ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STAVEBNÍ, OBOR GEODÉZIE A KARTOGRAFIE

KATEDRA GEOMATIKY

GEOINFORMATIKA

Úloha 1 – JPEG komprese rastru

Rok	**Semestr**	**Skupina**	**Vypracovali**	**Datum**
2025/2026	ZS		František Gurecký Vítek Veselý	14.10.2025

Zadání

Implementujte JPEG kompresi rastru bez využití vestavěných funkcí.

Kompresní algoritmus otestujte na různých typech rastru a s různými faktory komprese q.

Pro každou variantu spočtěte střední kvadratickou odchylku m jednotlivých RGB složek.

Na základě vypočtených údajů zhodnoťte, ke kterým typům dat je JPEG komprese nejvíce a naopak nejméně vhodná.

Bonusové úlohy

- Resamplování rastru
- Konverze pixelů do ZIG-ZAG sekvencí
- Huffmanovo kódování
- Náhrada DCT s využitím DFT
- Náhrada DCT s využitím DWT

Popis problému

Metoda JPEG je druh transformační komprese, která je ztrátová. Tyto metody využívají relativně malé citlivosti lidského oka na změny barev. Lze tedy upravit barvy tak, aby se dal rastr efektivněji uložit do paměti za cenu jisté ztráty dat, avšak bez ztráty vizuálního vjemu. Cílem této úlohy je implementovat tento druh komprese, včetně zpětné dekomprese.

Popis algoritmů

Na začátku celého procesu je načten obrazový soubor, který je následně rozložen na jednotlivé složky R, G a B. Tyto složky jsou transformovány do barevného systému $YCbCr$, jenž odděluje jasovou složku od barevných složek a tím umožňuje efektivnější kompresi obrazu.

V dalším kroku je proveden *resampling*, při němž jsou hodnoty určitého počtu pixelů nahrazeny jejich průměrnou hodnotou. Tento krok snižuje množství dat, která je nutné dále zpracovávat, a umožňuje dosažení vyšší míry komprese při zachování přijatelné kvality výsledného obrazu.

Následující část algoritmu pracuje po blocích o velikosti 8×8 pixelů. Pro každý blok je aplikována jedna z implementovaných transformací — *diskrétní kosinová transformace*

(DCT), *diskrétní Fourierova transformace (DFT)* nebo *diskrétní vlnková transformace (DWT)*. Volba konkrétní transformace je v implementaci určena parametrem na řádku 69 skriptu.

Po provedení transformace jsou hodnoty kvantizovány pomocí kvantizačních matic, které zvýrazní dominantní členy signálu a potlačí méně významné složky. Kvantizované hodnoty jsou následně zaokrouhleny a uloženy do výsledné matice.

Takto zpracovaný rastr je poté převeden do jednorozměrné formy pomocí tzv. „cik-cak“ průchodu (angl. *zig-zag*). Tento algoritmus prochází prvky matice diagonálně a přeskupuje je do jednoho vektoru. Implementace je optimalizována tak, že hodnoty zapisuje z obou konců výsledného pole současně, čímž je každý prvek zpracován pouze jednou. Díky tomu je průchod přibližně dvakrát rychlejší než klasický sekvenční algoritmus.

Na získaný vektor hodnot je následně aplikováno *Huffmanovo kódování*. Nejprve jsou spočteny četnosti jednotlivých symbolů, podle nichž je zkonstruován binární strom. V každém kroku jsou dvě nejméně pravděpodobné položky sloučeny do jednoho nadřazeného uzlu, dokud nezůstane jediný kořen stromu. Každému průchodu stromem je přiřazena binární hodnota - 1 pro levou, 0 pro pravou větev - čímž vznikne unikátní binární kód pro každý symbol. Častěji se vyskytující symboly získají kratší kód, zatímco méně časté delší. Výsledkem je mapa kódů, která se použije k převodu původních hodnot na komprimovanou bitovou sekvenci. V této implementaci je výsledná sekvence uložena jako řetězec, což má spíše demonstrační než paměťově úsporný charakter.

