikovat algoritmus komprese. Ostatní velikosti masky zanášejí do obrazy pravděpodobně již příliš velkou chybu, a tak kvalita oproti filtru 3x3 klesá. Úspěch filtrace je také značně závislý na vstupním obraze. Např. pro obraz Baboon nemá filtrace praktický žádný efekt, protože je obraz složen téměř ze samých hran.

7.1.2 Váhy vrcholů

Další možností předzpracování je stanovení důležitosti jednotlivých vrcholů na základě nějaké heuristiky. Tomuto tématu se již několik pokusů v původním projektu věnovalo – viz rastrové heuristiky v kapitole 3.2.2. Rastrové heuristiky vycházejí z předpokladu, že je potřeba více vrcholů na kontrastních oblastech obrazu (hranách). Tento předpoklad je rozhodně platný, jenže při detekci je takových kontrastních bodů nalezeno velmi mnoho – zpravidla více, než má být požadovaný počet vrcholů triangulace. Aby mohla být taková heuristika úspěšná, je nutné provést ještě výběr podmnožiny z těchto bodů. Dále je nutné, aby nebyly na hranách vrcholy všechny, což vyplývá z povahy trojúhelníkové sítě.

Navrhovaná heuristika je rozdělena na dvě části:

1. Detekce důležitých bodů pomocí hranového operátoru,
2. Filtrace pro snížení počtu důležitých bodů.

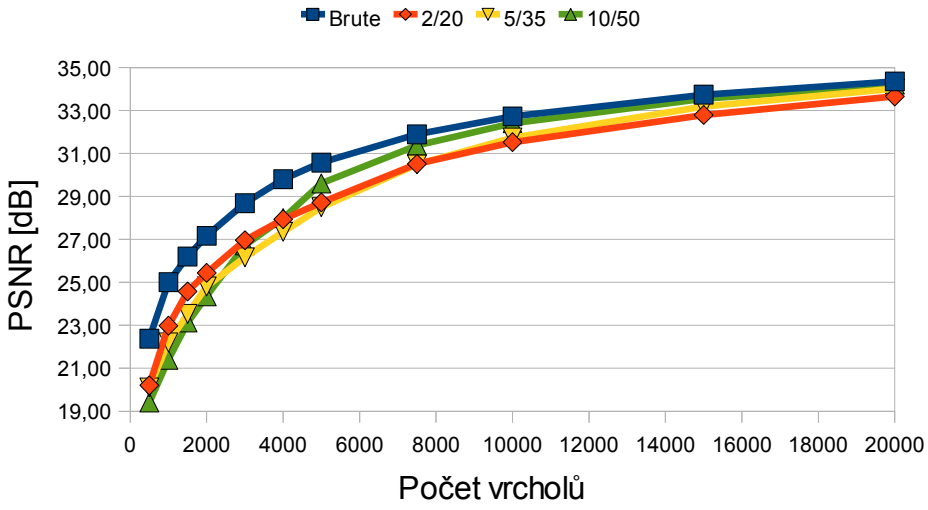
Pro reprezentaci hrany potřebuje triangulace alespoň čtyři vrcholy. Dva vrcholy na jedné polorovině a dva na druhé. Při detekci hran běžnými hranovými operátory není ale směr přechodů jasů (hrany) nijak rozlišen. Proto jsou při detekci použity dva směrové hranové operátory (vertikální a horizontální). Výstupem první fáze heuristiky je tak gradient každého pixelu.

Protože detekovaných bodů je příliš mnoho, následuje filtrace. Obraz (gradienty pixelů z první fáze) je procházen po čtvercových oblastech. V každé oblasti je pak vybrán pouze jeden pixel s podobným gradientem (směrem růstu jasu). Důsledkem je snížení počtu důležitých bodů, při zanechání pixelů na obou stranách hrany.

Obraz 7.2: Detekce důležitých bodů. Zleva: původní část obrazu Lena, detekce gradientu, filtrace důležitých bodů.

Výsledky heuristiky

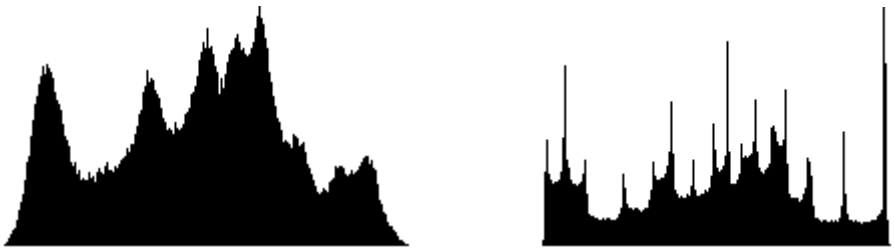
Z následujícího grafu 7.3 je vidět, že přidáním heuristiky se výsledná kvalita oproti standardní metodě Brute zhorší. Byly testovány různé velikosti filtračních masek a prahy podobnosti jednotlivých gradientů (2/20 odpovídá filtru o poloměru 2 a prahu podobnosti 20). Ze všech případů jsou uvedeny pouze tři charakteristická nastavení. Neúspěch metody je pravděpodobně zapříčiněn vytvořením nevhodných hran v Delaunayově triangulaci, což nelze heuristikou nijak ovlivnit. Je pravděpodobné, že prostá volba důležitosti jednotlivých pixelů s použitím jakékoliv rastrové heuristiky nebude odpovídat volbě vhodných bodů pro Delaunayovu triangulaci.



Graf 7.3: Kvalita obrazu Lena v závislosti na použité heuristice. Brute je původní metoda, 2/20 heuristika rozšířená o filtr o poloměru 2 s prahem podobnosti jasu 20. Heuristiky s filtry 5/35 a 10/50 jsou obdobné.

7.2 Kvantizace

V původním projektu nebyla využita kvantizace jasů ve vrcholech při výpočtu triangulace. Jejím zavedením by se dala úspěšně snížit výstupní velikost. Pokud bude kvantizace probíhat již při výpočtu triangulace, nemusí vzniknout přílišná chyba oproti osmibitovému originálnímu jasu. Zde lze s výhodou využít vlastností triangulace, protože kvantizovány jsou jenom hodnoty ve vrcholech. Interpolace uvnitř trojúhelníků pracuje s celým rozsahem a tak ve výstupním obraze bude stejný rozsah jasů jako v originále. Jen bude odlišně rozložen (obr. 7.3).



Obraz 7.3: Histogramy šedotónového obrazu Lena. Vlevo originál, vpravo triangulace se čtyřbitovou kvantizací.

Jakou úsporu lze očekávat? Zhruba dvě třetiny výstupu zabírají pozice vrcholů. Na informace o jasu tedy zbývá přibližně třetina souboru. Při kvantizaci jasu z osmi bitů na čtyři by mohl být výsledný soubor zmenšen přibližně o šestinu. V úvaze ovšem není zohledněna následná komprese celého souboru, která pravděpodobně úsporu ještě zmenší.

Je možné volit několik způsobů jak hodnoty kvantizovat:

- *Rovnoměrně po celém rozsahu*

  Z každé hodnoty je vybrána jen horní část bitů o požadovaném počtu. Nevýhodou přístupu je nezávislost na vstupních datech. Pokud by měl celý obraz jasový rozsah

od 0 do 128 (tedy pouze poloviční datovou šířku), využila by se na něj pouze polovina kvantizovaného rozsahu.

- *Dynamicky*

  Kvantizované hodnoty jsou vybrány v závislosti na vstupním obraze. Existuje mnoho možností jak hodnoty volit (viz [Wiki09d]), nejčastěji používanou metodou je tzv. Median cut navržený v [Heck82]. Výhodou metod je lepší přizpůsobení jasům přítomných v obraze a tak i lepší kvalita výstupu.

7.2.1 Median cut

Algoritmus je popsán pouze pro stupně šedi. Při více barevných kanálů je postup odlišný (viz [Heck82]). Algoritmus je složen z několika kroků:

1. Konstrukce histogramu, zjištění minima a maxima jasů. Tvorba počáteční množiny obsahující celý rozsah přítomných jasů.
2. Nalezení středního bodu množiny, který rozděluje rozsah na dvě stejně početné (přibližně) množiny.
3. Rozdělení na dvě množiny a výpočet průměrného jasu v množinách.
4. Výběr množiny s největším počtem bodů (nebo největší chybou) a volání kroku **2**, dokud není dosažena požadovaná bitová hloubka kvantizace.

Pro každý pixel je tímto postupem určena množina, do které dle jasu patří. Všechny pixely patřící do dané množiny jsou pak nahrazeny vypočteným průměrným jasem množiny.

Kvalita výstupního obrazu

Nejdříve je uvedeno srovnání statické a dynamické kvantizace pro nízké bitové hloubky. Následuje porovnání kvality kvantizovaného a nekvantizovaného komprimovaného obrazu při různých bitových hloubkách kvantizace a nakonec porovnání velikosti výstupů.

Obraz 7.4: Kvantizace na tři bity na pixel. Vlevo rovnoměrná kvantizace PSNR 26,3 dB, vpravo Median cut metoda PSNR 28,6 dB.

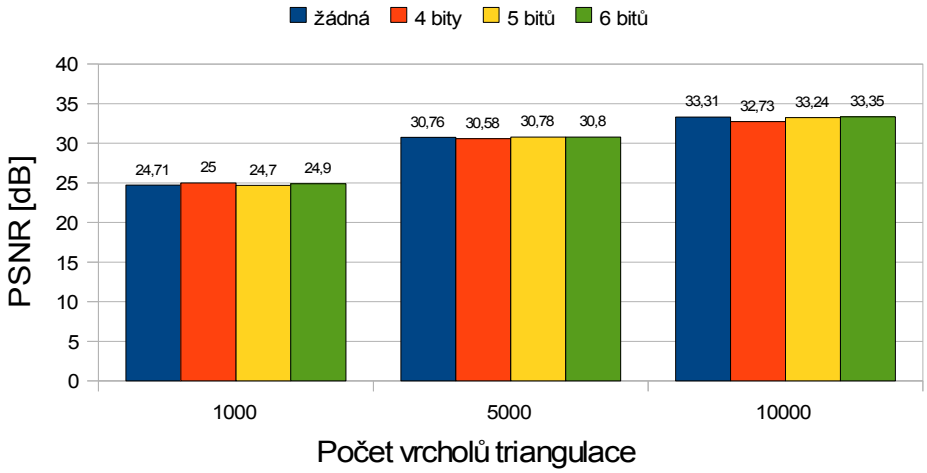
Ze srovnání obrazů (obr. 7.4) je vidět, že dynamická kvantizace podává dle očekávání lepší výsledky. Na levém obrazu (rovnoměrná kvantizace) si lze všimnout viditelného rozdělení na jednobarevné oblasti. Na dynamické kvantizaci si lze tohoto jevu také všimnout, ale již v menší míře, jak také ukazuje PSNR indikátor. V obou případech je vidět, že tři bity na pixel jsou nedostačující. Další testy budou tedy prováděny s dynamickou kvantizací na čtyřech a více bitech.

7.2.2 Výsledky

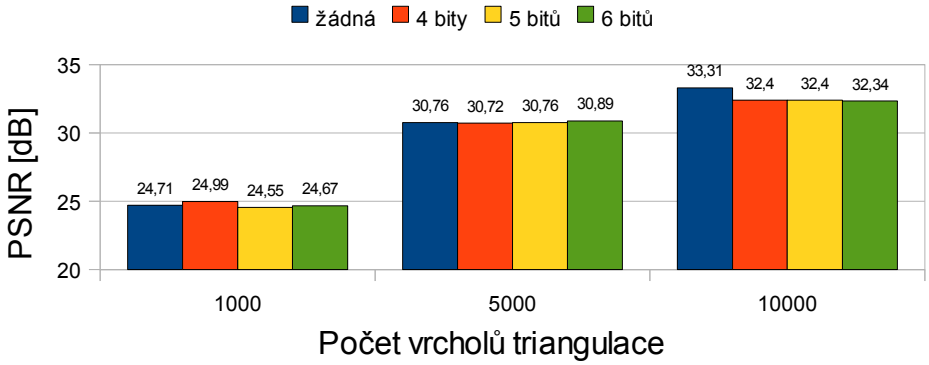
Následuje srovnání kvantizovaného a nekvantizovaného obrazu. Všechny testy jsou prováděny pro nízké počty vrcholů triangulace (1 000 – 10 000), protože vyšší počty jsou z hlediska komprese již značně nevýhodné.

Obraz 7.5: Obraz Lena s 5000 vrcholy bez kvantizace (vlevo) – PSNR 30,8 dB a s dynamickou čtyřbitovou kvantizací (vpravo) – PSNR 30,6 dB.

Z obrazu 7.5 je vidět, že při čtyřbitové kvantizaci je rozdíl oproti původnímu obrazu téměř zanedbatelný. Při větší hloubce kvantizace tak lze očekávat ještě menší rozdíly.

