

Sajbben Attila

Képtelefon szimuláló szoftver

Tankör: HIPI E6 (Távközlési Informatika)
Kandó Kálmán Műszaki Főiskola (Budapest)

1997

TARTALOM

Bevezetés

Digitális képfeldolgozás és képátvitel

Tér-frekvencia transzformációk

Kvantálás

Forrás kódolás

Mozgás-kompenzáció

Látási és képtechnikai alapismeretek

Különböző képtömörítési eljárások rövid áttekintése

Részsávkódolás

Transzformációs kódolások

Matematikai alapok

Fourier sor

Diszkrét Fourier transzformáció (DFT)

A DFT kapcsolata a vizsgált függvény harmonikus összetevőivel.

Transzformációs kódolások

A Karhunen-Loeve transzformáció

Diszkrét koszinusz transzformáció

A DCT definíciója a JPEG szabvány szerint

A DCT gyakorlati megvalósítása

Az FFT alap algoritmus

DCT algoritmus az FFT mintájára

A képtelefonok tömörítésének bemutatása

A normalizálás (kvantálás)

DC-kódolás és cikk-cakk szekvencia

Entrópia kódolás

Mozgáskompenzáció

Nemzetközi szabvány a képtelefon megvalósítására

A szimulációs szoftver

Összefoglaló

Summary

Irodalomjegyzék

Melléklet

Bevezetés

A digitális televízió technika, a számítástechnikai alkalmazások és a hálózaton történő átvitel lehetőségeinek kialakulásával a képadat tömörítés jelentősége folyamatosan növekszik. Fő alkalmazási területe a távközlés és a digitális média tárolás.

Több mint egy évszázada a távbeszélő hálózat jelenti az első számú nemzetközi távközlési infrastruktúrát. Az első távbeszélő rendszeren nem lehetett adatot, faxot stb. átvinni. Az átviteli rendszerek tökéletesítése, a központok korszerűsítése, a digitális hálózatok kialakításának irányába mutat. Ezekben a digitális hálózatokon már lehetőség nyílik nagyobb sebességű jó minőségű adatátvitelre. Felmerült az igény mind a szolgáltatói mind a felhasználói oldalon egy egységesített távbeszélő hálózat és szolgáltatási rendszer kialakítására. Egy ilyen jövőbeli új hálózaton már lehetőség lenne egyidőben nagysebességű kép- és hangátvitelre, változó minőségű és sebességű adatátvitelre. Ez az új hálózat már kialakulóban van, elnevezése: ISDN (Integrated Services Digital Network - Integrált Szolgáltatású Digitális Hálózat).

Már a 19. sz. végén megjósolták, hogy a 20. században megvalósul a képtelefon. Mindannyiunk számára jó hír, hogy a régóta várt képátviteli szolgálatok az ezredforduló táján várhatóan egyre szélesebb körben megvalósulnak. Ez a megvalósítás azonban nem könnyű. A kivitelezés útján előreláthatólag többféle műszaki, társadalmi, kulturális problémával kell megküzdeni.

Digitális képfeldolgozás és képátvitel

A távközlési hálózatok fejlődése sok új igényes szolgáltatás számára teremti meg a lehetőséget. Ilyen szolgáltatások például: - kis- és nagy sebességű egy és kétirányú álló- és mozgókép-átvitelre épülő - videotelefon, videokonferencia, video-adatlekérdezés, HDTV-műsorszórás, és HDTV-kommunikáció. Ezeknél a szolgáltatásoknál mindig problémát jelentett az álló és mozgóképek átvitele, főként a nagy sáv szélességük és információ tartalmuk miatt.

Az alábbiakban egy rendszerező áttekintést adok a különböző képátviteli szolgáltatásokról, az adatsebesség, a kívánt információ intenzitás, és az adattranszfer iránya szerint [5].