Při dekompresi jsou prováděny kroky inverzní ke všem výše uvedeným — tedy dekódování Huffmanova kódu, inverzní „cik-cak“ uspořádání a inverzní transformace (IDCT, IDFT nebo IDWT). Po zpětné transformaci do systému *RGB* vznikne dekomprimovaný obraz, u něhož je možné vyhodnocovat chyby pomocí střední kvadratické odchylky v jednotlivých složkách. Malé hodnoty těchto chyb znamenají, že výsledná ztráta kvality obrazu je minimální.

Použité vzorce

Rovnice `DCT`:

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

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

Rovnice IDCT:

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

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

Rovnice DFT:

$$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

Rovnice IDFT:

$$f(x,y)=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

Funkce pro DWT:

```
function imgt = dwt2(img)
    % Horizontal transformation
    L = (img(:,1:2:end) + img(:,2:2:end)) / 2;
    H = (img(:,1:2:end) - img(:,2:2:end)) / 2;

    % Vertical transformation
    LL = (L(1:2:end,:) + L(2:2:end,:)) / 2;
    LH = (L(1:2:end,:) - L(2:2:end,:)) / 2;
    HL = (H(1:2:end,:) + H(2:2:end,:)) / 2;
    HH = (H(1:2:end,:) - H(2:2:end,:)) / 2;

    % Output
    imgt = [LL, LH; HL, HH];
end
```

Funkce pro IDWT:

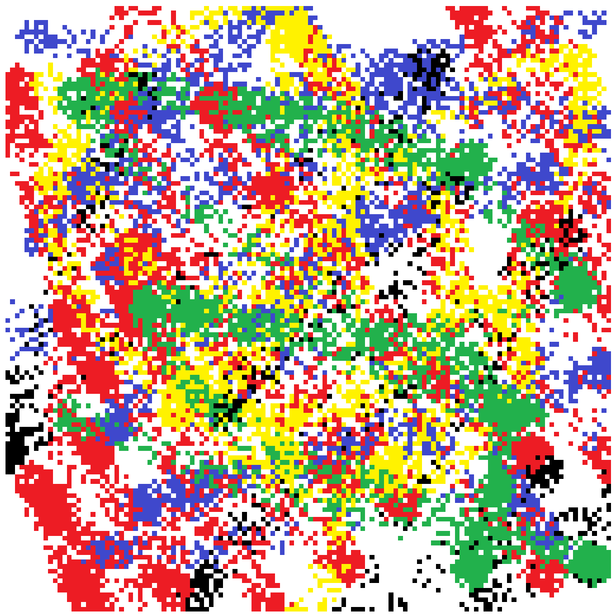
```
function img = idwt2(coef)
    [M, N] = size(coef);

    % Split into sub-blocks
    half_rows = M/2; half_cols = N/2;
    LL = coef(1:half_rows, 1:half_cols);
    LH = coef(1:half_rows, half_cols+1:end);
```

```
    HL = coef(half_rows+1:end, 1:half_cols);
    HH = coef(half_rows+1:end, half_cols+1:end);

    % Reconstruct verticaly
    L = zeros(M, half_cols);
    H = zeros(M, half_cols);
    L(1:2:end,:) = LL + LH;
    L(2:2:end,:) = LL - LH;
    H(1:2:end,:) = HL + HH;
    H(2:2:end,:) = HL - HH;

    % Reconstruct horizontaly
    img = zeros(M, N);
    img(:,1:2:end) = L + H;
    img(:,2:2:end) = L - H;
end
```