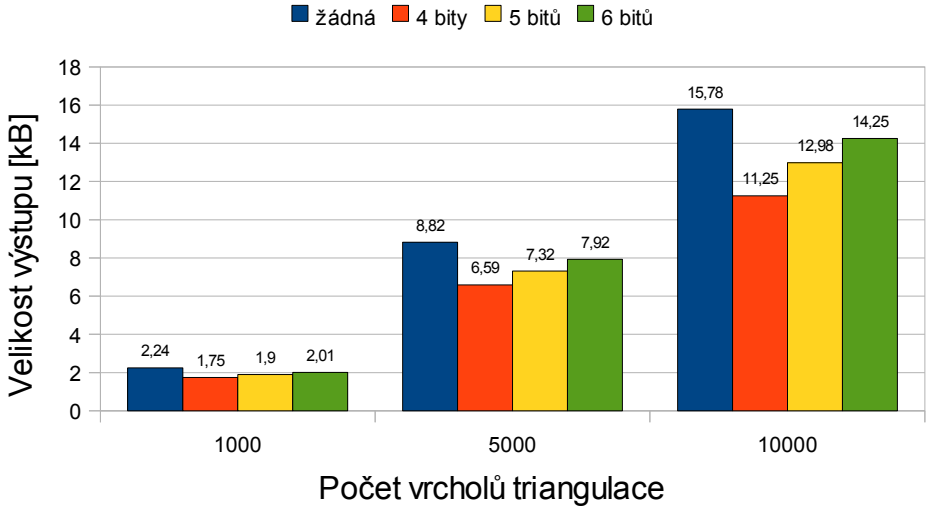


Graf 7.4: Vliv kvantizace na kvalitu obrazu Lena.



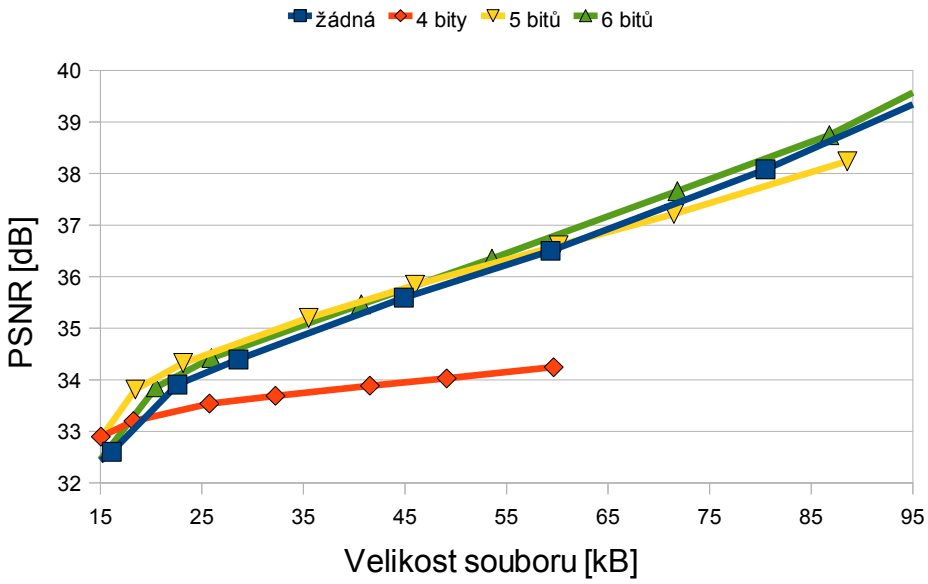
Graf 7.5: Vliv kvantizace na kvalitu obrazu Peppers.

Na grafech 7.4 a 7.5 si lze povšimnout, že vliv kvantizace na 4 a více bitů na kvalitu obrazu není zásadní. U obrazu Lena je největší propad 0,5 dB PSNR. V obraze Peppers je maximální propad 1 dB PSNR, což je poměrně hodně. Výsledek je ovšem ovlivněn nepřesností při konstrukci triangulace – viz kapitola 6.3 (jeden okraj obrazu není správně triangulován). Od čtyřbitové a větší kvantizace tedy lze očekávat poměrně malou ztrátu kvality, a teoreticky lze dosáhnout úspory až šestiny velikosti výstupu. Zbývá tedy zjistit, jaký vliv má kvantizace na velikost komprimovaného výstupu v praxi.



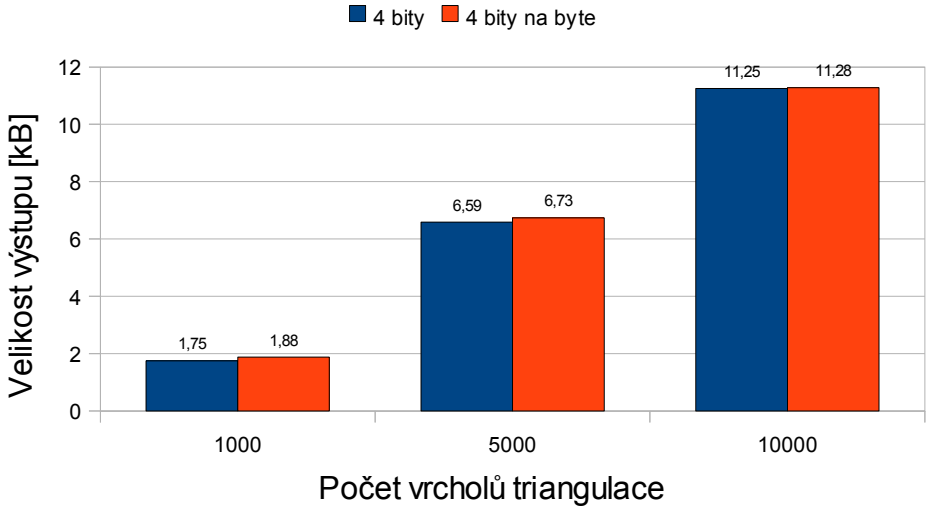
Graf 7.6: Vliv kvantizace na velikost výstupu pro obraz Lena.

Vrcholy jsou na výstup seřazeny metodou LZHEC a následně komprimovány lpaq kompresí (viz kapitoly 4.1 a 3.2.3), jejichž kombinace vycházela v testech nejlépe. Z grafu je patrné, že kvantizace je skutečně přínosná. S rostoucím počtem vrcholů je kvantizace čím dál výhodnější. Při 10 000 vrcholech triangulace je komprimovaný kvantizovaný výstup dokonce o čtvrtinu menší. Naopak při triangulacích s nízkým počtem vrcholů (pod 1 000) již kvantizace nemá velký smysl. Z grafu 7.7 je vidět výrazný pokles kvality čtyřbitové kvantizace pro obraz Peppers se vzrůstajícím počtem vrcholů (a velikostí) triangulace. Podobně se chovají i ostatní testované obrazy. Oproti tomu kvantizace na 5 bitů je při velkých triangulacích prakticky srovnatelná s triangulací bez kvantizace. Pro triangulace s 20 000 vrcholy a více je tak vhodnější použít pětibitovou kvantizaci. Pro menší triangulace je čtyřbitová kvantizace nejvýhodnější jak je vidět z předchozích grafů.



Graf 7.7: Vliv kvantizace na velikost výstupu pro kvalitní triangulace (zhruba 20 000 a více vrcholů) pro obraz Peppers.

Při kvantizaci byly kódovány na výstup průměrné hodnoty jasů množin vytvořených kvantizací. Pro rekonstrukci obrazu jsou informace na výstupu tedy dostačující. Protože jsou ale hodnoty rozprostřeny po celém osmibitovém rozsahu, může kvůli tomu být komprese méně úspěšná. Proto byl ještě proveden jeden pokus, ve kterém jsou místo kvantizovaných hodnot jasů ukládány na výstup identifikátory množin – pro čtyřbitovou kvantizaci tedy rozsah hodnot od 0 do 15. Pro rekonstrukci obrazu je pak ale nutné přidat na výstup mapovaní identifikátorů množin na skutečnou hodnotu jasu. K tomu je potřeba v tomto případě pouze seřazené pole kvantizovaných jasů – tedy 16 bytů. Je jasné, že pro kvantizace s vyšším počtem bytů toto pole roste (maximálně do 128 bytů při sedmibitové kvantizaci).



Graf 7.8: Rozdíl mezi uložením čtyřbitových kvantizovaných hodnot na 4 bity a uložením čtyřbitových kvantizovaných hodnot na jeden byte (obraz Lena).

Poslední graf 7.8 ukazuje, že ukládání identifikátorů množin místo kvantizovaných jasů může zmenšit výstup přibližně o další 4%. Toto zlepšení je už jen velice marginální a pravděpodobně se kvůli němu nevyplatí komplikovat algoritmus.

Dále byla ještě aplikována Huffmanova komprese, ale velikost výstupu tím pouze nepatrně vzrostla. Pro 1 000 vrcholů byl výstup komprimovaný lpaq kompresí o 0,2 kB větší, pro 10 000 vrcholů o 0,4kB větší. Aby byla Huffmanova komprese úspěšná, je nutné aby počty výskytů jednotlivých symbolů byly nerovnoměrné. Jenže právě na rovnoměrnosti počtu výskytů je založen použitý kvantizační algoritmus Median cut. Proto je následná Huffmanova komprese v tomto případě bezpředmětná.

7.3 Odlišná konstrukce triangulace

Protože zmíněné úpravy konstrukce triangulace stále nejsou dostačující k tomu, aby byla triangulace schopná konkurovat JPEG 2000 kodeku, bylo provedeno několik pokusů s odlišnými trojúhelníkovými reprezentacemi. V poslední době vzniká čím dál více článků věnovaných právě triangulacím pro reprezentaci obrazu. Většina z nich používá podobný postup jako projekt vytvořený na ZČU. Aby bylo ovšem dosaženo kvality srovnatelné alespoň s JPEG kompresí, jsou při konstrukci triangulace navíc zavedeny určité nedeterministické přístupy jako například přesun vrcholů triangulace algoritmem simulovaného žíhání. Shrnutí těchto metod lze nalézt v [Leh07]. Výsledky jsou vskutku kvalitnější, ovšem za cenu složitosti komprese. Pro obraz o rozměrech 512x512 trvá konstrukce triangulace řádově desítky minut, což je v praxi naprosto neúnosné. Proto jsem zvolil jiný přístup a tím je deterministická inkrementální konstrukce podobná projektu ze ZČU. Na rozdíl od projektu ale není výsledná triangulace Delaunayova, takže by nemuselo docházet k poškození hran vlivem nutnosti zachovat Delaunayovu podmínku (viz kapitola 3.1.3).

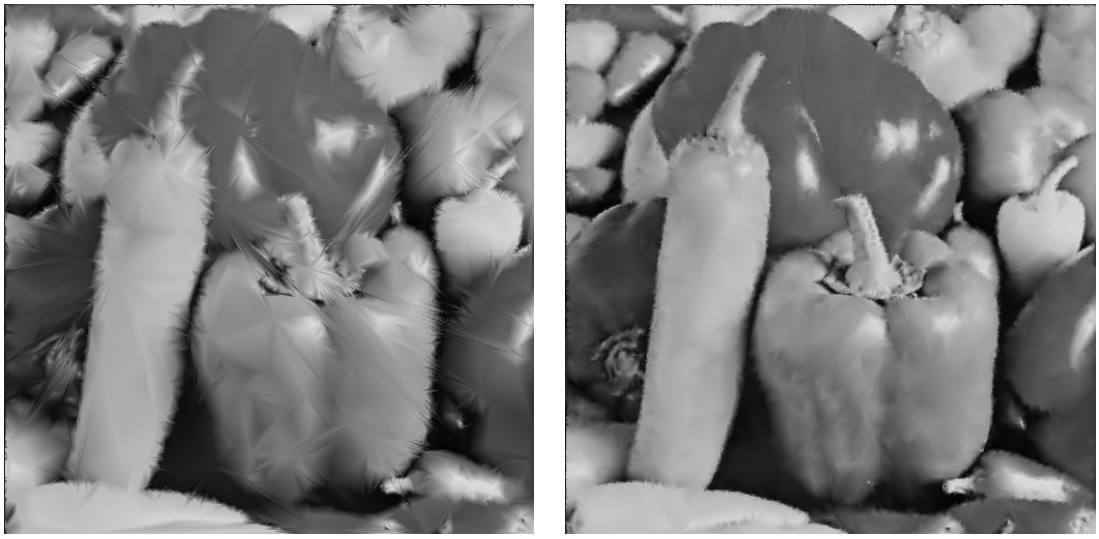
7.3.1 Konstrukce

Inkrementální konstrukce začíná tvorbou dvou trojúhelníků, který reprezentují celý obraz. Protože v triangulaci nemusí být splněno Delaunayovo kritérium opsané kružnice, není nutné vytvářet konvexní obálku jako tomu bylo v ZČU projektu (viz kapitola 3.2.1). V dalších krocích se pak přidávají vrcholy následujícím postupem:

1. Výběr trojúhelníku s největší chybou oproti rastrovému obrazu. Tento krok je podobný jako u metody Brute, která se v používá v ZČU triangulaci. U ZČU triangulace se však vybírá bod, jehož navazující trojúhelníky mají naopak nejmenší chybu, protože triangulace je decimována.
2. Dále je nutné tento trojúhelník rozdělit. Jednou z možností by bylo vybrat nějaký bod uvnitř trojúhelníku, a trojúhelník tak rozdělit na tři menší. Triangulace ovšem není Delaunayova, a tak by velmi rychle vzniklo mnoho úzkých a dlouhých trojúhelníků, protože nenásleduje prohazování hran. Aby bylo alespoň částečně zamezeno vzniku úzkých trojúhelníků, trojúhelník s největší chybou je rozdělen na jeho nejdelší hraně. Rozdělení tak zasáhne i sousední trojúhelník, takže dva původní trojúhelníky jsou nahrazeny čtyřmi (popřípadě jeden trojúhelník rozdělen na dva, pokud je to vnější hrana obrazu).
3. Krok **2** se opakuje, dokud není splněno určité kritérium. V implementaci je to pouze počet vrcholů, ale podmínka by šla snadno modifikovat na zastavení při požadovaném PSNR obrazu, protože algoritmus počítá chybu pro všechny trojúhelníky.