- video adatlekérdezés (kisebbségű, egyirányú állókép transzfer)
- video telefónia (kis sáv szélességű, kétirányú kommunikáció)
- video konferencia (csoporton belüli kis sáv szélességű, kétirányú kommunikáció)
- video műsorszétosztás (nagy sáv szélességű, egyirányú mozgókép transzfer)
- video kommunikáció (nagy sáv szélességű, kétirányú kommunikáció)
- HDTV műsorszétosztás (igen nagy sáv szélességű, egyirányú képtranszfer)
- HDTV kommunikáció (igen nagy sáv szélességű, kétirányú kommunikáció)

A televíziós jelnek a digitális hálózaton történő átvitelét CCIR 601-es ajánlása rögzíti. Ennek alapján a hagyományosan analóg videojel digitalizálása, 13.5 MHz-es frekvenciával történő mintavételezéssel, 16 bit/pixel mélységű kvantálással oldható meg. Ily módon egy hagyományos televíziókép 720x576, míg egy HDTV kép 1440x1152 képpont (pixel) méretű adatmátrixként tárolható.

Egy rövid gondolat kísérlettel szeretnék rámutatni a képátvitel által igényelt nagy adatátviteli kapacitás problémájára. Egyetlen, a fent említett CCIR ajánlásnak megfelelő 1440x1152 felbontású 16 bit mélységű színes (HDTV) állókép tárolási igénye több mint 3.3 Mbyte, így az adattranszfer egy 9600 baudos átviteli csatornán háromnegyed óra lenne. Negyedórányi műsor információ mennyisége (50-es képváltás esetén) 150 Gbyte, ennek átvitele ugyanezen átviteli csatornán akár több évig is eltartana.

Azért hogy a videojel nagyfokú redundanciáját lecsökkentsék, többlepcsős elő- és utófeldolgozást alkalmaznak, így lehetővé válik a képátvitel, kisebb sebességű adatátviteli csatornán is. **Ugyanis a digitális képátvitel esetén** (szemben például az adattovábbításra szolgáló rendszerekkel) **nem a tökéletes, az eredetivel bitről bitre megegyező, hanem elsősorban azt az információ tartalmában hűen visszaadó adatprodukciónak a cél** [5].

A különféle képátviteli módozatokat és azok legfőbb jellemzőit röviden összefoglalja az 2.1. táblázat [5]. A táblázatból leolvasható, hogy a hagyományos televíziós műsor terítésére a 34 Mbit/s-os digitális rendszerek használhatók. Ez alatt a végfelhasználó felé történő (további feldolgozást nem igénylő, ún. előfizetői minőségű) jeltovábbítás értendő. A további feldolgozást igénylő (stúdió minőségű) esetben többnyire 140 Mbit/s-os átvitelt használnak.

2.1. táblázat Digitális képátvitelre leginkább használatos formátumok és paramétereik

Képforrás típusa	Felbontás Hor. x Vert. [pixel]	Kép-váltás [kép-s]	Kvantálási mélység [bit/pixel]	Forrás-sebesség [Mbit/s]	Csatorna-sebesség [Mbit/s]	Tömörítési faktor
HDTV	1140 x 1152	50	16	1327	140	~10
TV	720 x 576	25	16	166	34	~5
Videoconf.	360 x 288	25	16	42	0.384	~110
Videophone	180 x 144	6	12	2.5	0.064	~40

A nem stúdió minőségű jelátvitelnél, a veszteséges tömörítő eljárások alkalmazásával jóval kisebb sávszélesség is elegendő. Ezeknél az eljárásoknál, mivel nem veszteségmentesek, bizonyos fokú információvesztés történik, de ez általában emberi szemmel nem, vagy csak alig érzékelhető. Ha a jel a visszaalakítási oldalon további feldolgozáson megy keresztül, az információ megváltozásával fellépő hibák halmozódnak és már akár észlelhető formában is megjelenhetnek. Ezért a veszteségmentes eljárásokat csak olyan helyeken szabad alkalmazni, ahol a tömörített jelet nem vetik alá még különböző elő- és utófeldolgozásnak. A fenti táblázatban szereplő tömörítési faktor értékek csak jól megválasztott veszteségmentes tömörítő eljárások alkalmazásával érhetők el.