Výsledky

Tabulka 1: Tabulka pro barvicky.png

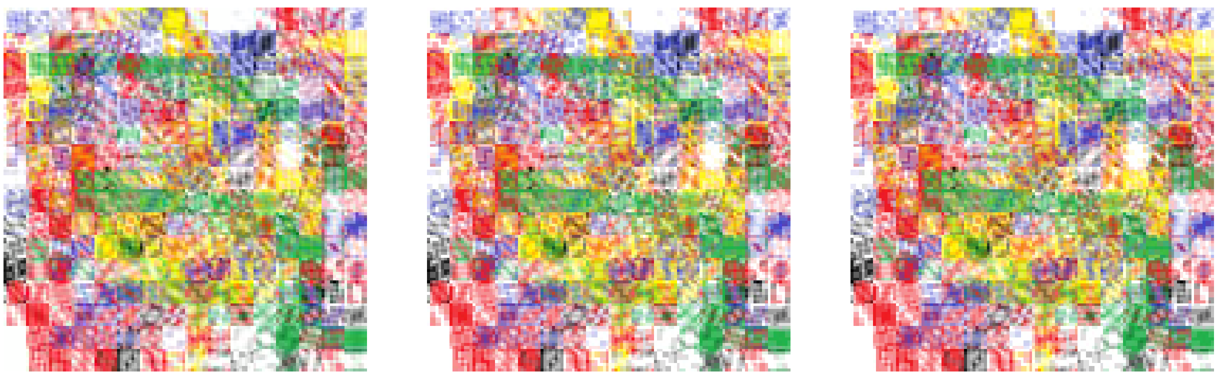
	DCT			DFT			DWT		
q	σ_R	σ_G	σ_B	σ_R	σ_G	σ_B	σ_R	σ_G	σ_B
10	73,267	77,449	84,687	77,724	84,395	89,372	147,241	114,530	164,743
50	69,562	74,956	80,409	76,677	83,884	88,125	70,587	75,113	81,897
70	69,081	74,567	79,352	76,647	83,884	88,180	68,763	74,494	80,301



