

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STAVEBNÍ, OBOR GEODÉZIE, KARTOGRAFIE A GEOINFORMATIKA
KATEDRA SPECIÁLNÍ GEODÉZIE

název předmětu

GEOINFORMATIKA

název úlohy

Úloha 1: JPEG komprese a dekomprese rastru

akademický rok	semestr	vypracoval(a)	Datum	klasifikace
2023/24	Zimní	Jan Koudelka, Vojtěch Müller	20.11.2023	

Technická zpráva

1 Zadání

Implementujte algoritmus pro JPEG kompresi rastru v prostředí MATLAB, zahrnující tyto fáze:

- Transformace do Y, C_B, C_R
- Diskrétní kosinova transformace
- Kvantizace koeficientů

Kompresní algoritmus otestujte na rastroch:

- Fotografie odstínu jedné barvy
- Barevná fotografie
- Vektorová kresba
- Barevný přechod
- Černobílý rastr

Různé typy rastrů otestujte pro hodnoty faktoru komprese $q = 10, 50, 70$.

Pro každou variantu spočítejte střední kvadratickou odchylku m jednotlivých barev RGB složek. Výsledné hodnoty umístěte do přehledných tabulek a zhodnoťte, pro které typy dat je JPEG komprese nejvíce a nejméně vhodná.

Dále do algoritmu implementujte volitelné funkce:

- Resamplování rastru
- Konverze pixelů do ZIG – ZAG sekvencí
- Huffmanovo kódování
- Náhrada diskretní kosinové transformace za diskretní Fourierovu transformaci

2 Popis

JPEG komprese je nejpoužívanější metodou komprese. Důvodem oblíbenosti této metody je vysoký kompresní poměr, tedy po kompresi je zajištěno výrazné zmenšení rastru. Jedná se o ztrátovou kompresi s několika ztrátovými prvky, při jejím použití je tedy potřeba, dávat si pozor, jaké prvky jsou součástí komprese a jaký faktor komprese je použit. Faktor komprese odpovídá procentuálnímu vyjádření a nabývá tedy hodnoty od 0 do 100. Tento faktor vyjadřuje množství „zanedbaných“ informací. Také je třeba rozlišovat, pro jaký typ rastru je komprese použita. V této úloze je testována vhodnost jednotlivých rastrů pro JPEG kompresi.

Podstatou JPEG komprese je fakt, že lidské oko více vnímá změny v jasu než v barvě. Komprese tedy probíhá na barvovém systému $YC_B C_R$ a minimálně ovlivňuje jasovou složku.

V kompresi je použita složitá matematická operace, a to diskretní kosinova transformace. **[4]**

3 Postup

3.1 Separace na R, G, B, složky. [4]

Rastr, který se skládá z červené, zelené a modré složky je rozdělen na jednotlivé matice, které odpovídají jednotlivým barvám. Následující kroky probíhají pro každou barvu zvlášť.

3.2 Transformace RGB do YCbCr [4]

Za účelem nižšího ovlivnění jasu, je provedena transformace do systému YCbCr. Tato transformace je provedena podle vzorců:

$$\begin{aligned}Y &= 0,2990 \cdot R + 0,5870 \cdot G + 0,1140 \cdot B \\CB &= -0,1687 \cdot R - 0,3313 \cdot G + 0,5000 \cdot B + 128 \\CR &= 0,5000 \cdot R - 0,4187 \cdot G - 0,0813 \cdot B + 128\end{aligned}$$

3.3 Rozdělení na submatice [4]

Samotná komprese probíhá po jednotlivých sekcích. Původní rastr je tedy rozdělen na menší rastry o velikosti 8x8 pixelů, které jsou reprezentovány maticemi 8x8. Na každé sekci probíhají následující kroky zvlášť.

3.4 Převzorkování (Resampling) [4]

Prvním ztrátovým krokem komprese je převzorkování, které provádí průměrování jednotlivých submatic. V matici je tedy hodnota prvků v submatici nahrazena průměrem všech hodnot v submatici. Tyto submatice mohou nabývat různých velikostí a to 2x1, 1x2, 2x2, 4x4.