A fent említettek miatt egyre több helyen jelentős kutatások folynak a digitalizált videojel tömörítésének optimalizálására. A jó hatásfokú képtömörítési eljárások szinte minden esetben többlépcsős feldolgozáson mennek keresztül. Ezek többnyire valamilyen (matematikai) transzformációs feldolgozás és veszteségmentes kódolás kombinációjából állnak: [5]

- a sík tartományból frekvenciára történő transzformáció;
- kvantálás;
- entrópia (vagy Huffman) kódolás;
- mozgásbecslés és kompenzáció;

Tér-frekvencia transzformációk

Egy kép valójában különböző színű és intenzitású képpontok x számú oszlopba és y számú sorba rendezett halmaza. Ezt a képet a digitalizálás után egy ugyanilyen (x oszlopból és y sorból álló) mátrix ír le, ahol az egyes helyekre az oda tartozó képpontok luminancia vagy krominancia értéke kerül. Ez a kétdimenziós matematikai mátrix a kép térbeli intenzitás-eloszlásáról tartalmaz információt. Tehát egy ilyen képmátrix felfogható egy egydimenziós, egyváltozós, (folytonos függvényből mintavételezett) diszkrét függvénynek. Ezen a függvényen végrehajtva valamilyen frekvencia transzformációt (Fourier, koszinusz, szinusz) megkapjuk, hogy milyen frekvenciájú összetevők milyen amplitúdóval fordulnak elő benne. Az emberi szem korlátozott frekvenciaérzékenysége miatt bizonyos kevésbé érzékelhető magasabb-frekvenciás összetevők elnyomhatók.

A transzformációk közül leginkább a **DCT** („digital Cosine Transform”) diszkrét koszinusz transzformációt szokták alkalmazni. Ez talán az egyetlen, **amelyik nem igényel túl nagy bonyolultságú számításokat és emellett jól megközelíti az elméleti optimumot**. A számítási igény a transzformálandó mátrix méreteinek növekedésével rohamosan nő. Ezért a legtöbb esetben nem a teljes adathalmazra, hanem annak kisebb blokkjaira érdemes elvégezni az átalakítást. Itt valamilyen kompromisszumot kellett keresni az optimális blokkméret megtalálása

érdekében. Ennek eredményeképpen 8x8-as, esetenként 16x16-os méretű blokkokra szokás osztani a képmátrixot.

Ez általában már elegendő redundancia-csökkentésre ad lehetőséget, gyors kalkulációs idő mellett. (Ez utóbbi hardware megvalósításnál gyakorlatilag „real time” feldolgozást kell, hogy jelentsen!).

Kvantálás

Ez a lépés csökkenti a képet leíró numerikus adatok számát és azok előfordulási tartományát, azon az áron, hogy a kép egyes, közeli elemeit megkülönböztető finomabb részletet elveszítünk.[5] A kvantálás során tulajdonképpen az emberi szem számára kevésbé fontos magas-frekvenciás összetevőket nyomjuk el (vagy kiszűrjük) egy alkalmasan megválasztott kvantáló mátrix segítségével.

Egy kép továbbításánál alkalmazott hatásos tömörítések kidolgozásánál legtöbbször az emberi szem fizikai tulajdonságait veszik figyelembe. Egyik az, hogy **a szemünk a fény intenzitására jóval érzékenyebb, mint annak színösszetételére**, a másik pedig, hogy **az emberi szemnek korlátozott a felbontóképessége**. Az előzőek azt jelentik, hogy az eredeti, (akár színes akár monokromatikus) képhez képest egy tömörített kép esetében sokkal zavaróbb az, ha egyes finomabb részleteket elválasztó kontrasztok eltűnnek, mint az, ha két egymás mellett lévő hasonló színű képpont színe a tömörítés utáni visszaalakításnál azonos lesz. Teljesen hasonlóan az emberi látás jellegéből adódóan egy kép igazán finom részleteit illetően sem vagyunk igazán kényesek (ez különösen igaz mozgókép esetén), ezért az azokat reprezentáló magasabb frekvenciás összetevőket elhanyagoljuk. (Természetesen egy állókép esetében a részletgazdagság nagyobb súllyal esik latba.)