Obrázek 1: Originální obrázek - barvicky.png

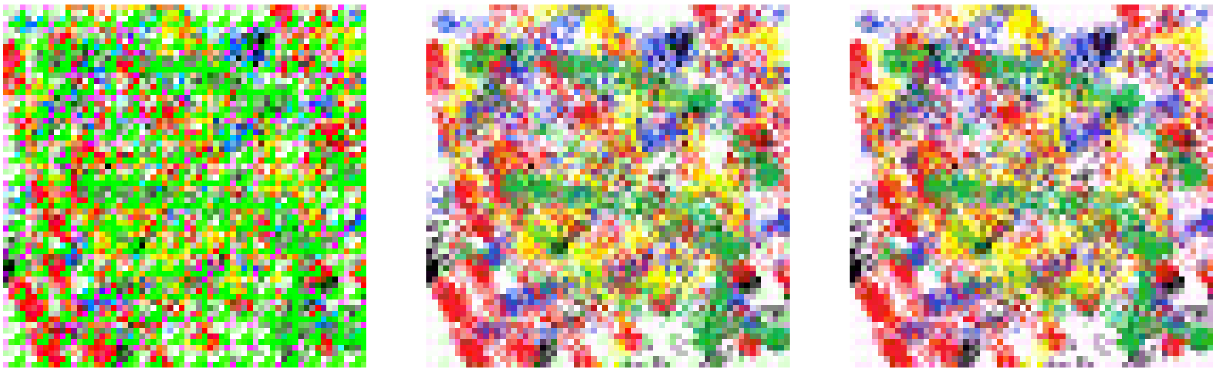
(a) q = 10 (b) q = 50 (c) q = 70

Obrázek 2: DCT



(a) q = 10 (b) q = 50 (c) q = 70

Obrázek 3: DFT

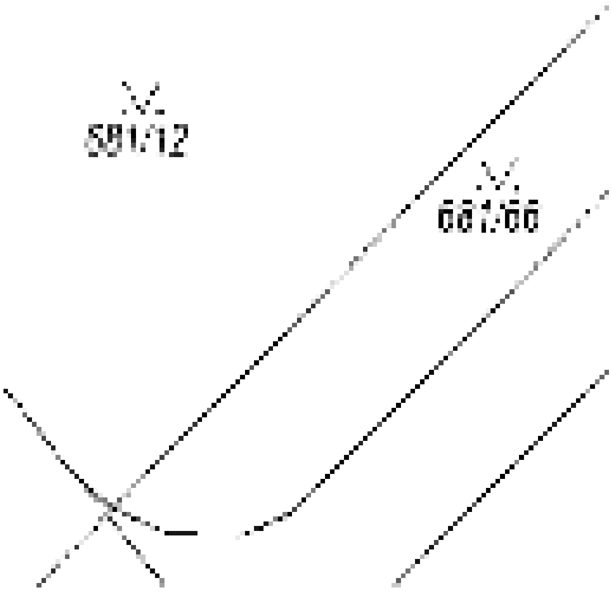


(a) q = 10 (b) q = 50 (c) q = 70

Obrázek 4: DWT

Tabulka 2: Tabulka pro dkm.png

	DCT			DFT			DWT		
q	σ_R	σ_G	σ_B	σ_R	σ_G	σ_B	σ_R	σ_G	σ_B
10	26,738	26,886	26,721	28,721	28,718	28,727	129,161	99,587	161,176
50	25,614	25,643	25,614	28,609	28,609	28,609	31,028	31,564	34,396
70	25,601	25,619	25,603	28,596	28,596	28,596	28,657	27,705	30,428



Obrázek 5: Originální obrázek - dkm.png

(a) q = 10

(b) q = 50

(c) q = 70

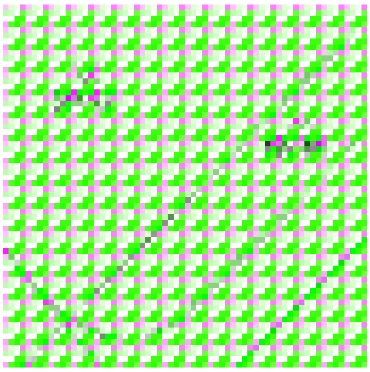
Obrázek 6: DCT

(a) q = 10

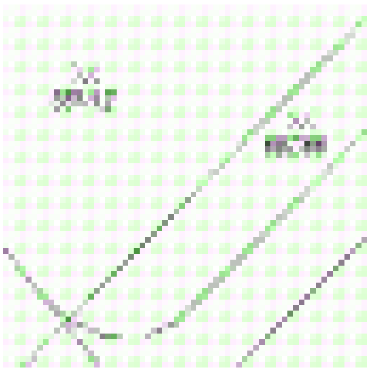
(b) q = 50

(c) q = 70

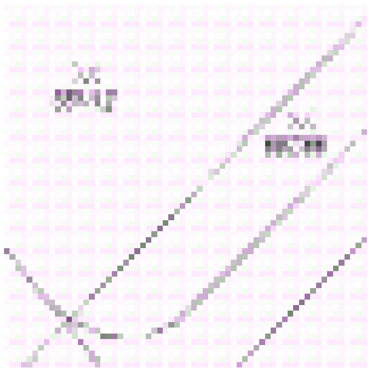
Obrázek 7: DFT



(a) q = 10



(b) q = 50



(c) q = 70

Obrázek 8: DWT

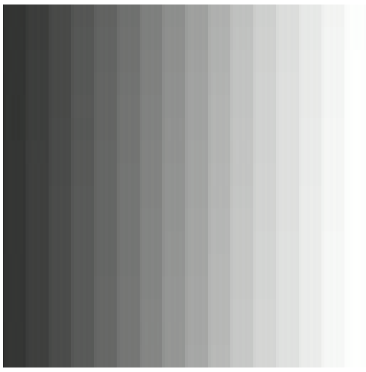
Tabulka 3: Tabulka pro gradient.png

	DCT			DFT			DWT		
q	σ_R	σ_G	σ_B	σ_R	σ_G	σ_B	σ_R	σ_G	σ_B
10	4,283	4,075	4,343	3,458	3,438	3,499	125,904	98,421	158,293
50	1,105	1,177	1,208	3,242	3,241	3,242	21,385	16,096	26,693
70	1,055	0,969	1,183	3,238	3,238	3,239	13,841	10,827	17,300

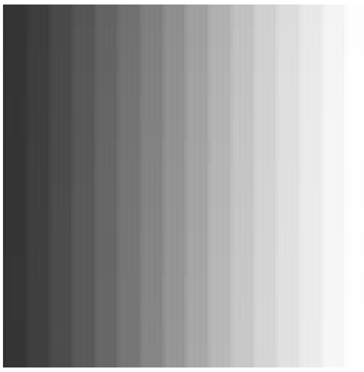
Obrázek 9: Originální obrázek - gradient.png

