

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STAVEBNÍ, OBOR GEODÉZIE A KARTOGRAFIE
KATEDRA GEOMATIKY

název předmětu

GEOINFORMATIKA

číslo
úlohy

U1

název úlohy

JPEG komprese a dekomprese rastru

školní rok

2024/2025

studijní skup.

sk05

číslo zadání

5

zpracoval

Vojtěch Skořepa
Magdaléna Soukupová

datum

16.11. 2024

klasifikace

1 Zadání

Implementujte algoritmus pro JPEG kompresi/dekompresi rastru v prostředí MATLAB (popř. v programovacím jazyce dle vlastního výběru), zahrnující tyto fáze:

- transformaci do Y CbCr modelu,
- diskrétní kosinovou transformaci,
- kvantizaci koeficientů,

a to bez využití vestavěných funkcí.

Kompresní algoritmus otestujte na různých typech rastru: rastr v odstínech šedi, barevný rastr (viz tabulka) vhodného rozlišení a velikosti (max 128x128 pixelů) s různými hodnotami faktoru komprese $q = 10, 50, 70$. Pro každou variantu spočítejte střední kvadratickou odchylkou m jednotlivých RGB složek.

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

Výsledky umístěte do přehledných tabulek pro jednotlivá q . Na základě výše vypočtených údajů zhodnoťte, ke kterým typům dat je JPEG komprese nejvíce, a naopak nejméně vhodná.

Rastr	Faktor komprese	Střední kvadratická odchylka m pro R složku	Střední kvadratická odchylka m pro G složku	Střední kvadratická odchylka m pro B složku
V odstínech šedi – čárová kresba	$q=10$	22,4981	22,3819	22,6083
	$q=50$	11,6345	11,6345	11,6345
	$q=70$	9,0692	9,0527	9,0905
Barevný – čárová kresba	$q=10$	18,5752	19,2019	19,8946
	$q=50$	11,9230	12,3873	12,4151
	$q=70$	10,1060	9,92187	10,9490
V odstínech šedi – fotografie	$q=10$	17,4865	17,4317	17,6140
	$q=50$	12,3669	12,3669	12,3669
	$q=70$	11,0466	11,0289	11,0647
Barevný – fotografie	$q=10$	11,7082	10,5392	11,9822
	$q=50$	5,5091	5,4487	5,7791
	$q=70$	4,7354	4,7159	4,9424

2 Údaje o bonusových úlohách

Resamplování ratru některou z metod.

Konverze pixelů do ZIG-ZAG sekvencí.

Huffmanovo kódování.

Náhrada DCT s využitím diskretní Fourierovy transformace.

Náhrada DCT s využitím diskretní vlnkové transformace.

3 Popis a rozbor problému

Typ formátu BMP byl vytvořen za cílem uchování černobílých a barevných obrazových dat. Soubory v tomto formátu obsahují nekomprimovaná data, takže jsou ideální pro ukládání a zobrazování vysoce kvalitních digitálních obrázků. Nevýhodou je obecně větší velikost souborů.

Za tímto účelem vznikl koncept ztrátové komprese JPEG, která odstraňuje vizuální data, jež lidské oko nevidí, a průměruje barevné odchylky. Nevýznamné změny v barvách jsou tedy odstraňovány, a naopak změny jasu jsou s co největší přesností uchovávány. To je z důvodu, že na malé změny barvy je lidské oko méně citlivé než na malé změny jasu. Výhodou je kompresí čtená menší velikost souboru, ale zhoršuje se tím kvalita obrazu.

Výše faktoru komprese q se nastavuje úroveň komprimace, to ovlivňuje množství informací, které budou kompresí zanedbány a hodnota odpovídá procentuálnímu vyjádření komprese.

JPEG komprese nepracuje s rastrem jako s celkem, ale dochází k rozdělení obrazu na submatice 8x8, postupuje zpravidla po řadách, z levého horního do pravého dolního rohu rastru. Kombinuje několik různých postupů pro zvýšení kompresního poměru. Využívá matematický aparát diskretní kosinové transformace (DCT). Tato komprese není složitá ani pro průměrně výkonný hardware.