3.5 Diskrétní kosinova transformace (DCT) [4]

Matematicky nejnáročnější operace probíhá podle rovnic:

$$F(u, v) = \frac{1}{4} C(u) \cdot C(v) \left[\sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cdot \cos \frac{(2x+1)u\pi}{16} \cdot \cos \frac{(2y+1)v\pi}{16} \right]$$
$$C(u) = \begin{cases} \frac{\sqrt{2}}{2}, & u = 0 \\ 1, & u \neq 0 \end{cases}, \quad C(v) = \begin{cases} \frac{\sqrt{2}}{2}, & v = 0 \\ 1, & v \neq 0 \end{cases}$$

Z DCT vznikne matice, která má nejvyšší hodnoty v levém horním rohu a hodnoty klesají směrem doprava dolů.

3.6 Diskrétní Fourierova transformace (DFT) [4]

DFT může být náhradou DCT. Matematicky se jedná o složité vzorce, které pracují s komplexními čísly. Tyto vzorce jsou:

$$F(u, v) = \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cdot e^{-j2\pi(\frac{ux}{8} + \frac{vy}{8})}$$

$$Amplituda[F(u, v)] = \sqrt{real[F(u, v)]^2 + imag[F(u, v)]^2}$$

$$Fáze[F(u, v)] = \tan\left(\frac{imag[F(u, v)]}{real[F(u, v)]}\right)$$

Kvantizace poté probíhá na jednotlivých složkách, které vystupují z diskretní Fourierovy transformace.

3.7 Kvantizace [4]

Kvantizace pracuje s předpokladem, že hodnoty ve vstupní matici jsou největší v levém horním rohu a nízké hodnoty v pravém dolním rohu matice mohou být tedy zanedbány. Probíhá přepočet pomocí kvantizačních matic $Q(u, v)_{50}^Y$ a $Q(u, v)_{50}^C$, které jsou vytvořeny tak, aby byla jasová složka co nejméně ovlivněna. Kvantizace probíhá na vzorcích:

$$Q(u, v)_{50}^Y = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 87 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 26 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}$$

$$Q(u, v)_{50}^C = \begin{pmatrix} 17 & 18 & 24 & 47 & 66 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 69 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{pmatrix}$$

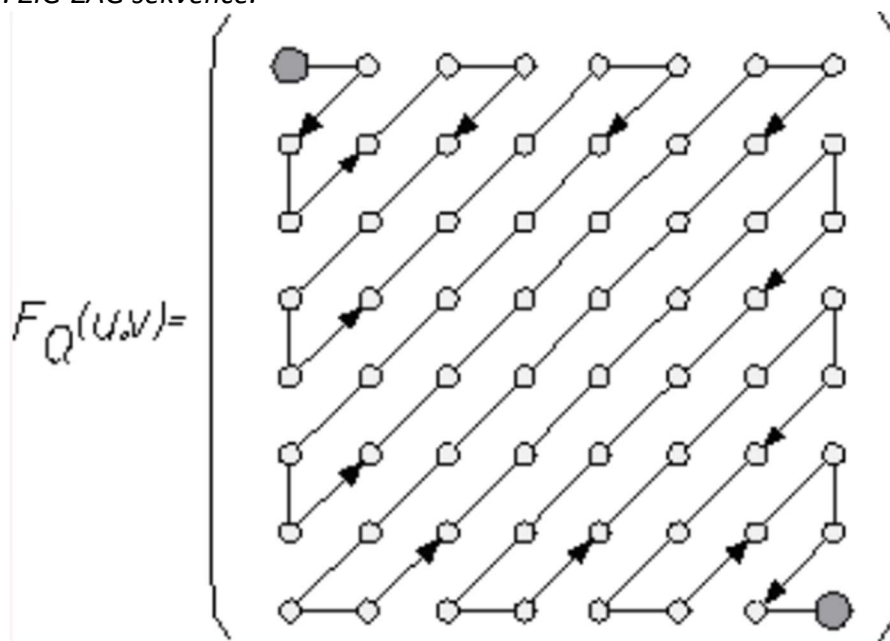
$$Q(u, v) = \frac{50 \cdot Q(u, v)_{50}}{q}$$

Faktor komprese q je uváděn v procentech a určuje množství „zanedbaných“ informací.