Video adathalmaz tömörítése során ilyen (redundanciacsökkentő) jellegű kvantálást a frekvenciatranszformáció (DCT) után szokás használni. **A kvantálás mértékét az adott alkalmazásnak megfelelően szokták megállapítani**. Fontos hogy ezt a megfelelő körültekintéssel, helyesen válasszuk meg, mert itt a visszaalakításnál információt veszítünk. Mivel - ahogy azt már az előbb említettem -, az emberi látásmechanizmus adottságai folytán a szemünk jóval érzékenyebb a fényesség intenzitására és elnézőbb a színesség kisebb hibáit illetően, a krominancia értékek kvantálása nagyobb mértékű lehet mint a luminancia adatoké. Ezért jóval nagyobb fokú tömörítés érhető el a színtonkomponenseket leíró adathalmazon, mint a fényességet leíró adatokon, emberi szemmel észrevehetetlen vagy alig észrevehető minőségromlás nélkül.

Egyébként a manapság alkalmazott video leképező rendszerek (videokamerák, filmfelvevő berendezések) felépítésüknél fogva sem képesek a nagyfrekvenciás komponensek tökéletes átalakítására. Például egy kamera optikai leképező rendszere az átalakítás és az átvitel szempontjából egy kétdimenziós aluláteresztő szűrőként viselkedik, amihez még hozzá jön a működési elvből fakadó és a hozzá kapcsolódó berendezések átviteli torzítása.

Forrás kódolás

Ennél a lépésnél az eddigi műveletek eredményeként előálló adathalmazon, egy újabb kódolási fázist szokás beiktatni, aminek eredményeképp az adatok mennyisége információ-vesztés nélkül csökkenthető. **A kvantálás eredményeképp nem csak a transzformált adatok mennyisége, hanem a tartomány is jelentősen csökken**, amelyből ezek értékeket vehetnek fel. **E kettő következményeként a kvantálást követően kialakuló adatmátrixban általában megnő az azonos elemek előfordulási gyakorisága**. Ez leginkább a nulla értékekre lesz igaz,

hiszen a kis értékű frekvencia komponensek a kvantálás során végrehajtott osztás és kerekítést követően „kinullázódnak”.

Így néhány elemérték sokkal gyakrabban fordul elő, mint mások, és alighanem akad olyan is, amely viszont feltűnően ritka az adott adathalmazon belül, ezért további feldolgozást lehet végezni az adattömörítés érdekében. **Ezt a tulajdonságot ugyanis kihasználhatjuk az entrópia vagy (Huffman) kódolásra.** Ez abból áll, hogy a többinél nagyobb gyakorisággal előforduló adatokhoz relatíve rövid kódszót rendelünk, például ha mondjuk a „nulla” érték a leggyakoribb, akkor nem 8, hanem 1 bites nullákat rendelünk hozzá. Hasonlóan járunk el a többi, előfordulási valószínűség szerint sorba rakott értékkel is, tehát a leggyakoribbhoz a legrövidebb, a legkevésbé valószínűhöz pedig viszonylag hosszú kódszót rendelünk. Természetesen nagyon kis valószínűség esetén előfordulhat hogy azt az eredetinél hosszabb kódszóra kell kicserélnünk, azonban az egész adathalmazra vett adatredukció még így is jelentős lehet.

Az eljárás hatékonysága még tovább javítható, amennyiben nemcsak egyes elemek hanem egymás környezetében lévő elem-csoportok előfordulási valószínűségei alapján rangsorolunk. Mivel az eredeti kép kvantált frekvenciamátrixára jellemző, hogy viszonylag sok zérus értéket tartalmaz, az entrópia kódolási eljárás során a különböző „futam hosszúságú” zérus-kombinációkhoz rövid kódszavak rendelése révén hatékony adatredukciót érhetünk el.