4 Popisy metod/algorithmů formálním jazykem

4.1 Komprese

- **Separace obrazu na R, G, B složky** - Separace R, G, B složek ze vstupního obrazu. Následná komprese probíhá po složkách.

Implementace v Matlabu:

```
R = double(ras1(:,:,1));  
G = double(ras1(:,:,2));  
B = double(ras1(:,:,3));
```

- **Transformace RGB do YCBCR** – Cílem je co nejmenší poškození jasové složky. Ta bude komprimována méně než obě složky chrominační.

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

Implementace v Matlabu:

```
Y = 0.2990 * R + 0.5870 * G + 0.1140 * B;  
CB = - 0.1687 * R - 0.3313 * G + 0.5000*B + 128;  
CR = 0.5000 * R - 0.4187 * G - 0.0813*B + 128;
```

- **Rozdělení na submatice** - Rozdělení barevných složek na submatice o rozměrech 8x8. Dále zpracovávají samostatně, bez ohledu na “okolní” submatice.

Implementace v Matlabu:

```
for i = 1:8:m-7  
    for j = 1:8:n-7  
        Ys = Y(i:i+7,j:j+7);  
        CBs = CB(i:i+7,j:j+7);  
        CRs = CR(i:i+7,j:j+7);
```

- **Transformace intervalu** - Transformace hodnot barevných složek Y, CB, CR z intervalu $\{0, 255\}$ na interval $\{-255, 255\}$ (označení Y' , $C'B$, $C'R$).

$$\begin{aligned}Y' &= 2 \cdot Y - 255 \\C'_B &= 2 \cdot C_B - 255 \\C'_R &= 2 \cdot C_R - 255\end{aligned}$$

Implementace v Matlabu:

```
Y_ = 2*Y-255;  
CB_ = 2*CB - 255;  
CR_ = 2*CR - 255;
```

- **Diskrétní kosinová transformace (DCT).**

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

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

Implementace v Matlabu:

```
Ydct = myDCT(Ys);  
CBdct = myDCT(CBs);  
CRdct = myDCT(CRs);
```

- **Kvantizace DCT koeficientů.** Nejvíce ztrátová část JPEG komprese. Provádí zaokrouhlení DCT koeficientů. Cílem je vypuštění koeficientů s malou hodnotou. Používá se k tomu, aby byly odděleny nízkofrekvenční složky (které obsahují hlavní strukturu obrazu) od vysokofrekvenčních složek (detaily a šum). V JPEG kompresi se DCT aplikuje na jednotlivé bloky 8x8 pixelů, což umožňuje komprimovat obraz efektivním způsobem.

$$F_Q(u, v) = \frac{F(u, v)}{Q(u, v)}$$

Kvantizační matice $Q(u, v)$ je z důvodu různého stupně komprese jiná pro složku Y a složky CB, CR.

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

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

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

Implementace v Matlabu:

```
% kvantizační matice  
Qy = [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];  
  
Qc = [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];  
  
% kvantizace  
Yq = Ydct./Qc;  
CBq = CBdct./Qy;  
CRq = CRdct./Qy;
```

4.2 Dekomprese

– Devantizace IDCT koeficientů.

Implementace v Matlabu:

```
Ysd = Ys.*Qc;  
CBd = CBs.*Qy;  
CRd = CRs.*Qy;
```

- **Inverzní diskrétní kosinová transformace (IDCT).** IDCT je opačný proces k DCT a slouží k rekonstrukci původního obrazu z frekvenčních koeficientů. Používá se po dekompresi.