(a) q = 10 (b) q = 50 (c) q = 70

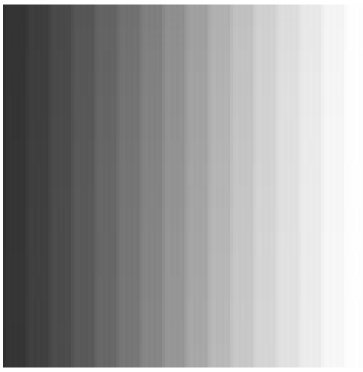
Obrázek 10: DCT



(a) q = 10

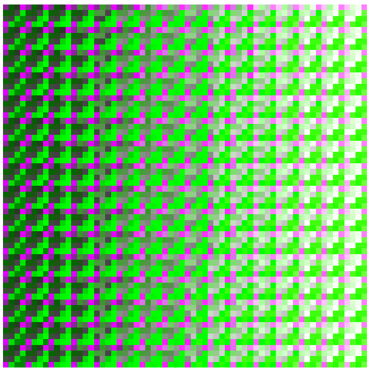


(b) q = 50

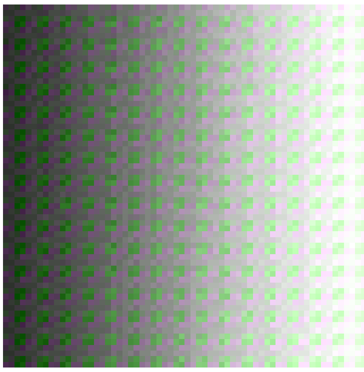


(c) q = 70

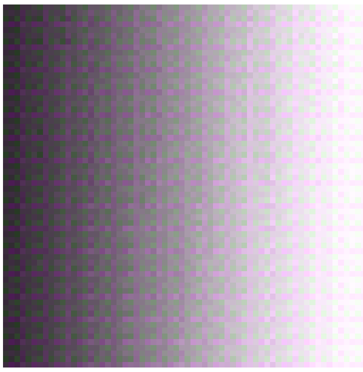
Obrázek 11: DFT



(a) q = 10



(b) q = 50

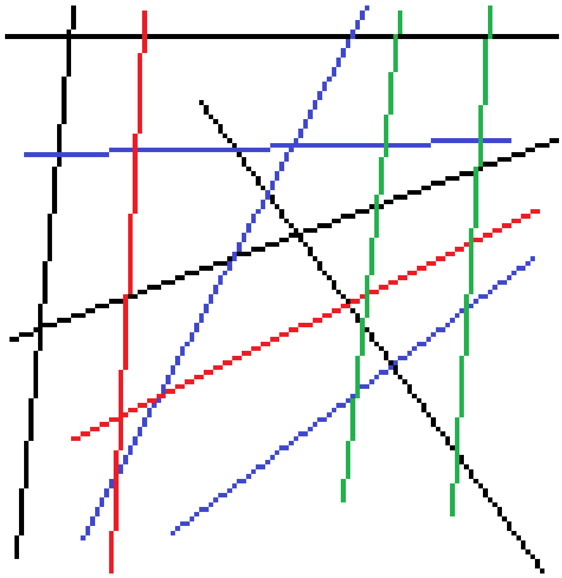


(c) q = 70

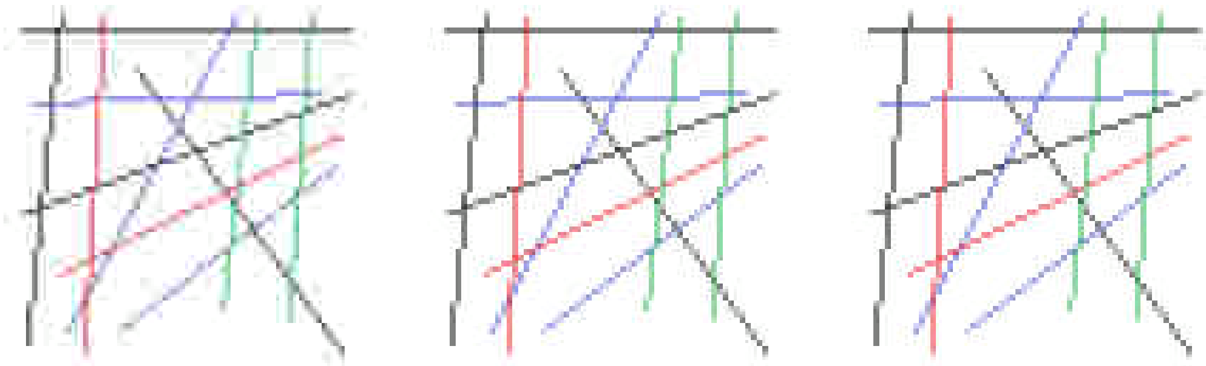
Obrázek 12: DWT

Tabulka 4: Tabulka pro rovne.png

	DCT			DFT			DWT		
q	σ_R	σ_G	σ_B	σ_R	σ_G	σ_B	σ_R	σ_G	σ_B
10	42,602	42,492	41,425	46,558	47,121	45,368	133,542	102,996	164,415
50	39,960	40,196	38,838	46,166	46,947	44,991	43,268	43,855	44,905
70	39,849	40,147	38,772	46,153	46,938	44,983	41,779	41,399	41,823