Mozgás-kompenzáció

A mozgás-kompenzáció a mozgóképek időbeli korrelációját használja ki a nagyobb adattömörítés megvalósítására. Ez az eljárás tulajdonképpen egy képszekvenciában az egymást követő képek közötti eltérések számítása révén teszi lehetővé, hogy a soron következő kép leírható legyen az előző kép(ek)re vonatkozó (tehát már ismert) információk felhasználásával.

A mozgáskompenzációs kódolások a legtöbb esetben úgy történnek, hogy definiálnak valamilyen alap-, vagy bázisképet. Először ezt a bázisképet viszik át és utána már a következő képeket a bázisképtől való eltérés alapján generálják. Ez olyan alkalmazásoknál különösen hatékony, ahol az egymást követő képek között csak kis változások vannak. Egy ilyen tipikus alkalmazás a képtelefon, ahol általában a telefonálók megjelenő arca a báziskép és az átviteli rendszeren csak az arc különböző változásait, mimikát, a szájmozgásokat kell átvinni. Néha természetesen szükségessé válhat újabb báziskép generálása és átvitele, ami ilyenkor dinamikusan terheli az átviteli csatornát. Ha az átviteli rendszer nincs felkészülve az ilyen dinamikus információlökésekre, akkor a mozgókép lelassul, szaggatottá válik, és ez zavaró.

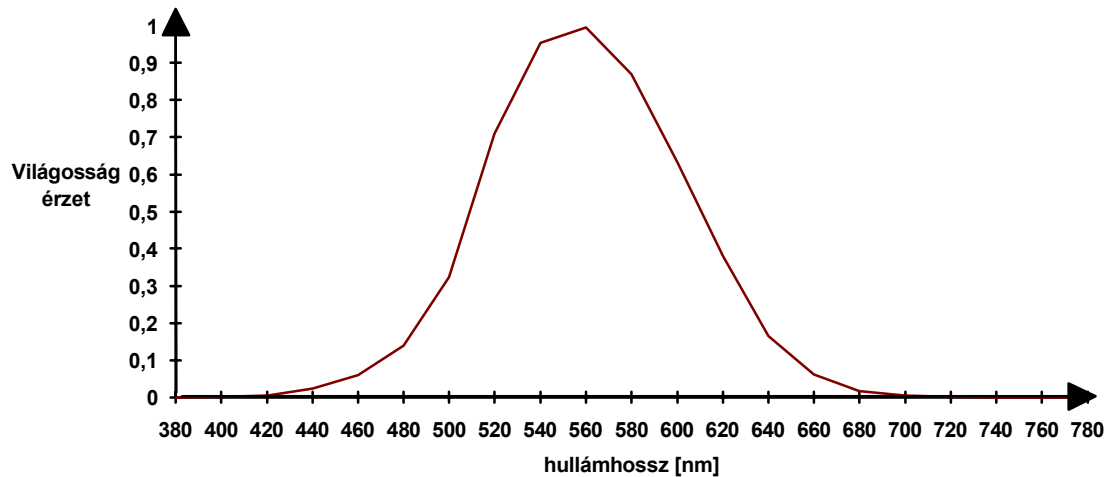
Matematikai transzformációk, mozgáskompenzációs, predikciós, iterációs eljárások széles skálájáról választhatunk a különféle képtömörítési célok elérése érdekében. Természetesen ezek kombinációjából a felhasználás jellege és a rendelkezésre álló lehetőségek szerint számos megvalósítás állítható össze.

Látási és képtechnikai alapismeretek

Alapvetően két csoportra bonthatók a képforrások. Általánosan elterjedt és széleskörben használt az ember számára látható tartomány leképezése, és speciális területeken használt a nem látható tartomány leképezése (ilyen pl. az infravörös tartomány, radar, stb.). A következőkben az első esettel foglalkozunk, de a megállapítások alkalmazhatók a második területre is.