3.8 Uspořádání do ZIG-ZAG sekvencí [4]

Submatice je seřazena do sekvence ve směru diagonál. Tímto seřazením budou prvky se stejnými hodnotami za sebou a lze je efektivně komprimovat například Huffmanovým kódováním.

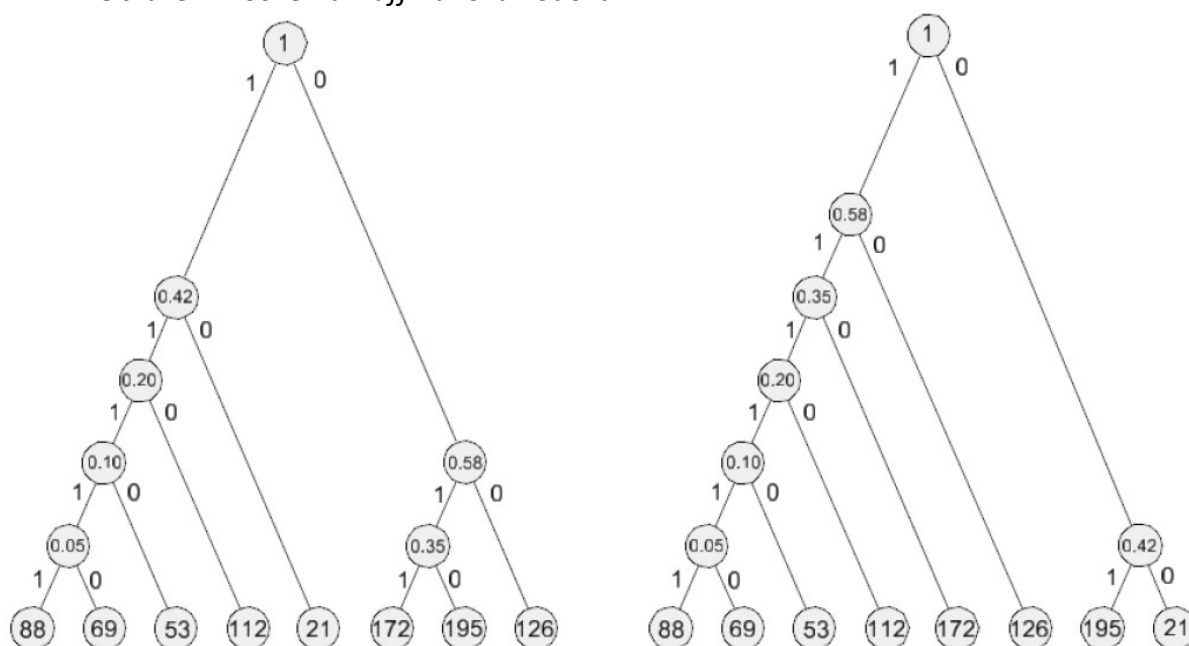
Obrázek 1: ZIG-ZAG sekvence.



3.9 Huffmanovo kódování [4]

Huffmanovo kódování je jedna z metod statistické komprese založené na analýze četnosti jednotlivých znaků vstupního souboru. Nejprve tak byla pro vstupní soubor vytvořena statistika obsahující unikátní hodnoty a jejich četnost v souboru. Z dané statistiky byly vždy nalezeny dvě hodnoty, které měly v souboru minimální četnost. Méně zastoupené hodnotě z této dvojice zapíšeme 1, té druhé 0. Z těchto hodnot vzniká kořen, jehož ohodnocení je součtem ohodnocení právě dvou listů. Stejný proces opakuje do té doby, dokud nedojdeme do konce.

Obrázek 2: Schéma Huffmanova kódování.



3.10 Dekompresi Huffmanova kódu [4]

Dekompresi probíhá od kořene dolů k listům. Načteme vždy vstupní binární znak, jeli roven 0, vstupujeme dále do pravé části vzniklého stromu. Jeli roven 1, jdeme do leva. Tímto postupem se dostaneme až k listu, jehož hodnotu zapisujeme. Postup se opakuje pro celý soubor binárních hodnot.

3.11 Naplnění submatice ze ZIG-ZAG sekvence [4]

Převedení ZIG-ZAG sekvence zpět na matici o rozměru 8x8.