Rozdělení hrany z kroku 2

Na jakém místě hranu trojúhelníku rozdělit? Mohlo by se zdát, že nejkvalitnější přístup by bylo projít všechny body hrany (tj. pozice pixelů, na kterých se hrana nachází) a pro každý bod pak trojúhelníky rozdělit a vypočítat vzniklou chybu. Pak by stačilo vybrat bod, ve kterém nastala nejmenší chyba. Kromě toho, že výpočet by byl v každém kroku velice náročný (pokaždé by byl prováděn n-krát, kde n je počet bodů na hraně), nemá tento přístup ani dobré výsledky. Opět totiž rychle dochází ke vzniku dlouhých a úzkých trojúhelníků. Aby mohl být tento přístup výhodný, muselo by se vytvořit nějaké kritérium, které by trojúhelníková síť musela splňovat (podobně funguje podmínka opsané kružnice jako Delaunayovo kritérium). Lepší volbou, při tomto konkrétním algoritmu, je prozkoumat pouze několik bodů na hraně v blízkosti středu hrany (viz následující obr. 7.6).

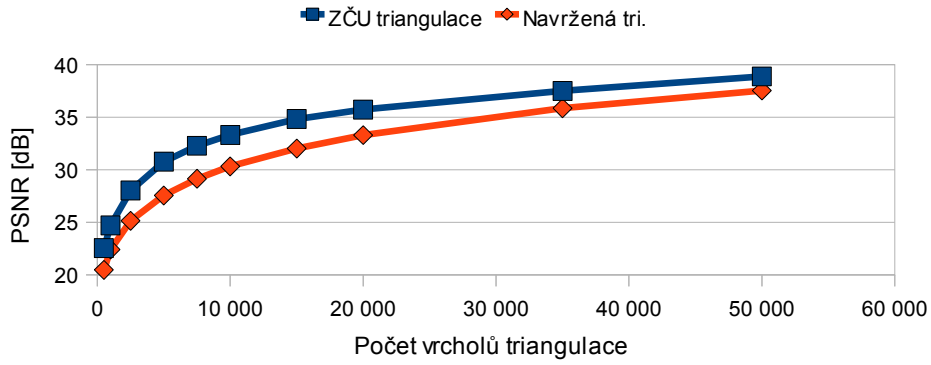


Obraz 7.6: Navržené triangulace obrazu peppers o 7 500 vrcholech s odlišným výběrem rozdělení hrany trojúhelníku. Vlevo výběr bodu s nejmenší chybou (PSNR 25,5 dB), vpravo omezení výběru na střed hrany (29 dB).

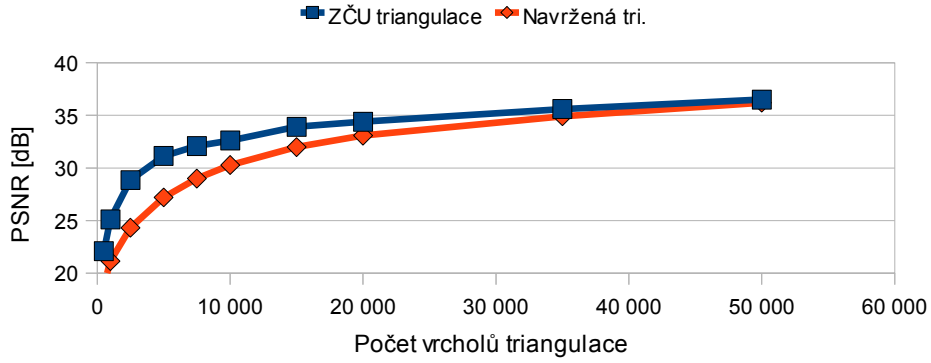
7.3.2 Výsledky

Navržený algoritmus nedává tak dobré výsledky, jako triangulace ze ZČU (viz následující grafy 7.9 a 7.10). Ukazuje se, že je problematické zachovat v obraze ostré hrany i při absenci Delaunayovy podmínky. Pokud není v triangulaci žádné kritérium zabraňující vzniku úzkých trojúhelníků, dochází k jejich vzniku příliš často. Protože v algoritmu se vzniklé vrcholy již nepřesouvají, nelze tyto úzké trojúhelníky jednoduše odstranit (přesunutím vrcholu či prohozením hrany). Algoritmy, ve kterých tyto úpravy probíhají jsou však většinou nedeterministické a velice pomalé (viz shrnutí těchto přístupů v [Leh07]). Jejich použitím lze však také dosáhnout značně lepších výsledků.

Vzhledem k porovnání navržené metody s triangulací vytvořenou na ZČU nebyla implementována komprese navržené triangulace. Protože Delaunayova triangulace z projektu potřebuje ukládat pouze souřadnice vrcholů a hodnoty jasů, bude výstup navržené metody přinejlepším stejně velký. Kvalita navržené triangulace je však o 2-3 dB PSNR nižší.



Graf 7.9: Porovnání kvality triangulací v PSNR v závislosti na počtu vrcholů triangulace pro obraz Lena.



Graf 7.10: Porovnání kvality triangulací v PSNR v závislosti na počtu vrcholů triangulace pro obraz Peppers.

7.4 Shrnutí úprav

V této kapitole je provedeno krátké shrnutí vyzkoušených úprav triangulace (konkrétně triangulací vytvářených v rámci projektu na ZČU). Byly zkoušeny tyto úpravy:

- **Předzpracování**
  - **Redukce šumu vstupního obrazu**

    Aplikací 3x3 filtru redukujícího šum bylo dosaženo průměrného zvýšení o 0,4 dB PSNR. Pro triangulace s velmi nízkým počtem vrcholů (1000 a méně) je možno použít i filtry s větších rozměrů. Filtr 5x5 či 11x11 pak v tomto případě může vylepšit kvalitu oproti triangulaci originálního obrazu přibližně o dalších 0,3 dB. Celkově je zlepšení však poměrně nevýrazné a pravděpodobně nemá smysl jím komplikovat algoritmus komprese.
  - **Úprava důležitosti jednotlivých vrcholů při výpočtu triangulace**

    Byla navržena rastrová heuristika založená na charakterizaci hran podle gradientu jasů a následné filtraci důležitých vrcholů. Ve výsledku je ovšem kvalita obrazu horší než při přímém použití heuristiky Brute, takže tato úprava nemá smysl. Je pravděpodobně, že žádná jednoduchá rastrová heuristika nevybere body, které jsou důležité pro triangulaci.

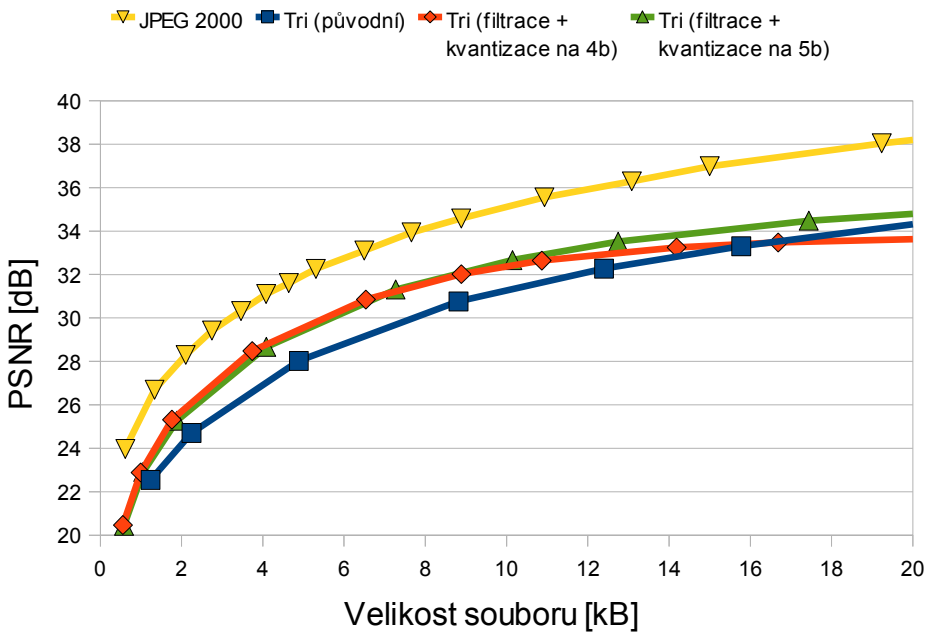
- **Kvantizace**

  Kvantizací jasů pixelů na čtyři bity metodou Median cut lze snížit velikost výstupního komprimovaného souboru přibližně o jednu pětinu velikosti. S rostoucím počtem vrcholů triangulace je kvantizace čím dál účinnější až do přibližně 20 000 vrcholů. Kvantizace na čtyři bity při více vrcholech již není dostatečně kvalitní a tak je nutné použít alespoň kvantizaci na 5 bitů.

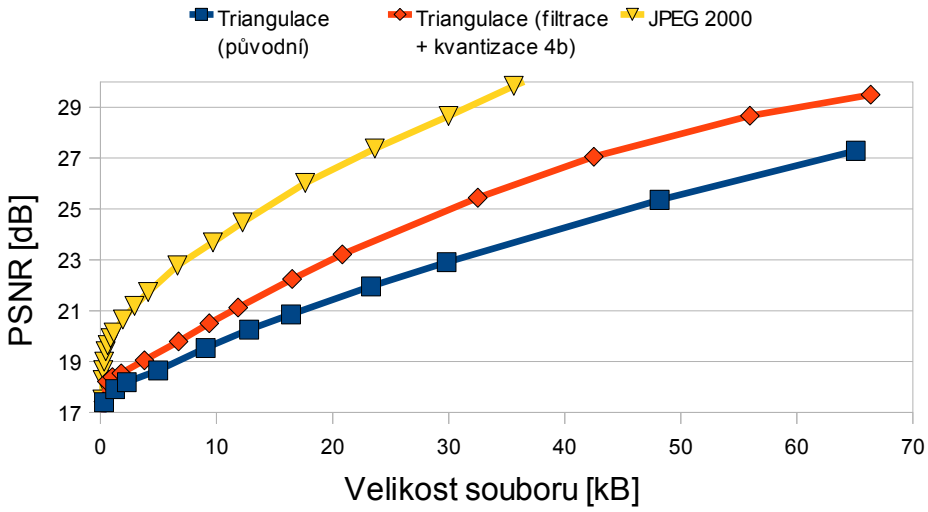
- **Odlišná konstrukce triangulace**

  Konstrukce odlišných triangulací (od Delaunayovy) je problematická, protože dochází ke vzniku velice úzkých trojúhelníků s velikostí téměř přes celý obraz. Po zabránění vzniku těchto trojúhelníků je výsledek lepší, ale nedosahuje kvality triangulace vyvinuté na ZČU. Navíc umožnění vzniku úzkých a dlouhých trojúhelníků bylo důvodem vzniku navržené metody, protože jimi lze dobře reprezentovat ostré hrany. Dalšími modifikacemi, které by zvýšily kvalitu obrazu (přesun vrcholů, prohazování hran) by algoritmus musel využívat nedeterministické postupy (viz [Leh07]).

Pokud porovnáme kvalitu triangulace před úpravami a po úspěšných úpravách (redukce šumu a následná kvantizace), je vidět jisté zlepšení (viz následující grafy 7.11 a 7.12). Při stejné velikosti souboru je dosaženo zlepšení přibližně o 1-2 dB PSNR v závislosti na obraze. Pro vyšší požadovanou kvalitu obrazu je však nutné volit kvantizace na více bitů. Z hlediska této práce jsou ale důležité zejména triangulace s velmi nízkým počtem vrcholů. Oproti JPEG 2000 kompresi však triangulace nadále zaostává (přibližně o 2 dB PSNR pro malé velikosti souborů). V obrazech, kde není příliš velké množství hran by šla kvalita triangulace ještě zvýšit, pokud by byla nalezena lepší heuristika při decimaci triangulace na požadovaný počet vrcholů. V obraze s mnoha hranami (obraz Baboon) dle mého názoru nelze triangulací dosáhnout srovnatelné kvality oproti rastrovým reprezentacím. V triangulaci musí být přítomno příliš mnoho vrcholů (téměř v každém bodě), a tak vzhledem k nutnosti jejich ukládání nemůže konkurovat kompresním poměrům rastru (jak ukazuje graf obrazu Baboon).