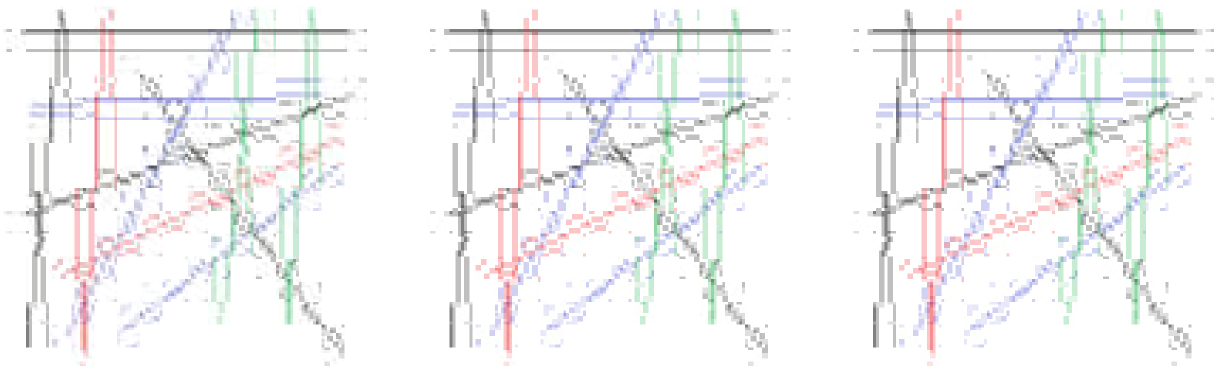


Obrázek 13: Originální obrázek - rovne.png



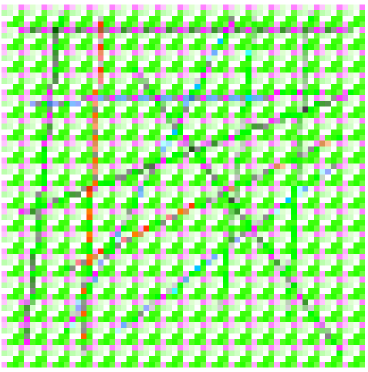
(a) q = 10 (b) q = 50 (c) q = 70

Obrázek 14: DCT

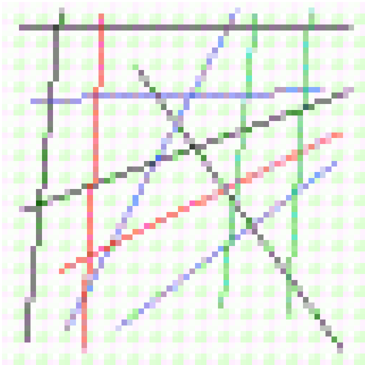


(a) q = 10 (b) q = 50 (c) q = 70

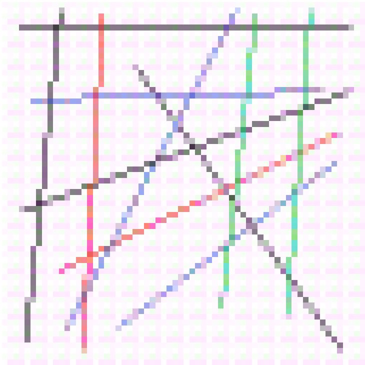
Obrázek 15: DFT



(a) q = 10



(b) q = 50



(c) q = 70

Obrázek 16: DWT

Tabulka 5: Tabulka pro text.png

	DCT			DFT			DWT		
q	σ_R	σ_G	σ_B	σ_R	σ_G	σ_B	σ_R	σ_G	σ_B
10	15,172	15,499	15,130	20,512	20,507	20,519	127,296	97,676	159,638
50	14,819	14,880	14,818	20,413	20,413	20,413	23,313	23,948	27,648
70	14,861	14,905	14,863	20,418	20,418	20,419	19,968	18,621	22,429

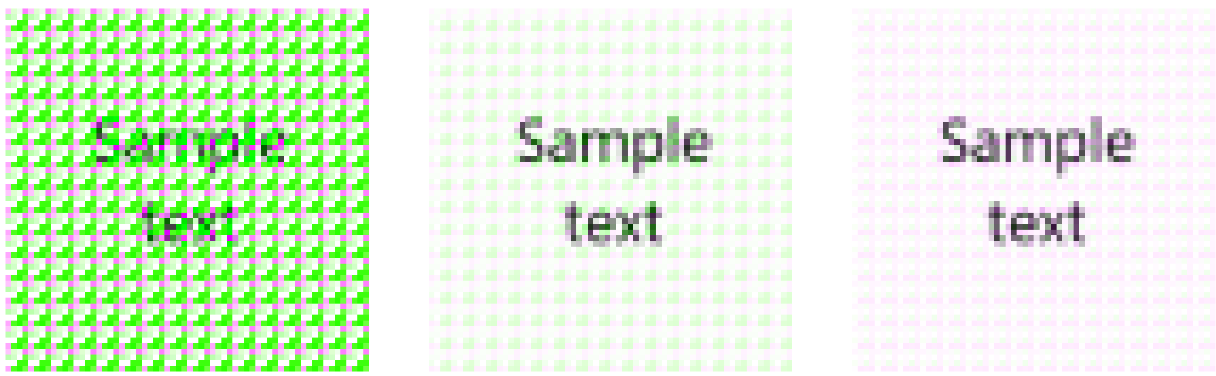
Obrázek 17: Originální obrázek - text.png

(a) q = 10 (b) q = 50 (c) q = 70

Obrázek 18: DCT

(a) q = 10 (b) q = 50 (c) q = 70

Obrázek 19: DFT



(a) q = 10 (b) q = 50 (c) q = 70

Obrázek 20: DWT

Závěr

Z jednotlivých příkladů je patrné, že při použití kompresního koeficientu q = 10 dochází k největší ztrátě přesnosti oproti originálnímu obrazu. Zároveň lze pozorovat určité trendy v kvalitě rekonstrukce mezi jednotlivými