

DCT kompresija slik

Diskretna kosinusna transformacija

DCT preslika kvadratni blok amplitud signala (v našem primeru barv piksla) v blok koeficientov DCT enakih dimenzij v frekvenčnem prostoru. Standard JPEG, ki prav tako uporablja DCT, uporablja bloke velikosti 8×8 pikslov, čemur sta prilagojeni tudi spodnji formuli.

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

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

Oznaka $f(x,y)$ je originalna vrednost obravnavane barvne komponente piksla, $F(u,v)$ transformirana vrednost po izvedbi DCT, $C(0)=1/\sqrt{2}$ in $C(i)=1$ za $i = 1, \dots, 7$. DCT daje boljše rezultate, če je zaloga vrednosti amplitud simetrično porazdeljena okoli 0. Kateri rezultati DCT so boljši in kateri so slabši? Boljši so tisti, ki omogočajo boljše stiskanje s kodiranjem npr. s tekočo dolžino. Naš cilj je prideliti čim več 0 v matriki koeficientov. Izkaže se, da se najpomembnejši koeficienti nahajajo v levem zgornjem koncu matrike. Le-ti predstavljajo nižje frekvence v signalu. Še posebej je pomemben koeficient v skrajnem levem zgornjem kotu, ki ima tudi posebno ime - koeficient DC. Ostale koeficiente imenujemo koeficiente AC. Z oddaljenostjo od levega zgornjega kota se njihove absolutne vrednosti običajno manjšajo, vse več jih je 0, lahko pa z uporabo faktorja stiskanja tudi umetno postavimo določeno število koeficientov v desnem spodnjem kotu matrike na 0. Faktor stiskanja interaktivno vnese uporabnik (to prakso poznamo iz JPEG). To je dobro upoštevati že pri sami implementaciji FDCT, da se izognemo nepotrebnemu računanju.

Kvantizacija

15	14	13	12	11	10	9	8
14	13	12	11	10	9	8	7
13	12	11	10	9	8	7	6
12	11	10	9	8	7	6	5
11	10	9	8	7	6	5	4
10	9	8	7	6	5	4	3
9	8	7	6	5	4	3	2
8	7	6	5	4	3	2	1

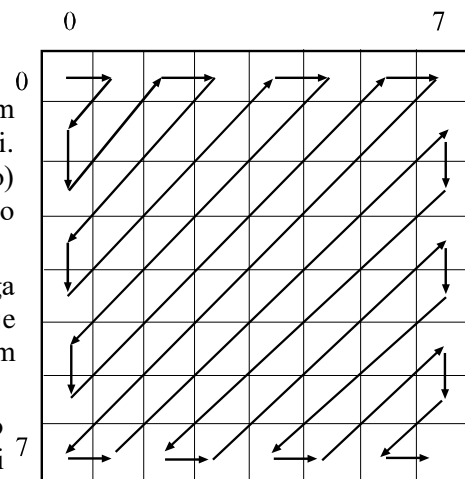
Dogovorimo se, da bo faktor stiskanja celoštevilaska vrednost v intervalu [0,15]. Vrednost 0 pomeni, da pustimo matriko nespremenjeno, vrednost 15 pa, da prav vse koeficiente vključno s koeficientom DC postavimo na 0 (resetiramo). Spodnja slika prikazuje, na katere koeficiente v matriki vplivajo različni faktorji stiskanja. Prikazane so le meje trikotnega območja, znotraj katerega postavljamo koeficiente na 0, vedno pa resetiramo tudi vse koeficiente, ki so desno in spodaj od mejnega. Če je npr. izbran faktor stiskanja 8, postavimo na 0 vse koeficiente v poljih, ki so na sliki označena z 1, 2, 3, 4, 5, 6, 7 ali 8.

Algoritem in izhodni format

Na začetek datoteke shranimo širino in višino slike (v piksljih), podatka, iz katerih lahko pri restavriranju dobimo število blokov 8×8 v horizontalni in vertikalni smeri. Oba podatka naj bosta dolžine po 2 zloga. Sledijo stisnjeni bloki v zaporedju od leve proti desni in od zgoraj navzdol.

Stiskanje posameznega bloka (seveda za posamezni barvni kanal) poteka po naslednjem algoritmu:

1. Izvedite FDCT, nato upoštevajte faktorja stiskanja.
2. Koeficiente DCT zaokrožimo v celoštevilске vrednosti.
3. Zaokrožene koeficiente DCT uredimo v enodimenzionalni seznam v zaporedju cik-cak, ki ga tvorimo, kot prikazuje slika na desni. Prednost tega načina je, da bomo na koncu seznama (verjetno) dobili daljši podseznam samih ničel, ki ga bomo lahko učinkovito kodirali.
4. Sedaj začnemo s tvorbo izhodnega (kodiranega) niza, kakršnega bomo vpisali v datoteko. Najprej kodiramo koeficient DC. Le-ta je v intervalu $[-2040, 2040]$ in za predstavitev v dvojiškem komplementu potrebujemo 12 bitov.
5. Nadaljevanje bomo kodirali s tekočo dolžino, kjer se bomo osredotočili le na zaporedja ničel. Koeficiente AC bomo zakodirali v obliki zaporedja podseznamov, ki so v eni izmed naslednjih treh oblik:



- a)

0	Tekoča dolžina	<i>dolžina</i>	AC
---	----------------	----------------	----

1 bit 6 bitov 4 biti $|dolžina|$ bitov
- b)

0	Tekoča dolžina
---	----------------

1 bit 6 bitov
- c)

1	<i>dolžina</i>	AC
---	----------------	----

1 bit 4 biti $|dolžina|$ bitov

Tekoča dolžina je število zaporednih ničel v seznamu cik-cak. Ker vsebuje blok največ 63 koeficientov AC, potrebujemo za kodiranje brez predznaka 6 bitov. Po želji je tu možna tudi manjša izboljšava. Tako kodirnik kot dekodirnik v vsakem trenutku veda, koliko koeficientov AC je bilo že obdelanih oziroma koliko jih še manjka do 63. Zato tekoče dolžine ne bi bilo treba kodirati v formatu fiksne dolžine (6 bitov), ampak le s številom bitov, ki jih zahteva razlika $63 - |\text{število že obdelanih koeficientov AC}|$.

AC je vrednost koeficienta AC, ki sledi zaporedju ničel oz. predhodno kodiranemu neničelnemu koeficientu. Zanesljivo je v obsegu $[-4080, 4080]$ in za predstavitev v dvojiškem komplementu potrebujemo maksimalno 13 bitov. Natančno število bitov nam pove vrednost v polju **dolžina**, ki se vedno nahaja pred poljem AC. Vrednost med 1 in 13 lahko zakodiramo v formatu fiksne dolžine s 4 biti. Prvi bit v takšnem podseznamu (vrednost **0** ali **1**) pove, ali v zaporedju cik-cak sledi podzaporedje ničel (kodiramo podseznam tipa a ali b) ali sledi neničelni koeficient (tip c). Povejmo še, da se podniz tipa b lahko nahaja le na koncu zaporedja, torej v primeru, ko je skrajni desni spodnji koeficient AC v matriki koeficientov enak 0 (kar je zelo pogosto).

6. Kako zaznamo konec bloka 8×8 ? Enostavno štejemo koeficiente DCT, ki smo jih že zakodirali oz. dekodirali, in končamo (oz. se lotimo naslednjega bloka 8×8), ko jih naštejemo 64.
 - a) Koeficient DC prispeva k vsoti 1.
 - b) Podseznam tipa a prispeva k vsoti $|\text{tekoča dolžina}| + 1$.
 - c) Podseznam tipa b prispeva k vsoti $|\text{tekoča dolžina}|$.
 - č) Podseznam tipa c prispeva k vsoti 1.

PRIMER

Vhodne originalne vrednosti pikslov.

5	5	5	5	45	45	45	45
5	5	5	5	45	45	45	45
5	5	5	5	45	45	45	45
5	5	5	5	45	45	45	45
5	5	5	5	45	45	45	45
5	5	5	5	45	45	45	45
5	5	5	5	45	45	45	45
5	5	5	5	45	45	45	45

Piksle kodiramo z FDCT, kjer dobljene koeficiente zaokrožimo. Kvantizacije ne uporabimo v tem primeru.

200	-145	0	51	0	-34	0	29
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	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

Po zaporedju cik-cak kodiramo koeficiente v binarni tok.

	200	-145	0	0	0	0	51	0	0	0	0	0	0	0	0	-34	→
→	0	0	0	0	0	0	0	0	0	0	0	0	29	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	0	0	0	0	

Kodiranje poteka tako:

Postopek	Vrednost	Št. potrebnih bitov	Št. koeficientov
2. Koeficient DC (200) zahteva 12 bitov.	200	12 = 12 bitov	(1 koeficient)
3. -145 zakodiramo po tipu c)	1 9 -145	1+4+9 = 14 bitov	(2 koeficienta)
4. Štiri ničle in 51 zakodiramo po tipu a)	0 4 7 51	1+6+4+7 = 18 bitov	(7 koeficientov)
5. Osem ničel in -34 zakodiramo po tipu a)	0 8 7 -34	1+6+4+7 = 18 bitov	(16 koeficientov)
6. Dvanajst ničel in 29 zakodiramo po tipu a)	0 12 6 29	1+6+4+6 = 17 bitov	(29 koeficientov)
7. Petintrideset ničel zakodiramo po tipu b)	0 35	1+6 = 7 bitov	(64 koeficientov)

Skupaj potrebujemo 86 bitov oz. **11 zlogov**, v BMP brez kompresije pa **64 zlogov**.