Graf 7.11: Kvalita komprese obrazu Lena v závislosti na velikosti komprimovaného výstupu.



Graf 7.12: Kvalita komprese obrazu Baboon v závislosti na velikosti komprimovaného výstupu.

8 Návrhy hybridních metod

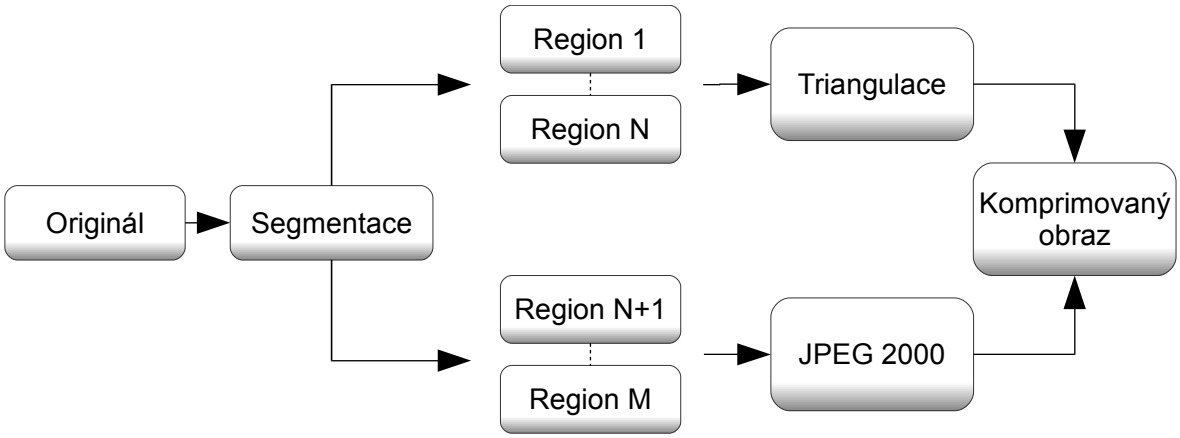
Požadavek na vytvoření formátu pro uchování digitálních obrazů kombinující jak rastrový, tak vektorový přístup vznikl zhodnocením výsledků dosažených v projektu GAAV ČR KJB101470701 na ZČU (viz [Koh09]). Projekt testoval různé možnosti uchování obrazu pomocí triangulací. Bral v potaz různé typy triangulace, určování důležitosti konkrétních bodů v obraze, kompresní metody zaměřené na triangulace apod. Shrnutí dosažených výsledků je uvedeno v kapitole 3.2. Rastrová metoda se dle zadání této práce zužuje pouze na formát JPEG 2000, který je v dnešní době jedním z nejkvalitnějších (poskytuje vysokou kvalitu obrazu při malé velikosti výstupního souboru a rozumné rychlosti komprese). Formát je podrobněji rozebrán v kapitole 4.2 spolu s formáty JPEG a PNG, které jsou nyní nejpoužívanější.

Je možné, že kombinací obou přístupů lze dosáhnout dobrých výsledků, je ale třeba si uvědomit několik nevýhod tohoto přístupu. Obraz musí být rozdělen na dvě části, nebo musí být metody aplikovány sériově (například triangulace předzpracuje obraz pro JPEG 2000 kompresi). Výsledná komprese bude tedy zajisté pomalejší kvůli druhé kompresi a případně segmentaci. Dalším problémem je ztráta jednoduché geometrické manipulace s obrazem (plynoucí z interpolace trojúhelníkové sítě – viz kapitola 3.1) při reprezentaci triangulací. Hybridní metoda by tedy měla dosáhnout lepšího kompresního poměru než samotný JPEG 2000 při stejné či lepší kvalitě, aby byla konkurenceschopná.

Následuje návrh několika možností kombinace triangulace a JPEG 2000 komprese. U každé metody je zdůvodněno z jakých předpokladů vychází, popsán postup a uvedeny výsledky.

8.1 Segmentace na regiony

Segmentace na regiony rozděluje obraz na dvourozměrné oddělené oblasti (viz obr. 8.2). U každé oblasti je pak vybrána vhodná kompresní metoda. Problémem metody je nutnost uchování hranic regionů, aby bylo možné obraz rekonstruovat. Pro segmentaci originálního obrazu jsou použity metody popsané v kapitole 5.3 – SRG a JSEG. Metoda využívá implementace segmentací vytvořené přímo jejich autory. A algoritmech tak nebyly provedeny žádné úpravy, které by potenciálně mohly segmentaci přizpůsobit dané metodě.



Obraz 8.1: Diagram metody segmentace na oblasti.

Oblasti

Prvotním problémem segmentace na regiony je volba optimálního počtu oblastí. Metody použité pro segmentaci přímou volbu počtu oblastí neposkytují, místo toho oblasti postupně slučují, dokud není dosaženo optimálního oddělení, podle vlastností regionů (závislých na dané metodě). Lze sice volit koeficient podobnosti oblastí, nelze ovšem předpovídat jejich výsledný počet pro obecný obraz.

Obraz 8.2: Segmentace obrazu Lena metodou JSEG (27 regionů).

Na výstup je potřeba uložit následující informace pro každou oblast:

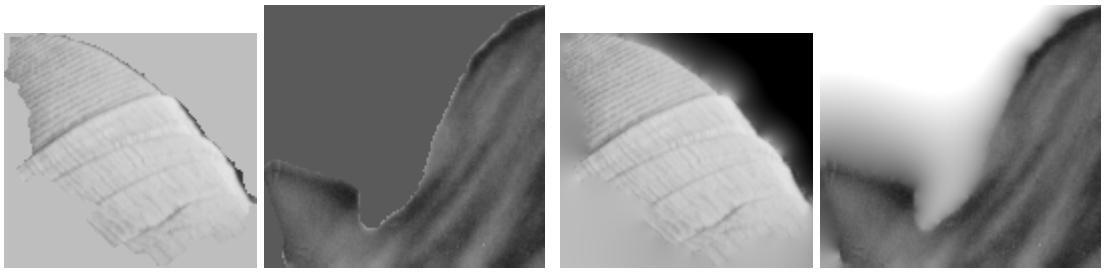
- hranice regionu,
- typ komprese regionu,
- komprimovaný region.

Aby byla metoda úspěšná, je nutné ukládat co nejméně informací o hranicích regionů. Pokud předpokládáme, že triangulace bude určité regiony komprimovat kvalitněji (při stejné kvalitě jako JPEG 2000 bude komprimovaný výstup triangulace menší), musí být triangulované regiony menší alespoň o velikost uchování hranic regionů. Z kapitoly 6 o analýze vhodných obrazových dat ovšem vyplývá, že přílišnou úsporu nelze očekávat.

Pokud z obrazu extrahujeme jednotlivé regiony (viz následující obr. 8.3), je vidět, že jejich hranice jsou poměrně složité. I při jejich kódování podle směrové růžice a následné kompresi budou pravděpodobně hranice příliš velké. Jinou možností by bylo hranice nějak zjednodušit. Např. na sérii po sobě jdoucích úseček. Podobný přístup byl zvolen v [Par03], nicméně zde se nejednalo o kombinaci dvou odlišných kodeků. Navržená metoda dosahovala jistých úspěchů při porovnání s JPEG kompresí při PSNR menším než 23 dB. Zjednodušení hranic však vede k degeneraci regionů, které by pak mohly obsahovat hrany z okolních regionů, které byly důvodem proč jsou vlastně regiony oddělené. Metoda tedy zachovává původní podobu hranic kóduje ji čtyř-směrovou růžicí. Na jeden pixel hranice jsou tak potřeba 2 bity. Poté ještě následuje ZIP komprese, která sníží celkovou velikost téměř na polovinu.

Obraz 8.3: Tři extrahované regiony obrazu Lena segmentované metodou JSEG.

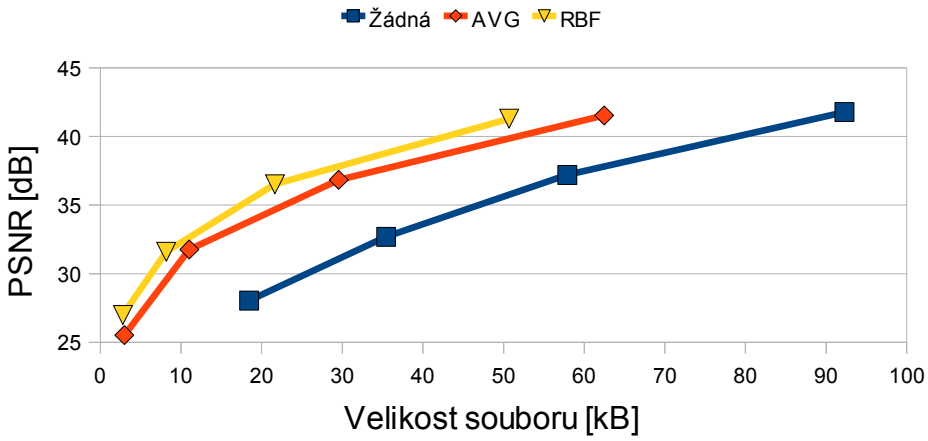
Dalším problémem je samotná komprese regionu. Protože vstupem JPEG 2000 komprese i triangulace vyvíjené na ZČU jsou obdélníkové obrazy, musí se kódovat regiony uložené v obdélníku (tzn. tak jak jsou zachyceny na obr. 8.3). To má za následek vznik výrazných hran okolo celého regionu kvůli přechodu z okolní černé. V takovém případě lze těžko očekávat dobrou kompresi. Proto je okolí regionu interpolováno. Testy byly prováděny s interpolací průměrného jasu regionu a interpolací pomocí radiálních bázových funkcí (podrobné informace o interpolaci obrazových dat pomocí radiálních bázových funkcích lze nalézt v [Zap09]). Ukázka obou přístupů je na obraze 8.4.



Obraz 8.4: Interpolace okolí regionu – obrazy vlevo interpolace průměrem, obrazy vpravo interpolace radiálními bázovými funkcemi.

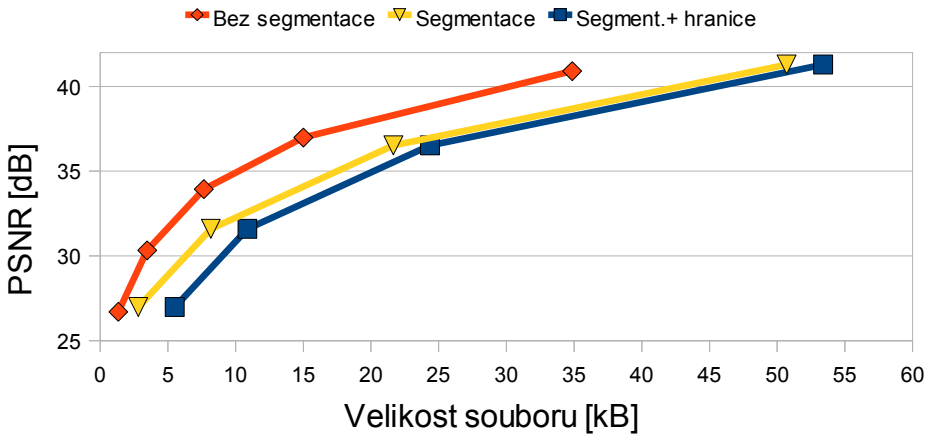
Výsledky

Interpolace pomocí radiálních bázových funkcí je z uvedených nejvýhodnější jak pro triangulaci, tak pro JPEG 2000 (graf 8.1, graf pro triangulaci je obdobný). Je vidět, že bez použití interpolace se spotřebuje velké množství dat jen na popis ostrých hran vzniklých extrakcí regionu do obdélníku. Interpolace okolí průměrným jasem regionu by šla pravděpodobně ještě vylepšit, pokud by byly brány v úvahu pouze pixely na hranici. V množství regionů je ovšem hranice dosti jasově různorodá – viz např. klobouk v obraze 8.4. Z jedné strany je hranice tmavá a z druhé světlá. Další úpravou metody, která by tento problém řešila už bychom se blížili RBF interpolaci.



Graf 8.1: Test interpolací regionů komprimovaných JPEG 2000 kodekem (obraz Lena).

Pokud srovnáme velikosti souborů vytvořených regionální kompresí při použití pouze JPEG 2000 kodeku (žádná oblast se tedy nekóduje triangulací) a přímé komprese JPEG 2000 kodeku na celý obraz (graf 8.2), je vidět, že regionální metoda má téměř 2x horší kompresní poměr. Tento výsledek je navíc dosažen bez uložení hranic regionů, které jsou k rekonstrukci obrazu nutné. Při uložení i těchto hranic se kompresní poměr ještě zhorší a to zejména pro méně kvalitní výstup, kde pak uložení hranice představuje až polovinu výstupní velikosti. Srovnání bez přítomnosti uložení hranic bylo provedeno jako odhad nejlepšího případu nějaké lepší reprezentace hranic (např. po sobě jdoucích úseček nebo parametrických křivek). Z výsledků plyne, že segmentací obrazu na regiony pravděpodobně nelze dosáhnout lepšího kompresního poměru, než přímou kompresí celého obrazu.