3.12 Dekvantizace [4]

Výpočet probíhá pomocí kvantizačních matic vypsanych v sekci kvantizace a pomocí vzorce:

$$F(u, v) = F_Q(u, v) \cdot Q(u, v)$$

3.13 Inverzí diskretní kosinova transformace [4]

Transformace probíhá pomocí vzorců:

$$f(x, y) = \frac{1}{4} \left[\sum_{u=0}^7 \sum_{v=0}^7 C(u) \cdot C(v) \cdot F(x, y) \cdot \cos \frac{(2x+1)u\pi}{16} \cdot \cos \frac{(2y+1)v\pi}{16} \right]$$

$$C(u) = \begin{cases} \frac{\sqrt{2}}{2}, & u = 0 \\ 1, & u \neq 0 \end{cases}, \quad C(v) = \begin{cases} \frac{\sqrt{2}}{2}, & v = 0 \\ 1, & v \neq 0 \end{cases}$$

3.14 Inverzí diskretní Fourierova transformace [4]

Transformace probíhá pomocí vzorců:

$$real[F(u, v)] = amplituda[F(u, v)] \cdot \cos(Fáze[F(u, v)])$$

$$imag[F(u, v)] = amplituda[F(u, v)] \cdot \sin(Fáze[F(u, v)])$$

$$F(x, y) = \frac{1}{64} \sum_{u=0}^7 \sum_{v=0}^7 f(u, v) \cdot e^{+j2\pi(\frac{ux}{8} + \frac{vy}{8})}$$

3.15 Převod $YC_B C_R$ [4]

Zpětný převod barvového systému do RGB podle vzorců:

$$R = 1,0091 \cdot Y - 0,0032 \cdot (C_B - 128) + 1,3955 \cdot (C_R - 128)$$

$$G = 1,0091 \cdot Y - 0,3472 \cdot (C_B - 128) - 0,7206 \cdot (C_R - 128)$$

$$B = 1,0091 \cdot Y + 1,7689 \cdot (C_B - 128) - 0,0066 \cdot (C_R - 128)$$

3.16 Výpočet odchylek [4]

Střední kvadratická odchylka pro jednotlivé RGB složky je spočtena ze vzorce:

$$m_i = \sqrt{\frac{\sum_{i=0}^{m \cdot n} (z - z')^2}{m \cdot n}}$$

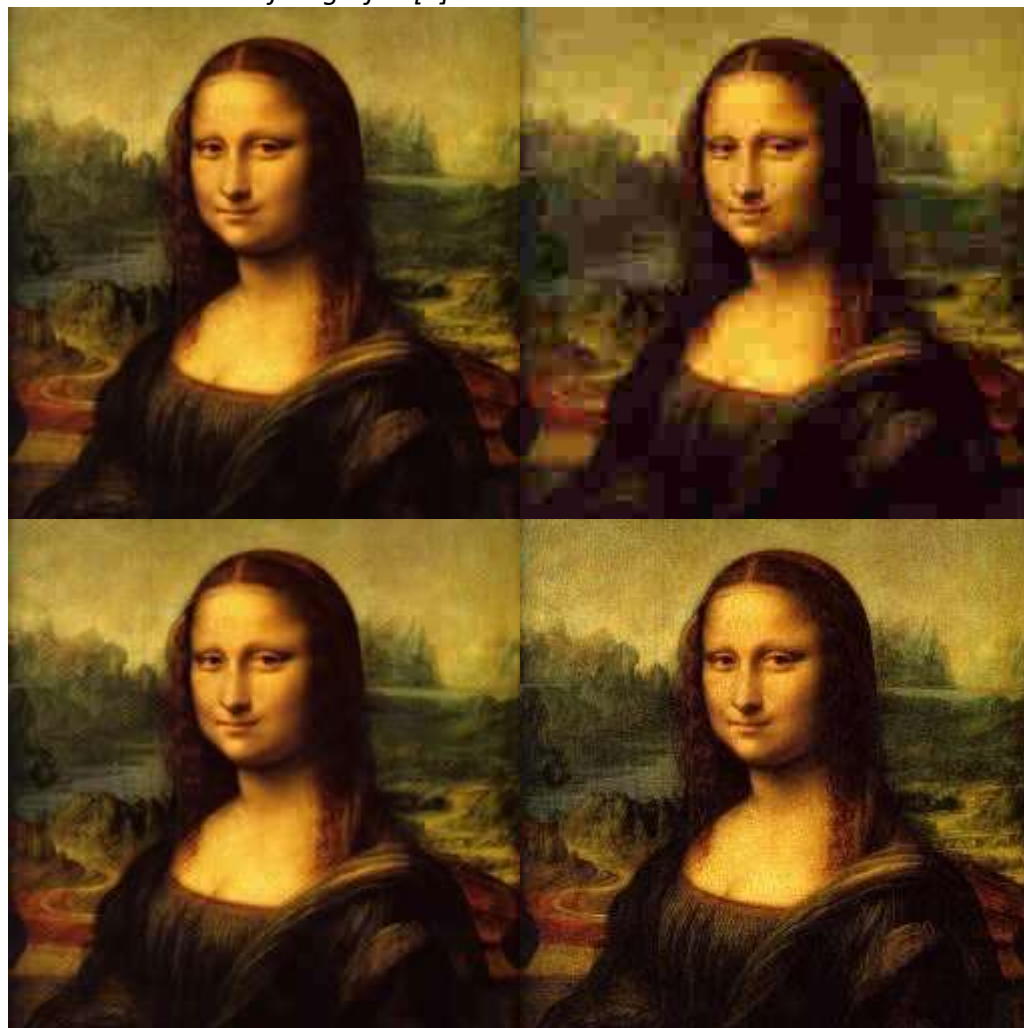
4 Výsledky

Rastry jsou vždy v pořadí, kdy první je původní snímek a následují snímky po kompresi, a to v pořadí faktoru komprese 10, 50, 70.

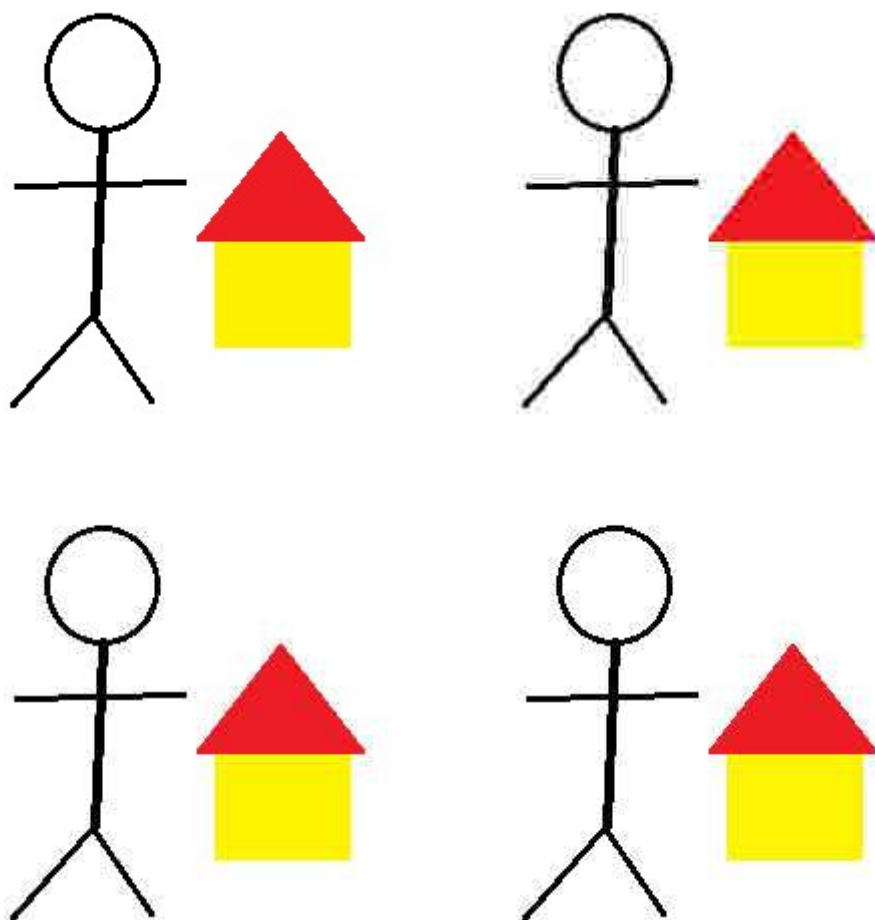
Obrázek 2: Fotografie odstínu jedné barvy. [1]