transformacemi. Nejlepších výsledků ze zadaných transformací dosáhla DCT, u které jsou směrodatné odchylky téměř ve všech případech nejmenší. Již na první pohled na rekonstruované obrázky je patrné, že tato transformace poskytuje nejkvalitnější výsledky.

Při použití DFT je patrné, jak se při kompresi a následné dekompresi uplatňují bloky o velikosti 8×8. Transformace si nejlépe poradila s postupným gradientem – přechody jsou sice viditelné, ale probíhají plynule. Nejhůře si DFT poradila s liniovými obrazci a s oblastmi, kde se často střídají barvy.

Transformace DWT se ze všech použitých metod neosvědčila nejlépe, což je nejmarkantnější při kompresním faktoru q = 10. V tomto případě se obraz zaplní pixely se zelenou barvou, takže původní obraz je prakticky nerozeznatelný. Při zvýšení kompresního faktoru na q = 70 se kvalita komprese a následné dekomprese zlepšila, přesto však zůstává výrazně horší než u DCT s q = 10, kromě případů, kdy dochází k časté změně barev. Jak je patrné na prvním příkladu, zde si DWT vedla velmi dobře a poskytuje výsledky srovnatelné s DCT.

V Praze dne 14.10.2025

František Gurecký
Vítek Veselý