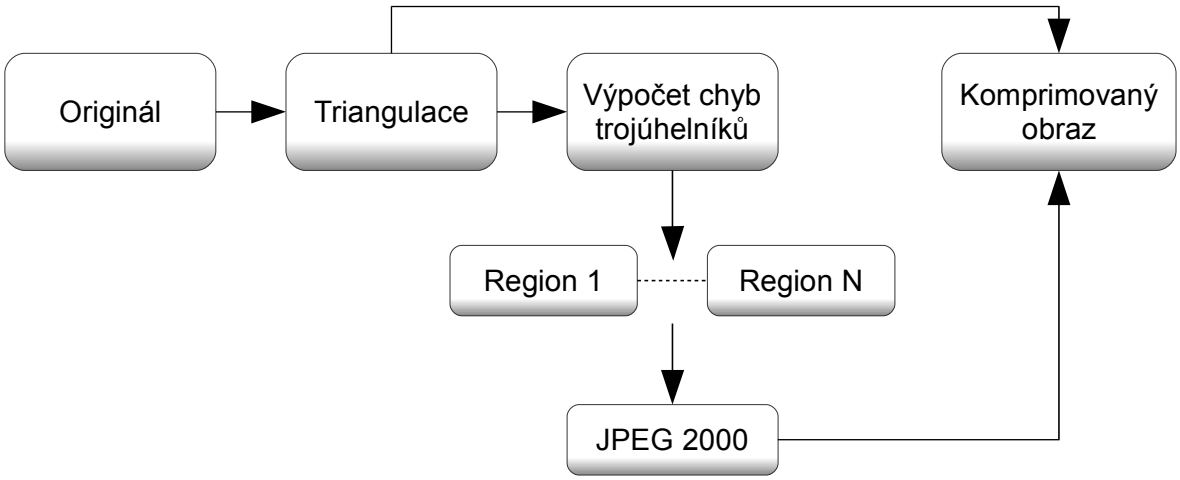
Graf 8.2: Porovnání komprese obrazu Lena bez segmentace a se segmentací. Všechny regiony získané segmentací jsou komprimovány JPEG 2000.

Pokud připustíme, že triangulace dokáže určité regiony komprimovat lépe než JPEG 2000 (viz kapitola 6), jsou to typově takové části obrazu, které neobsahují příliš mnoho informací (plynulé jasové přechody, vektorová data). Takovéto oblasti zabírají v běžných obrazech malou nebo dokonce žádnou plochu. Úspora na těchto místech je vzhledem k celkovému obrazu tedy mizivá, a tak nedokáže eliminovat přírůstek velikosti způsobený rozkladem obrazu na regiony.

Aby mohla být tato hybridní metoda úspěšná, bylo by nutné co nejvíce snížit počet oblastí (na pět či ještě méně). Dále by alespoň jedna oblast musela být dobře komprimovatelná triangulací (např. jasový přechod). Typově by se mohlo jednat o nějakou scenérii, kde by polovinu obrazu zabírala čistá obloha. V takovém případě by pravděpodobně bylo možné nepatrně snížit velikost výstupu oproti samotné JPEG 2000 kompresi. Nicméně by to bylo za cenu zásadní komplikace celé komprese.

8.2 Segmentace na trojúhelníky

Metoda má stejný princip, jako předchozí segmentace na regiony. Regiony jsou v tomto případě ale trojúhelníky triangulace. Triangulace musí být tak provedena pro celý obraz. Volba, zda použít na konkrétní trojúhelník JPEG kompresi (místo interpolace z triangulace) je provedena na základě výpočtu chyby každého interpolovaného trojúhelníku oproti originálu. Oproti předchozí metodě jsou regiony již implicitně dány triangulací, takže není nutné uchovávat další hranice. U každého trojúhelníkového regionu, který je komprimován JPEG 2000 kodekem stačí pouze uvést odkaz na trojúhelník.



Obraz 8.5: Diagram metody segmentace obrazu na trojúhelníky.

Trojúhelníkové oblasti

S rostoucím počtem trojúhelníků bude triangulace přesnější a budou vznikat menší trojúhelníky. Pro každou trojúhelníkovou oblast, která bude komprimována JPEG 2000, je nutné kódovat obdélníkový region – vzniknou tedy uměle vytvořené hrany (viz předchozí metoda o rozdělení na regiony a interpolace okolí regionů). Čím jsou trojúhelníky menší, tím narůstá počet uměle vytvořených hran pro trojúhelníky reprezentované JPEG 2000. Tento počet by měl ovšem s kvalitnější triangulací klesat, protože bude potřeba reprezentovat JPEG 2000 kompresí méně trojúhelníků. Nejvhodnější kvalita triangulace bude pravděpodobně také závislá na komprimovaném obraze.

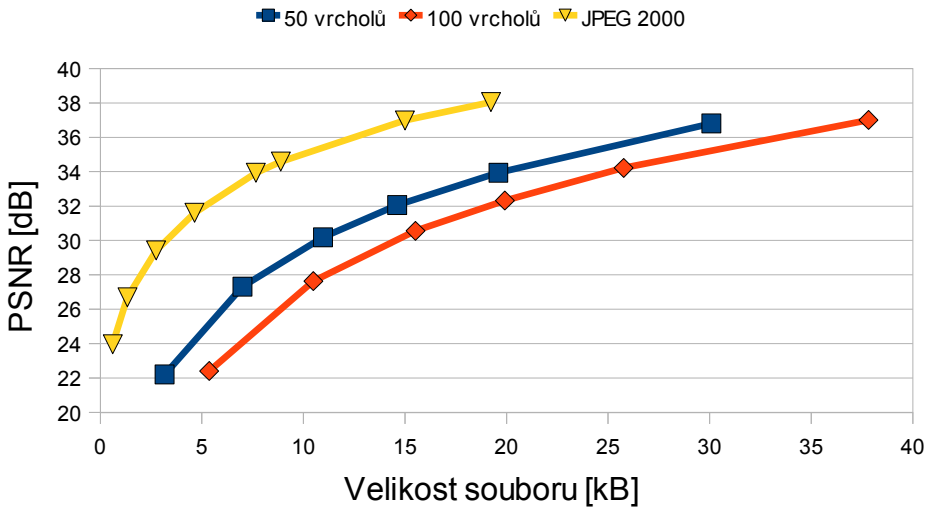
Volba, kterou metodou bude trojúhelník komprimován, je určena podle požadovaného PSNR obrazu podobně jako u předchozí metody. Pokud má daný trojúhelník nižší než požadované PSNR, bude komprimován JPEG 2000 kodekem. Nelze ovšem předem určit kolik má mít vrcholů triangulace, řádově to ale budou muset být pouze desítky vrcholů.

Metoda je podobná předchozímu přístupu, a tak se dají očekávat i podobné výsledky. Protože je pravděpodobné, že většina oblastí bude komprimována JPEG 2000 kompresí, je

nutné, aby měla triangulace co nejméně vrcholů. Čím bude mít triangulace více trojúhelníků, tím více jich bude komprimováno oddělenými JPEG soubory, čímž bude zvýšena velikost výstupu. Opět se dá očekávat, že triangulací bude reprezentováno velmi málo oblastí.

Výsledky

Z grafu 8.3 uvedeného níže je opět patrné, že přímá JPEG 2000 komprese je mnohem výhodnější. I při malém počtu vrcholů triangulace je metoda neúspěšná. Následující rozbor se vztahuje na šedotónový obraz Lena. Výsledky ostatních testovaných obrazů jsou zpravidla ještě horší, protože mají méně oblastí s nízkým kontrastem. Při 50ti vrcholech triangulace nemá ani jedna trojúhelníková oblast alespoň 30 dB PSNR. Nad tuto hranici tedy metoda v podstatě rozdělí obraz na více než 50 trojúhelníkových oblastí, a každou z nich komprimuje odděleně JPEG 2000 kodérem, což je velice neefektivní. Při zvýšení počtu vrcholů triangulace na 100 je kvalita nižší. Triangulací je kódováno naprosté minimum trojúhelníků – pouze tři trojúhelníky mají alespoň 30 dB PSNR. Pro obecný digitální obraz je tedy metoda nepoužitelná.

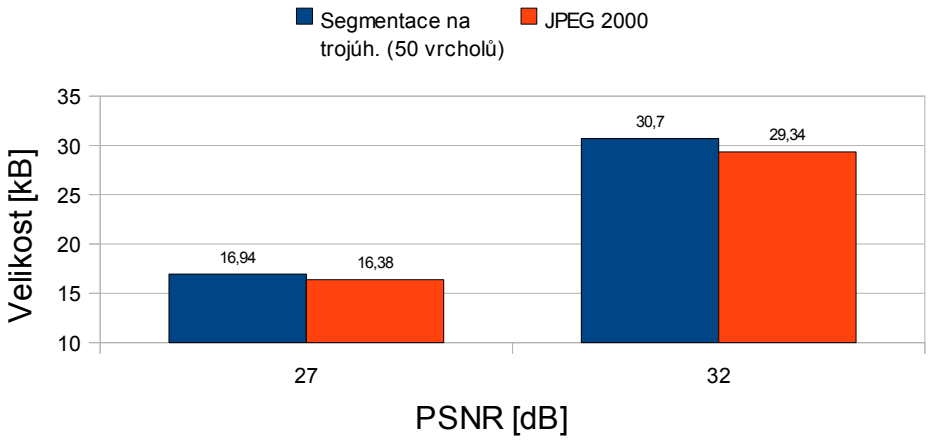


Graf 8.3: Segmentace na trojúhelníky pro obraz Lena. Porovnání segmentace triangulací o 50ti vrcholech, 100 vrcholech a přímou kompresí obrazu JPEG 2000.

Pokud se omezíme na obrazy, které by mohly obsahovat segmenty pro triangulaci výhodné, výsledek pravděpodobně nebude o mnoho lepší, protože budou pořád jednotlivé JPEG 2000 segmenty komprimovány odděleně.

Další možností by tedy bylo metodu upravit tak, aby docházelo ke spojování okolních segmentů v případě, že budou komprimovány JPEG 2000 kodérem (podobná úprava by šla provést i pro předchozí metodu). Ukazuje se, že ani po této změně není hybridní metoda výhodnější, než samotný JPEG 2000, a to i při typově vhodném obraze (viz obraz 8.6 a následující graf 8.4). Rozdíl ve výsledné velikosti je pouze nepatrný, ale je vidět, že ani v tomto případě nemá hybridní metoda navrch. Přesto je zde určitý potenciál – při typově podobných obrazech ve velmi vysokých rozlišeních by se již metoda mohla vyplatit. Na jasový přechod bude potřebovat triangulace pořád stejný počet vrcholů, zatímco velikost přechodu v JPEG 2000 bude růst v závislosti na rozlišení. Omezením jak typu obrazu, tak i jeho velikosti se však použitelnost metody příliš zužuje, a tak se domnívám, že další pokusy nemají v praxi smysl.

Obraz 8.6: Typově vhodný obraz pro navrhovanou hybridní metodu. Vpravo část obrazu komprimovaná JPEG 2000.

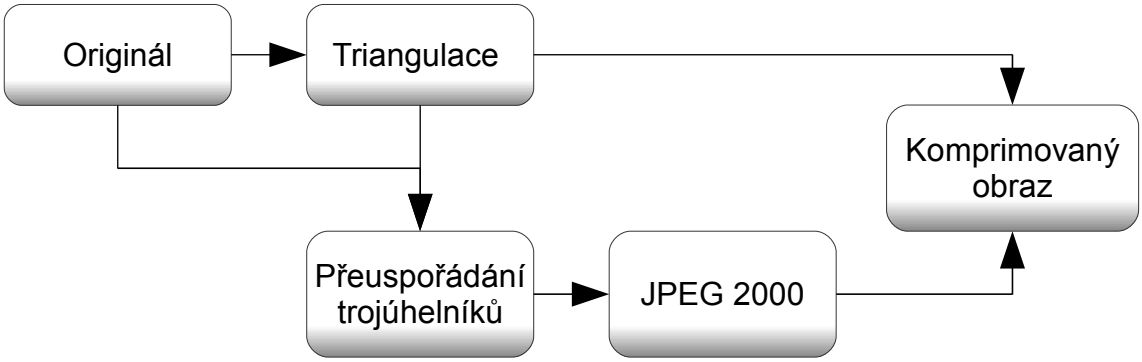


Graf 8.4: Segmentace na trojúhelníky pro obraz krajiny výše. Porovnání segmentace triangulací o 50ti vrcholech a přímou kompresí obrazu JPEG 2000.