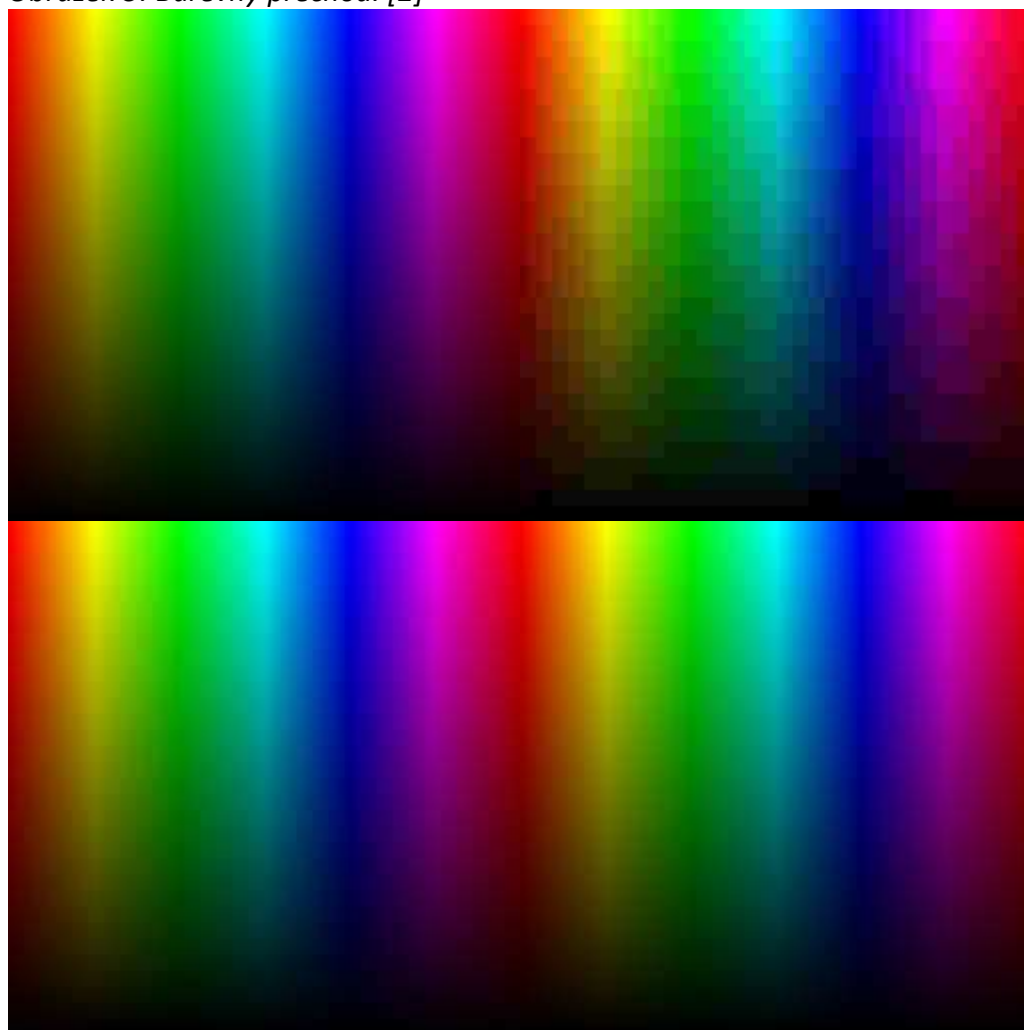
Obrázek 3: Barevná fotografie. [1]



Obrázek 4: Vektorová kresba.



Obrázek 5: Barevný přechod. [2]



Obrázek 6: Černobílý rastr. [3]



Tabulka 1: Tabulka hodnot středních kvadratických odchylek jednotlivých složek RGB pro různé rastry.