Képfelvételen azt az eljárást értjük, amelynek során a valóságos világot egy képsíkra képezzük le az emberi látás fiziológiai tulajdonságait figyelembevéve. A képsík (x,y) helykoordinátái, jellemezhetőek egy $f(x,y)$ függvénnyel. Mozgó képek esetén a képfüggvény még az időnek is függvénye, tehát a képfüggvény: $f(x,y,t)$.

Az emberi látás spektrális érzékenységet a hullámhossz függvényében a láthatósági függvény, a $V(\lambda)$ adja meg (3.1. ábra) [9].



3.1. ábra Láthatósági függvény

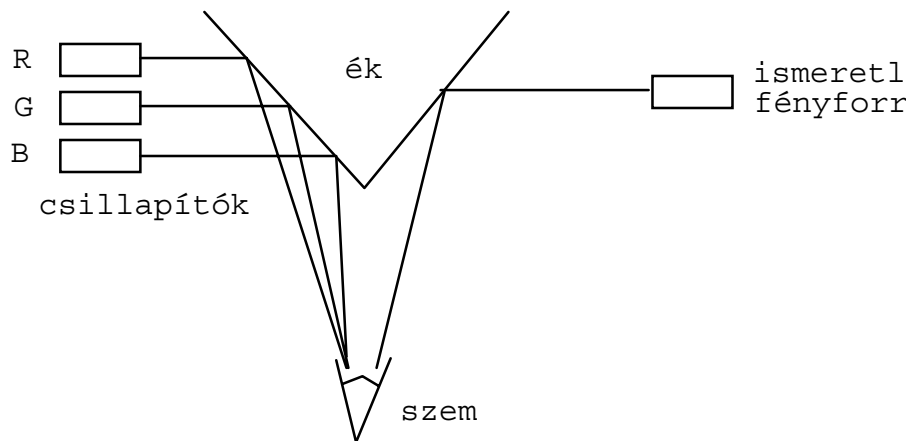
A tapasztalatok azt mutatták, hogy tetszőleges színérzet előállítható három, megfelelően megválasztott alapszín segítségével. Ezek a CIE szabvány szerint:

$$\lambda_R = 700,0nm$$

$$\lambda_G = 564,1nm$$

$$\lambda_B = 435,8nm$$

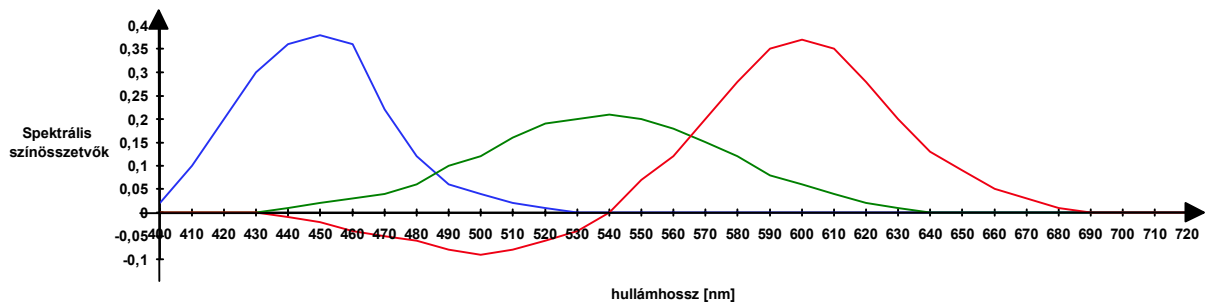
Az összehasonlító színmérés elve:



A képen látható megfigyelőnek a csillapítókat úgy kell beállítania, hogy a két oldalról érkező fény egyforma érzetet keltsen. Ennek a módszernek az alkalmazásával ki lehet mérni a hullámhossz függvényében, hogy a három alapkompontent milyen arányban kell venni a kívánt spektrálszín előállításához.