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

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

Implementace v Matlabu:

```
function Rt=myidct(R)  
Rt = R;  
  
%Output raster: rows  
for x = 0:7  
  
    %Output raster: columns  
    for y = 0:7  
  
        %Input raster: rows  
        F = 0;  
        for u = 0:7  
            %Cu  
            if u == 0  
                Cu = sqrt(2)/2;  
            else  
                Cu = 1;  
            end  
  
            %Input raster: columns  
            for v = 0:7  
                %Cv  
                if v == 0  
                    Cv = sqrt(2)/2;  
                else  
                    Cv = 1;  
                end  
  
                F=F+1/4*Cv*(R(u+1, v+1)*cos((2*x+1)*u*pi/16)*cos((2*y+1)*v*pi/16));  
            end  
        end  
  
        %Output raster  
        Rt(x+1,y+1) = F;  
    end  
end  
end
```

- **Zpětná transformace intervalu** $\langle -255, 255 \rangle$ na interval $\langle 0, 255 \rangle$.

$$Y = 0,5 \cdot (Y' - 255)$$

$$C_B = 2 \cdot C'_B - 255$$

$$C_R = 2 \cdot C'_R - 255$$

- **Zpětná konverze mezi modely** Po aplikaci IDCT a dekvantizaci jsou složky obrazu Y, Cb a Cr převedeny zpět na RGB barevný model.

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1,000 & 0,000 & 1,4020 \\ 1,000 & -0,3441 & -0,7141 \\ 1,000 & 1,7720 & -0,0001 \end{pmatrix} \begin{pmatrix} Y \\ C_B - 128 \\ C_R - 128 \end{pmatrix}$$

Implementace v Matlabu:

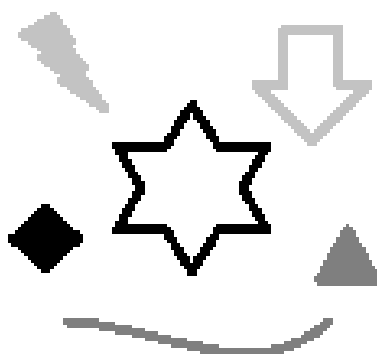
```
Rd = Y + 1.4020*(Cr-128);
```

```
Gd = Y - 0.3441*(Cb-128) - 0.7141*(Cr-128);
```

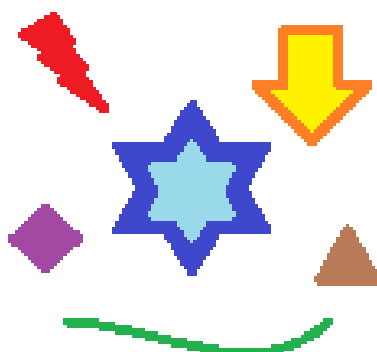
```
Bd = Y + 1.7720 * (Cb-128) - 0.0001*(Cr-128);
```

5 Vstupní data

Vstupními daty jsou dle zadání dva typy rastrů, dva rastry v odstínech šedi a druhé dva barevné. Rastry jsou ve formátu .bpm, jedná se o formát pro ukládání rastrové grafiky u kterých není použita žádná komprese. Což v rámci této úlohy, ve které testujeme jpeg kompresi, potřebujeme. Všechny rastry mají velikost 128 x 128 pixelů.



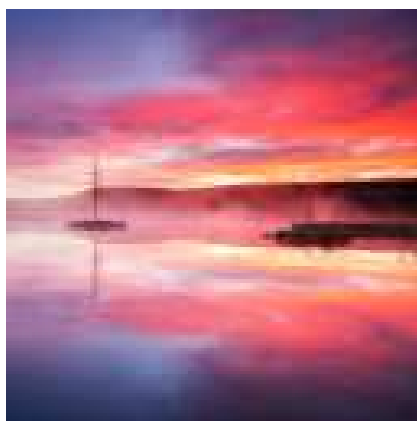
Obrázek 1 Vybraný vstupní rastr čárové kresby v odstínech šedi