8.3 Triangulace jako předzpracování

Triangulaci je možné použít také jako předzpracování pro následnou JPEG 2000 kompresi. Myšlenkou tohoto přístupu je přerovnání dat takovým způsobem, aby byla vylepšena JPEG 2000 komprese. Pokud by triangulace jednotlivými trojúhelníky dobře oddělovala jednotlivé segmenty obrazu, bude každý tento trojúhelníkový segment buď jasový přechod, nebo jednobarevná plocha. Když bychom dokázali trojúhelníky v obraze přerovnat podle například průměrného jasu segmentu, měl by vzniknout obraz, který by zhruba odpovídal jednomu jasovému přechodu. Samozřejmě se nelze vyhnout šumu a dalším vysokofrekvenčním datům, která triangulace nepostihne.

K uchování celého obrazu je tedy zapotřebí uložit triangulaci originálního obrazu a přeuspořádaný obraz komprimovaný JPEG 2000 kodekem. Samotné uspořádání vychází pouze z jasů v trojúhelníkové síti, takže není nutné ukládat.



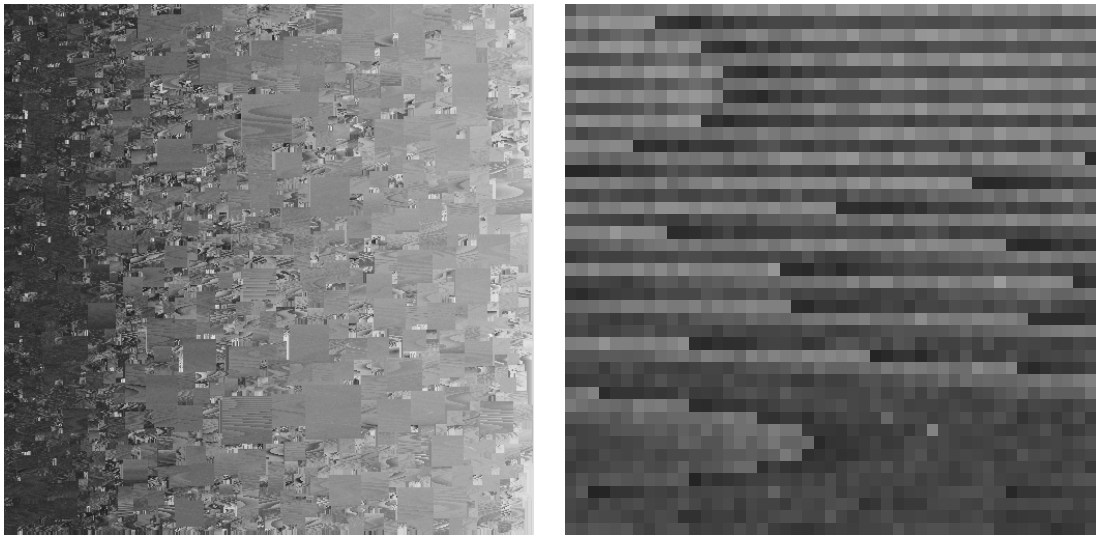
Obraz 8.7: Diagram metody založené na předzpracování obrazu triangulací.

Po přerovnání trojúhelníků lze ovšem těžko trojúhelníky poskládat tak, aby se vešly do původních rozměrů obrazu. Pro zachování všech trojúhelníků by bylo nutné přeuspořádaný obraz rozšířit. Navíc by mezi trojúhelníky vznikly mezery, které by se musely vyplnit podle okolních dat, aby nedošlo k vytvoření umělých hran. Pro zjednodušení výpočtu byly trojúhelníky převedeny na čtvercové oblasti. Díky tomu bylo možné zvolit jednoduchý algoritmus pro poskládání těchto oblastí do jednoho obrazu.

Vizualizace a analýza

Z obrazu 8.8 je vidět, že v přeuspořádaném obraze je poměrně hodně šumu, což pravděpodobně bude mít velký vliv na následnou JPEG 2000 kompresi. Samotná komprimovaná triangulace obrazu Lena o 10 000 vrcholech však již zabírá téměř 16 kB. Originální obraz komprimovaný JPEG 2000 při velikosti 15 kB má 37 dB PSNR. Z toho vyplývá, že triangulace musí mít mnohem méně vrcholů. Pro menší triangulace však přeuspořádáním trojúhelníků vzniká velké množství hran, které v původním obraze nebyly (viz obr. 8.9). Je tomu tak, protože triangulace při tak málo vrcholech již nedokáže postihnout všechny hrany, a tak některé procházejí daným trojúhelníkovým segmentem. Po převedení trojúhelníkového segmentu na čtvercový je pak hrana roztroušena po celém segmentu.

Obraz 8.8: Triangulace jako předzpracování. Přeuspořádání trojúhelníků triangulace do čtvercových segmentů seřazených dle jasu původního trojúhelníku. Vlevo triangulace obrazu Lena na 10 000 vrcholech, vpravo přeuspořádaný obraz.



Obraz 8.9: Přeuspořádání trojúhelníků triangulace do čtvercových segmentů seřazených dle jasu původního trojúhelníku pro triangulaci o 1 000 vrcholech. Vpravo zvětšený a zvýrazněný segment obrazu, který obsahuje hranu.

Výsledky

Komprimovaná triangulace obrazu Lena o 1 000 vrcholech zabírá 2,2 kB. Pod 29 dB PSNR je samotný JPEG 2000 menší. Pokud bychom chtěli dosáhnout například 35 dB PSNR, má samotný JPEG 2000 výstup 11 kB. Pro použití v hybridní metodě nám tedy zbývá maximálně 9kB. Přeuspořádaný obraz komprimovaný JPEG 2000, který zabírá 9 kB, má však pouze 26 dB PSNR. Obraz komprimovaný hybridní metodou je tedy o 9 dB PSNR méně kvalitní. Z toho plyne, že takovéto přeuspořádání JPEG 2000 kodeku zásadně nesvědčí.

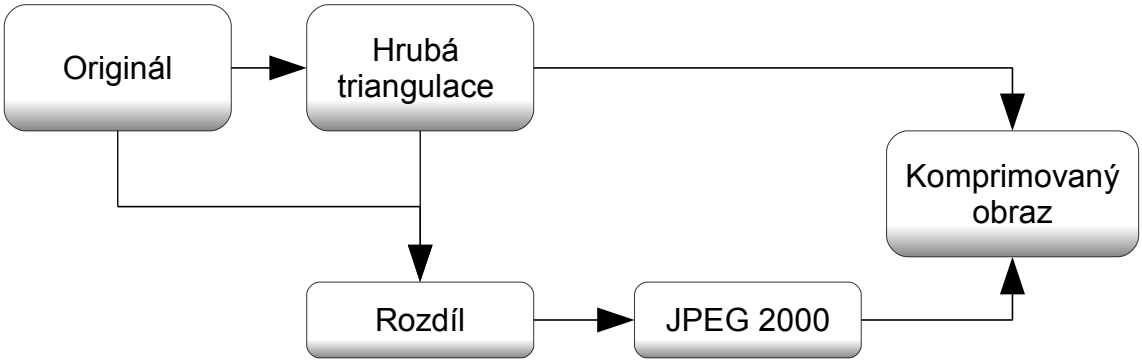
Metoda na přeuspořádaný výstup ukládala jasy z původního obrazu. Pokud trojúhelník obsahoval jasový přechod, bude tento přechod převodem na čtvercovou oblast roztroušen podobně jako hrana v obr. 8.9. Tomuto problému lze zamezit uložením rozdílu původního jasu v obraze a interpolovaného jasu v trojúhelníku. Po této úpravě dojde k vylepšení uvedeného příkladu o 4 dB. Použitím samotného JPEG 2000 je ale obraz stále o 5 dB PSNR kvalitnější. Domnívám se, že rozdíl je tak zásadní, že nemá smysl testovat další obrazy. Většinou obsahují více hran než obraz Lena (např. obraz Fishingboat a Baboon), a tak bude výsledná kvalita nižší.

8.4 Rozdílová metoda

Metoda je založena na jednoduchém faktu, že čím má triangulace méně bodů, tím roste její efektivita oproti rastrové kompresi. Protože je nutné uchovávat v triangulaci pozice všech vrcholů, se vzrůstajícím počtem vrcholů přestává být triangulace konkurenceschopná rastrovým metodám. Tam kde rastr potřebuje pouze hodnotu jasu vrcholu, triangulace potřebuje ještě pozici v mřížce – tedy x a y souřadnici. Na vrchol musí tak uchovávat 3krát více informací.

Analýza obou přístupů (viz kapitola 6) ukazuje, že triangulace dokáže pomocí méně informací uchovat ostré hrany obrazu a gradientní přechody. JPEG 2000 je naopak úspěšnější v kompresi nízkých a vysokých frekvencí v obraze. Pokud bude triangulace dobře postihovat ostré hrany a přechody, může JPEG 2000 zpracovávat zbytek obrazu.

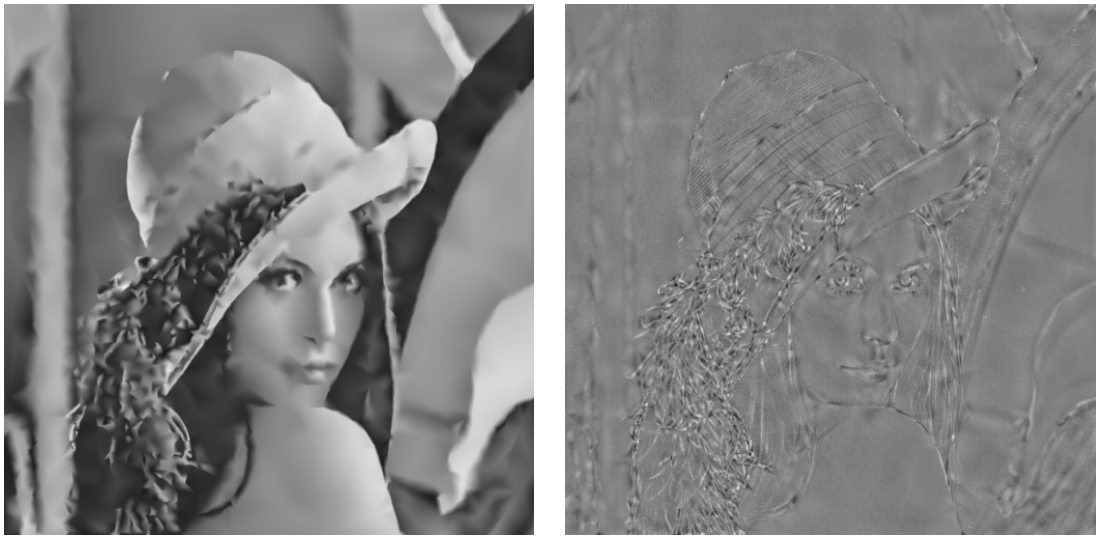
Oddělení obou částí je v této metodě provedeno pouze odečtením. Nejdřív je provedena hrubá triangulace pod 5000 vrcholů. Větší triangulace nemají smysl kvůli nedostatečné kompresi oproti JPEG 2000. Dále je rekonstruován výsledek triangulace (tedy komprimovaný obraz). Výsledný obraz je odečten od originálu. Tím vznikne rozdílový obraz obsahující všechny informace, které triangulace nezpracovala. V tomto obraze by se tedy neměly vyskytovat zejména několika-pixelové hrany a přechody. Většina rozdílového obrazu by měl být vysokofrekvenční šum či data. Protože komprese JPEG 2000 má ztrátovou i bezztrátovou variantu, šla by tato metoda teoreticky použít pro uložení jakékoliv kvality.



Obraz 8.10: Diagram rozdílové metody.

Vizualizace

Následující obraz 8.11 zobrazuje ukázku rozdílového obrazu. Je vidět, že ostré hrany nebyly triangulací reprezentovány dostatečně přesně, protože se v rozdílovém obraze vyskytují. Např. v oblasti „střapce“ klobouku je to pochopitelné, protože triangulace má pouze 2 000 vrcholů, takže tuto oblast nelze triangulací popsat o mnoho lépe. Problémem je spíše rozhraní klobouku s pozadím, případně hrany na pozadí. Očekávalo se, že právě tyto části obrazu bude kódovat pouze triangulace. Jasové přechody (např. rameno Leny na obraze 8.11) jsou triangulací charakterizovány dobře, a tak se v rozdílovém obraze nachází téměř jednobarevná plocha. Přechody však zabírají ve výstupu JPEG 2000 jen malou část souboru.

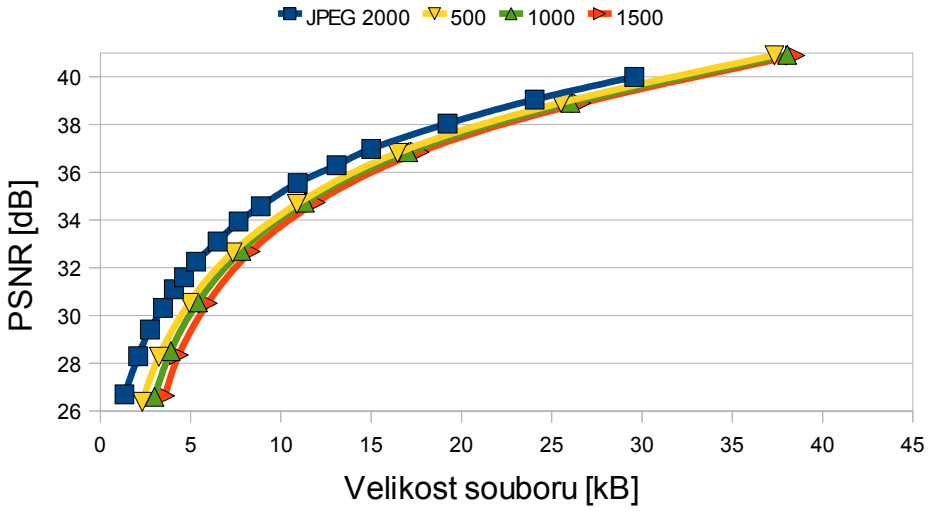


Obraz 8.11: Vlevo triangulace obrazu Lena o 2000 vrcholech. Vpravo rozdíl od originálního obrazu, který bude komprimován JPEG 2000.