Faktor komprese q	10			50			70		
Odchylka	m _R	m _G	m _B	m _R	m _G	m _B	m _R	m _G	m _B
Fotografie odstínu jedné barvy	12,09	4,05	8,47	6,25	1,80	2,89	5,64	1,62	2,49
Barevná fotografie	9,55	8,26	11,14	4,57	4,24	5,01	4,18	3,95	4,59
Vektorová kresba	13,49	13,64	15,69	6,80	6,66	7,15	5,84	5,43	5,86
Barevný přechod	7,48	4,80	8,20	2,17	1,73	2,54	1,89	1,36	2,06
Černobílý rastr	14,98	14,98	14,98	9,14	9,14	9,14	4,76	4,76	4,76

5 Závěr

V obrázcích a tabulkách lze pozorovat, jak JPEG komprese ovlivňuje jednotlivé typy rastrů. Největší kvadratickou odchylku lze pozorovat ve vektorové kresbě. To je způsobeno ostrými přechody, mezi podkladovou barvou a jednotlivými prvky vektorové kresby. Komprese probíhá v jednotlivých sekcích a tyto sekce poté výrazně rozostřují tyto přechody. Použití pro tento typ rastru je výrazně nevhodné.

Naopak nejnižší hodnota kvadratické odchylky je u barevného přechodu, kde barvy přechází postupně. Při vysokém faktoru komprese je ovšem v obrázku výrazně vidět vytvoření sekcí.

Použití pro klasickou fotografii je tedy možné, pokud se zde nenachází výrazné přechody. Je ale potřeba si dávat pozor na použité hodnotě faktoru komprese.

Do funkce byly přidány další volitelné prvky jako řazení do ZIG-ZAG sekvence, nebo možnost volby diskrétní Fourierovy transformace místo diskrétní kosinové transformace. Pro tuto transformaci jsou však výsledky výrazně znehodnoceny, protože při kvantizaci dochází k větší ztrátě informací.

Do výpočetního skriptu nebyla zaimplementována funkce pro Huffmanovo kódování a pro zpětný postup Huffmanova kódování. Hlavní funkce obsahuje parametr pro volbu, zda má být kódování použito. Samotné kódování ale neproběhne.

6 Přílohy

- Skript pro spuštění funkce a s popisem funkce (*Koudelka_Muller_U1.m*)
- Skript funkce JPEG komprese (*jpg_komprese.m*)
- Skript s funkcí diskrétní kosinové transformace (*mydct.m*)
- Skript s funkcí inverzní diskrétní kosinové transformace (*mydctinv.m*)
- Skript s funkcí diskrétní Fourierovy transformace (*mydcft.m*)
- Skript s funkcí inverzní diskrétní Fourierovy transformace (*mydcftinv.m*)
- Skript s funkcí pro převzorkování (*resampling.m*)
- Skript s funkcí pro ZIG-ZAG sekvence (*zigzag.m*)
- Skript s funkcí pro zpětné ZIG-ZAG sekvence (*zigzag_back.m*)

7 Zdroje

[1]

https://www.google.com/search?sca_esv=583929409&sxsrf=AM9HkKmlhh0vch9PqHmPnslth63kkywmon:1700472969721&q=foto+256x256+mona+lisa&tbm=isch&source=Inms&sa=X&ved=2ahUKEwia hU26o9KCAxXa9bslHSdDAdgQ0pQJegQIDBAB&biw=1920&bih=923&dpr=1#imgsrc=6U0DMXAlmNcDjM

[2]

https://www.google.com/search?q=256x256+color+transition&tbm=isch&ved=2ahUKEwjOqfn4o9KCAxUxhCcCHa-2CGEQ2-cCegQIABAA&oq=256x256+color+transition&gs_lcp=CgNpbWcQAzoECCMQJzoGCAAQBxAeOgYIABAFEB5QnQJY-xRgiRdoAHAAeACAAUSIAc0FkgECMTKYAQCgAQGqAQtn3Mtd2l6LWltZ8ABAQ&sclient=img&ei=DCIbZc7zK7GInsEPr-2iiAY&bih=923&biw=1920#imgsrc=sgeaZSUsbhNiXM

[3]

<https://stackoverflow.com/questions/46673496/histogram-equalization-implementation-using-opencv-c>

[4]

<https://github.com/k155cvut/ygei/blob/main/prednasky/geoinf2.pdf>