Obrázek 2 Vybraný vstupní barevný rastr čárové kresby

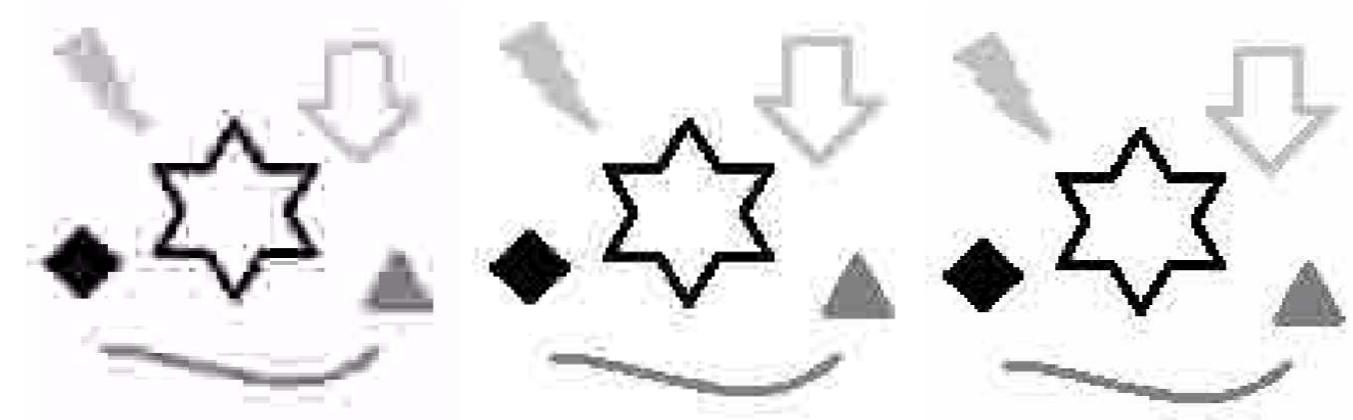


Obrázek 3 Vybraný vstupní rastr fotografie krajiny v odstínech šedi



Obrázek 4 Vybraný vstupní barevný rastr fotografie krajiny

6 Výstupní data



Obrázek 5 Rastr v odstínech šedi komprimovaný s kompresním faktorem 10

Obrázek 6 Rastr v odstínech šedi komprimovaný s kompresním faktorem 50

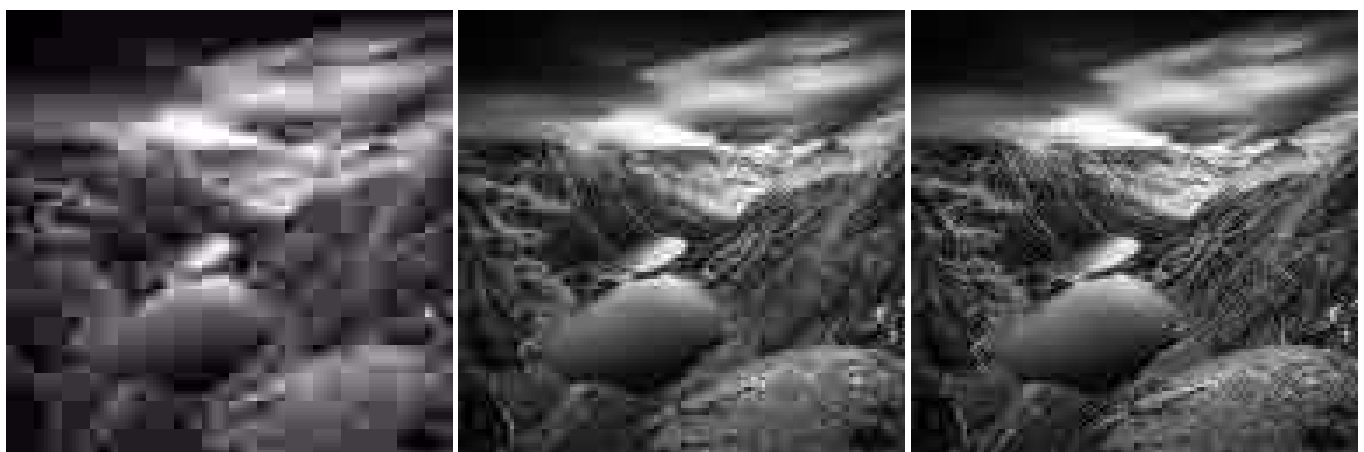
Obrázek 7 Rastr v odstínech šedi komprimovaný s kompresním faktorem 70