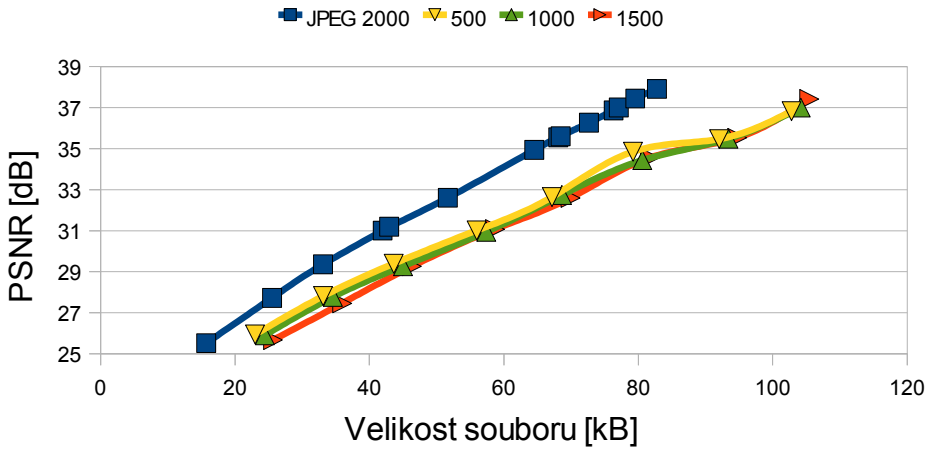
Výsledky

Metoda byla testována pro standardní sadu již popsaných obrazů. Triangulace byla prováděna v rozsahu 100 až 5 000 vrcholů. Větší triangulace již nejsou z hlediska velikosti výstupu konkurenceschopné. U triangulací byla použita navržená kvantizace a filtrace šumu, aby bylo dosaženo co nejlepších výsledků.

JPEG 2000 komprese rozdílového obrazu probíhala v rozsahu 27 – 41 dB PSNR. Protože odstíny šedi rozdílového obrazu nemají takový rozsah jako v původním obraze, je výsledkem komprese rozdílu při nižším PSNR pouze jednobarevná plocha.



Graf 8.5: Rozdílová metoda pro obraz Lena. Porovnání samostatné JPEG 2000 komprese a rozdílové metody při triangulaci o 500, 1000, 1500 vrcholech.

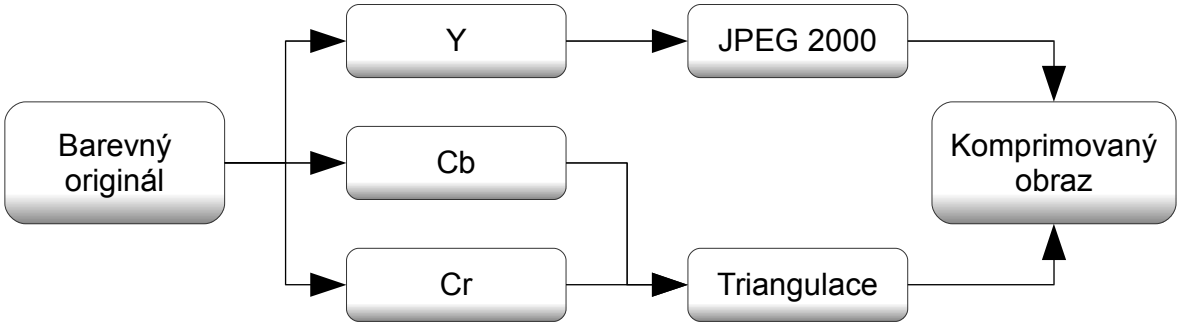


Graf 8.6: Rozdílová metoda pro obraz Baboon. Porovnání samostatné JPEG 2000 komprese a rozdílové metody při triangulaci o 500, 1000, 1500 vrcholech.

Z grafů 8.5 a 8.6 je vidět, že navrhovaná metoda není v žádném případě výhodnější než samotná JPEG 2000 komprese. Pokud pomineme výsledky samotného JPEG 2000 kodeku, nejlépe vychází použití triangulace o pouze 500 vrcholech. Při menších triangulacích jsou výsledky již horší, protože příliš hrubá triangulace zanáší do rozdílového obrazu značnou chybu. Při více vrcholech je triangulace zase příliš velká na úkor JPEG 2000 části.

8.5 Segmentace podle kanálů barveného obrazu

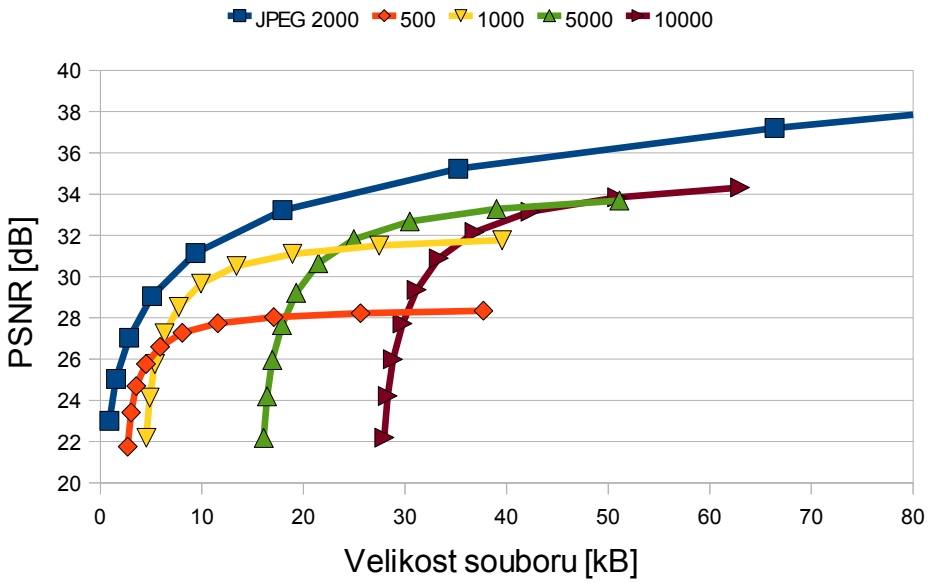
Kanály barevného obrazu v barevném modelu, který odděluje jas a odstíny nemají stejnou důležitost. Např. v modelu YCbCr (viz kapitola 2.2.2) je nejdůležitější jasový kanál Y. Lidské oko je na odstín méně citlivé než na jas, a tak se často v tomto barevném modelu používá podvzorkování obou kanálů s odstínem. Tzn. například na oblasti 2x2 jsou čtyři vzorky jasu, ale jen jeden vzorek od každého kanálu s odstínem. Podobného přístupu využívá tato hybridní metoda, akorát podvzorkování je zde přítomno v podobě hrubé triangulace pro kanály odstínu (viz diagram 8.12).



Obraz 8.12: Diagram segmentace obrazu dle kanálů.

Výsledky

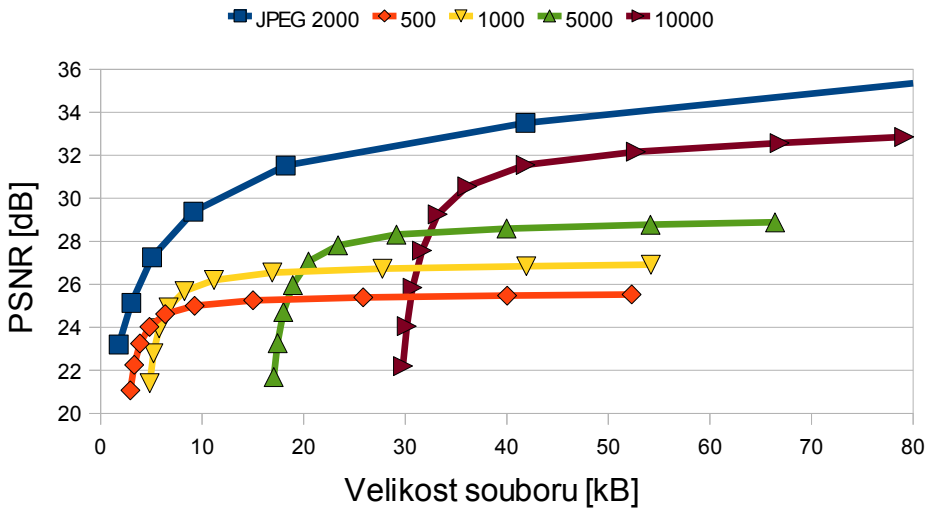
Pro porovnání kvality výstupu je obraz vždy převeden do RGB modelu. Výsledky indikátoru PSNR v následujících grafech 8.7 a 8.8 ukazují, že přímá JPEG 2000 komprese je lepší, než navržená hybridní metoda. Graf SSIM indikátoru je podobný. Na grafech je vidět, že ani při velmi hrubé triangulace obou kanálů odstínu na 500 vrcholů a při nízké kvalitě je výsledný soubor příliš velký. Obdobný výsledek dávají i ostatní testované obrazy.



Graf 8.7: Segmentace podle barevných kanálu pro obraz Lena. Porovnání přímé komprese JPEG 2000 a komprese jasového kanálu JPEG 2000 a odstínů triangulacemi o různém počtu vrcholů.

Při kvalitě 30 dB komprimovaného jasového kanálu (pomocí JPEG 2000) zabírají triangulace obou kanálů o 500 vrcholech polovinu velikosti souboru, což by byl dobrý výsledek pokud by výsledné PSNR bylo také 30 dB. Celkově ale PSNR pro barevný obraz klesne vlivem hrubých triangulací na 28 dB, a samotná JPEG 2000 komprese barevného obrazu zabírá pak méně, než obě triangulace bez Y kanálu.

Při vizuálním porovnání obou přístupů pro stejné velikosti výstupu jsou však rozdíly minimální, i když PSNR indikátor ukazuje na rozdíl kvality až o 3 dB. Lidským okem je rozdíl téměř nezaznamenatelný. Důvodem je samozřejmě použitý barevný model, protože kanály odstínu kódované triangulací obsahují méně informací, než jasový kanál. PSNR pro barevný model navíc nezohledňuje důležitost jednotlivých kanálů. Pokud by se tak činilo, byl by propad kvality menší. Tento fakt ovšem neodůvodňuje použití hybridní metody, neboť i kdyby byla kvalita srovnatelná, je mnohem jednodušší použít samostatný JPEG 2000 a to jak z hlediska komplikovanosti algoritmu, tak rychlosti komprese.



Graf 8.8: Segmentace podle barevných kanálu pro obraz Peppers. Porovnání přímé komprese JPEG 2000 a komprese jasového kanálu JPEG 2000 a odstínů triangulacemi o různém počtu vrcholů.

9 Závěr

Cílem práce bylo navrhnout hybridní kompresní metodu kombinující triangulace s JPEG 2000 kodekem. Práce se z velké části omezuje na triangulační metodu navrženou v projektu GAAV ČR KJB101470701 v rámci ZČU. Seznámení s projektem je částí zadání práce a v projektu je také zmíněna hybridní metoda jako jedna z možných cest.