Obrázek 8 Barevný rastr komprimovaný s kompresním faktorem 10

Obrázek 9 Barevný rastr komprimovaný s kompresním faktorem 50

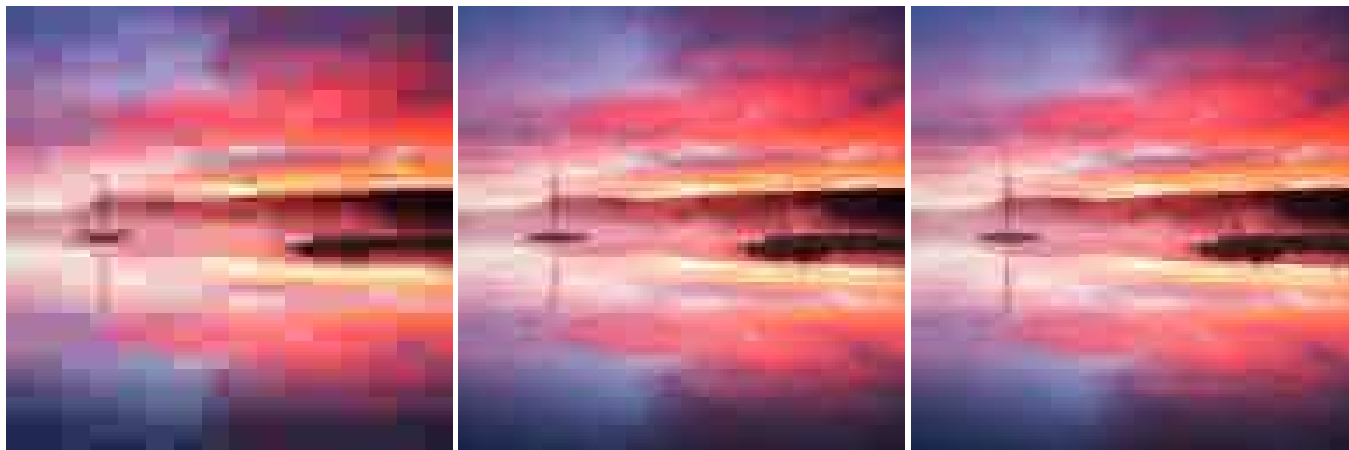
Obrázek 10 Barevný rastr komprimovaný s kompresním faktorem 70



Obrázek 11 Rastr v odstínech šedi s kompresním faktorem 10

Obrázek 12 Rastr v odstínech šedi s kompresním faktorem 50

Obrázek 13 Rastr v odstínech šedi s kompresním faktorem 70



Obrázek 14 Barevný rastr komprimovaný s kompresním faktorem 10

Obrázek 15 Barevný rastr komprimovaný s kompresním faktorem 50

Obrázek 16 Barevný rastr komprimovaný s kompresním faktorem 70

7 Závěr

JPEG komprese je široce využívána pro zmenšení velikosti obrazových souborů, což přináší řadu výhod, jako je rychlejší přenos dat, úspora místa na disku. Využití diskretní kosinové transformace (DCT) umožňuje efektivní kompresi tím, že se odstraní méně důležité vysokofrekvenční složky obrazu. Nevýhodou však je ztrátovost komprese, což při vyšších úrovních komprese může vést k viditelným artefaktům a ztrátě detailů. I přesto JPEG nabízí vyvážený kompromis mezi kvalitou a velikostí souboru, a proto je stále klíčovým formátem v praxi.

Obrázky s čistými hranami a liniemi ztratí při kompresi část své ostrosti. Další problémy způsobuje komprese u barevných obrázků, kde na ostrých hranách dochází k degradaci dat. Jednou z nedokonalostí je vznik moiré efektu, což je rušivý optický efekt, který vzniká právě u barevných přechodů. Charakteristikou JPEG komprese je, že při ní dochází k rozdělení obrazu na submatice 8x8, jež jsou při vysokém faktoru komprese viditelné a ovlivňuje to výslednou kvalitu komprimovaného obrazu.