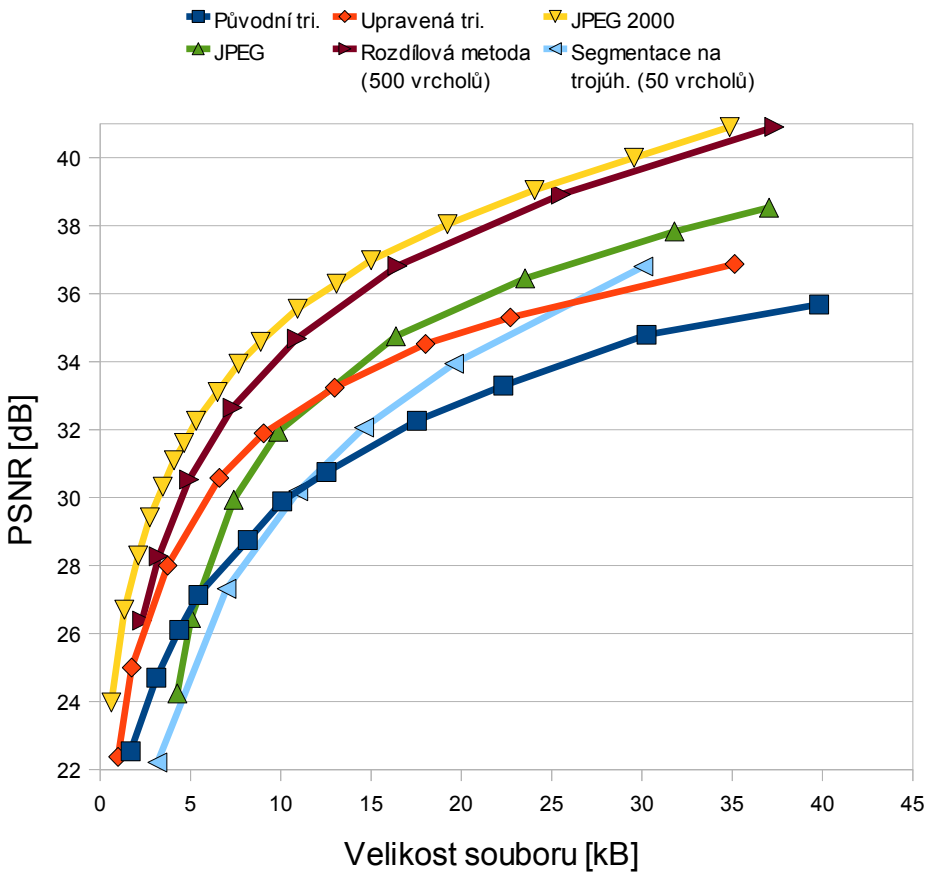
Z analýzy obou kompresí v kapitole 6 vyplývá, že triangulace jsou vhodné především pro obrazy s málo informacemi – plynulé jasové/barevné přechody, oblasti s velmi nízkým šumem a ostrými hranami. Shrnutím těchto charakteristik jsou v podstatě vektorové obrazy. V takovém případě se dá lepší kompresní poměr triangulace očekávat, vzhledem k tomu, že triangulace představuje pro digitální obraz vektorovou reprezentaci. Při kompresi obecných obrazů (zejména fotek), se ale takových oblastí vyskytuje v obraze velice málo. Je také nutno podotknout, že JPEG 2000 je v oblasti ztrátové komprese velice zdatný oponent. Je to dnes prakticky nejlepší dostupný kodek pro rastrovou reprezentaci digitálního obrazu.

Vzhledem k uvedenému srovnání obou kompresí tak bylo provedeno několik pokusů o vylepšení komprese triangulací. Jednalo se zejména o metody, které zasahovaly do již existující tvorby triangulace navržené v projektu, na který tato práce volně navazuje. Menší úspěch byl zaznamenám filtrováním šumu vstupního obrazu před samotnou triangulací. Tím bylo dosaženo zejména pro menší triangulace (pod 15 000 vrcholů) drobného vylepšení kvality obrazu – v některých případech až o 1 dB PSNR. Druhou úspěšnou úpravou triangulace bylo zavedení kvantizace jasů při výpočtu a uložení triangulace. Díky tomu byla výstupní velikost triangulace snížena až o jednu čtvrtinu (kvantizace jasu na 4 bity) při nepatrné ztrátě kvality (přibližně 0,3 dB PSNR) pro triangulace pod 10 000 vrcholů. Ostatní změny – úprava důležitosti jednotlivých vrcholů při tvorbě triangulace podle navrženého hranového filtru, použití jiných typů triangulací, byly neúspěšné. Celková kvalita triangulace byla tak úpravami zvýšena až o 1-2 dB při stejné velikosti výstupního souboru. Při srovnání s JPEG 2000 kompresí však triangulace obecného digitálního obrazu stále zaostává minimálně o další 3 dB při jakékoliv velikosti.

V hlavní části práce bylo navrženo pět hybridních metod kombinujících triangulaci vyvinutou na ZČU s JPEG 2000 kompresí (srovnání metod lze nalézt v následujícím grafu 9.1) .

- Segmentace obrazu na regiony se ukázala jako nevhodná, protože samotným rozdělením obrazu na regiony se velikost výstupu při stejné kvalitě dvakrát zvětší i pokud se každý region komprimuje jen JPEG 2000. Triangulace by tak musela být pro alespoň polovinu regionů dvakrát menší, aby se velikost výstupu vyrovnala. Pro obecný digitální obraz je však takových regionů přítomno naprosté minimum.
- Segmentace obrazu na trojúhelníkové oblasti vzniklé triangulací je sice oproti předchozí metodě úspěšnější, ale pořád nedosahuje kvality JPEG 2000 komprese. S narůstajícím počtem vrcholů triangulace se kvalita při stejné velikosti výstupu značně snižuje, protože se jednotlivé regiony komprimují odděleně. Nejlepší výsledky tato kombinace dává pro triangulace s minimálním počtem vrcholů (pod 50). Metoda je pro určité speciální obrazy téměř na úrovni JPEG 2000, ale musí se jednat o obrazy s velkými plynulými jasovými přechody (např. obraz krajiny s oblohou). Lepšího kompresního poměru by metoda pro tento případ mohla dosáhnout, ale muselo by se jednat o obrazy s velkými rozměry (řádově desetitísíce pixelů).

- Při pokusu o využití triangulace jako předzpracování pro následnou JPEG 2000 kompresi se narazilo na dva problémy. Metoda by mohla být efektivní pouze pro triangulace s velkým počtem vrcholů, nebo pro obrazy, ve kterých jsou triangulací dobře charakterizovány hrany. V prvním případě však již samotná triangulace zabírá příliš mnoho místa, takže se vyplatí rovnou použít JPEG 2000. Druhý případ nastává spíše pouze pro vektorové obrazy.
- Rozdílová metoda, která obraz dělí na dvě části pomocí triangulace je z navržených metod nejefektivnější. Je tak ovšem zejména z důvodu, že obraz komprimovaný JPEG 2000 představuje většinovou část informací originálního obrazu. I v tomto případě je ovšem přímá JPEG 2000 komprese efektivnější. Rozdíl hybridní metody oproti JPEG 2000 kompresi je v průměru 1 dB PSNR při stejné velikosti souboru.
- Hybridní metoda pro kompresi barevných obrazů využívá rozdělení jasové informace a informace o odstínech obrazu konverzí na YCbCr barevný model. Kanály odstínu jsou pak komprimovány triangulací, protože obsahují menší šum a množství informací, zatímco JPEG 2000 komprimuje jasový kanál. Z hlediska standardních indikátorů kvality je metoda zhruba o 3 dB PSNR horší, než samotný JPEG 2000. Ztráta kvality je ovšem lidským okem téměř nezaznamenatelná, protože citlivost oka ve vnímání rozdílu odstínu není tak velká, jako u rozdílu jasu.



Graf 9.1: Porovnání kompresních metod pro šedotónový obraz Lena. Původní triangulace byla vytvořena v rámci projektu, na který práce navazuje. Upravená triangulace zahrnuje vylepšení navržená v této práci. Z hybridních metod je uvedena pouze rozdílová metoda a segmentace na trojúhelníky.

Z výsledků plyne, že pro obecné digitální obrazy je nejvýhodnější použít samotnou JPEG 2000 kompresi. Použití triangulace vytvořené na ZČU, či nějaké navržené hybridní metody, má smysl pouze pro vektorová data. V takovém případě se však velikosti komprimovaných souborů pro oba přístupy pohybují v řádu jednotek kilobytů. Kombinace triangulace a JPEG 2000 se tak zdá být slepou uličkou.

Nejlepší kvality dosahují triangulace založené na stochastických úpravách, jejichž přehled lze najít v [Leh07]. Při srovnání s rastrovou reprezentací JPEG 2000 však triangulační metody pro obecný digitální obraz taktéž zaostávají. Na rozdíl od deterministických přístupů tvorby triangulace však rozdíl není tak značný. Tvorba a úprava těchto triangulací, tak aby splňovala určité kvalitativní kritéria, je navíc velice pomalá a komprese jednoho obrazu se zpravidla pohybuje v řádech minut. Jednou z možností dalšího vývoje by mohlo být urychlení konstrukce využitím hardwarové akcelerace grafických karet. Přesto se domnívám, že tyto triangulace nebudou pro obrazy, ve kterých převažují vysokofrekvenční informace, kvalitativně srovnatelné s rastrovými reprezentacemi.

V kontextu hybridních metod by mohla být triangulace využita pro urychlení fraktálové komprese digitálního obrazu, která dosahuje jinak velice kvalitních výsledků. Další možností je použití transformací obrazu (např. waveletových) přímo na trojúhelnících. Tím by potenciálně vznikla metoda, která by kvalitně reprezentovala výrazné hrany v obraze triangulací (i při zvětšení obrazu) a zachovávala přijatelnou kvalitu vysokofrekvenčních dat v oblasti uvnitř trojúhelníků.

Použité zdroje

[Abo01] Aboufadel E. *JPEG 2000: The Next Compression Standard using wavelet technology*. <http://faculty.gvsu.edu/aboufade/web/wavelets/student_work/EF> [4. prosince 2010]

[Dem06] Demaret L., Dyn N., Iske A. *Image compression by linear splines over adaptive triangulations*. Signal Processing 2006, 86(7):1604-1616

[Deng99] Deng Y., Majnunath B.S., Shin H. *Color Image Segmentation*. Department of Electrical and Computer Engineering, University of California, Santa Barbara, 1999.

[Dev01] Devillers O, Pion S, Teillaud M. *Walking in triangulation*. Proceedings of 17th Annual Symposium on Computational Geometry, ACM, Medford, Massachusetts, USA, June 3-5, str. 106-114, 2001.

[Heck82] Heckbert P.S. *Color Image Quantization for Frame Buffer Display*. Proceedings of SIGGRAPH, str. 297-307, 1982.

[Jan09] Janák T. *Interpolation method for Triangulation Represented Digtal Image*. Bakalářská práce, Západočeská univerzita v Plzni, Česká republika, 2008.

[JFIF92] Hamilton E. *JPEG File interchange format*. V 1.02, 1992.

[JPE00] *JPEG 2000 Image Coding system*. ISO/IEC FCD15444-1, V. 1.0, 2000.

[JPE93] *Digital compression and Coding of Continous-Tone Still Images – Requirements and Guidelines*. International Telecomunication Union, CCITT, T.81, 1992.

[Koh09] Kohout J. *Alternative representation of image information*. Technical Report No. DCSE/TR-2009-11, Západočeská univerzita v Plzni, Česká republika, 2009.

[Leh07] Lehner B., Umlauf G., Hamann B. *Survey of Techniques for Data-dependent Triangulations Approximating Color Images*. Department of Computer Science, University of Kaiserslautern, Germany, 2007.

[LPAQ10] *Lpaq Compression* <http://compressionratings.com/i_lpaq.html> [12. března 2010]

[Nock04] Nock R., Nielsen F. *Statistical region merging*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No. 11, 2004.

[OJP09] *OpenJPEG 2000 library*. Open-source JPEG 2000 codec written in C language. <http://www.openjpeg.org> [10. října 2009]

[Par03] Partyk M., Polec J., Kolingerová I. *Hybrid Scheme with Triangulations for Transform Coding*. Dept. of Telecommunications, Slovak University of Technology, Bratislava, Slovensko, 2003.

[PNG10] *Portable Network Graphics (PNG) Specification.* Version 1.2. <http://www.libpng.org/pub/png/spec/1.2> [23. ledna 2010]

[Ros98] Rossignac J. *Edgebreaker: Connectivity compression for triangle meshes*. GVU Technical Report GIT-GVU-98-35, Georgia Institute of Technology, 1998.

[Scha07] Schanda J. *Colorimetry: Understanding the CIE System*. Wiley Interscience, str. 61–64, ISBN 978-0-470-04904-4, 2007.

[Syk08] Sýkora R. *Komprese barevných digitálních obrazů s využitím triangulace*. Bakalářská práce, Západočeská univerzita v Plzni, Česká republika, 2008.

[Wang04] Wang Z., Bovik A. C., Sheikh H. R., Simoncelli E. P. *Image Quality Assessment: From Error Visibility to Structural Similarity*. IEEE Transactions on image processing, Vol. 13, No. 4, 2004.

[Wiki10a] *Delaunay triangulation*. Wikipedia, the Free Encyclopedia. <http://en.wikipedia.org/wiki/Delaunay_triangulation> [5. února 2010]

[Wiki10b] *Portable Network Graphics*. Wikipedia, the Free Encyclopedia. <http://en.wikipedia.org/wiki/Portable_Network_Graphics> [10. dubna 2010]

[Wiki10c] *JPEG*. Wikipedia, the Free Encyclopedia. <http://en.wikipedia.org/wiki/JPEG> [20. dubna 2010]

[Wiki10d] *Color quantization. Wikipedia*, the Free Encyclopedia. <http://en.wikipedia.org/wiki/Color_quantization> [10. dubna 2010].

[Zap09] Zaplétal J. *Aplikace radiálních bázových funkcí*. Diplomová práce, Západočeská univerzita v Plzni, Česká republika, 2009.

[Žal03] Žalik B, Kolingerová I. *An incremental construction algorithm for Delaunay triangulation using the nearest-point paradigm*. International Journal of Geographical Information Science, 17(2), str. 119-138, 2003.