Az alábbi (3.2. ábra) azt mutatja, hogy a nemzetközileg elfogadott három alapszín segítségével bármely más hullámhosszúságú fényvel azonos színérzetet elő lehet állítani. Ezek alapján a különböző hullámhosszúságú egységnyi teljesítményű monokromatikus $S(\lambda)$ fény kiegyenlítésével megkapjuk a spektrális alapszín-összetevőket:

$$S(\lambda) = r(\lambda)*[R]+g(\lambda)*[G]+b(\lambda)*[B]$$



3.2. ábra

A függvény megadja, hogy az [R], [G], [B] értékeknek milyen értékűeknek kell lenni, vagyis hogyan kell súlyozni az alapszíneket az adott hullámhosszúságú spektrálszín kikeveréséhez.

A kék görbe neve $b(\lambda)$, a vöröské $r(\lambda)$ a zölké $g(\lambda)$

Általános esetben egy tetszőleges X keverékszínre (ahol a X tetszőleges teljesítmény-eloszlású mérendő fény) a kiegyenlítés feltétele:

$$X = R*[R]+G*[G]+B*[B]$$

Az R, G, B, összetevők értéke folytonos spektrumú összetevők esetén:

$$R(x, y, t) = \int_0^{\infty} C(x, y, t, \lambda) \cdot r(\lambda) d\lambda$$

$$G(x, y, t) = \int_0^{\infty} C(x, y, t, \lambda) \cdot g(\lambda) d\lambda$$

$$B(x, y, t) = \int_0^{\infty} C(x, y, t, \lambda) \cdot b(\lambda) d\lambda$$

Fekete-fehér kép esetén a képfüggvény:

$$f(x, y, t) = Y(x, y, t) = \int_0^{\infty} C(x, y, t, \lambda) \cdot V(\lambda) d\lambda$$

ahol $C(x, y, t, \lambda)$ $C(x, y, t, \lambda)$ (színingerfüggvény) egy az x, y, t, λ , értékektől függő szorzó, a $V(\lambda)$ a hullámhossztól függő intenzitás.

Színes képek esetén:

Az Y és az R, G, B jelek közötti kapcsolatot az alábbi összefüggés adja meg:

$$Y = 0,3R + 0,59G + 0,11B$$

Ezt az egyenletet átrendezve azt kapjuk, hogy:

$$0,3(R - Y) + 0,59(G - Y) + 0,11(B - Y) = 0$$

Az R-Y, G-Y, B-Y jeleknek nincs fénysűrűség-tartalma, csak színesség-információt hordoznak. Ezeket a jeleket színkülönbségi jeleknek hívjuk. Látható, hogy kettő ismeretében a harmadik mindig kiszámítható. A televíziótechnikában a színes kép átvitelére az Y, az R-Y és a B-Y jeleket használják.

Különböző képtömörítési eljárások rövid áttekintése

Két nagy csoport különböztethető meg a képkódolás technikájában a visszaállítási tulajdonságok szerint [2]:

- veszteségmentes tömörítési eljárások, ilyen például a futamhosszkódolás, Huffman-kódolás, aritmetikai kódolás, stb., tehát entrópia csökkentő eljárások;
- veszteséges tömörítési eljárások, ilyen például a transzformációs kódolás, részsávkódolás, vektorkvantálás, stb., ahol a kódolás során információveszteség lép fel.

Az első csoportba tartozó eljárások alkalmazása során az eredeti kép tökéletesen visszaállítható, hátránya a második csoportba tartozó eljárásokkal szemben: a kisebb tömöríthetőség. A második csoportba tartozó eljárások összehasonlítása jóval nehezebb, gyakorlatilag csak szubjektív minősítési kritériumok alapján történhet. A gyakorlati követelményeket elsősorban a veszteséges eljárásokkal elégítik ki, csak néhány speciális terület igényli a veszteségmentes tömörítési eljárásokat, ilyen lehet például néhány orvosi alkalmazás, vagy az úrkutatás során készült légifelvétel. **A veszteséges kódolási alkalmazások során az alkalmazott adat-tömörítés mértéke és a keletkező kép minősége között kell a kompromisszumot megtalálni.**

Hatékony kódolás a következőképpen valósítható meg:



A reprezentáció váltás és a kvantálás együtt alkotja a veszteséges tömörítési eljárást, míg a veszteségmentes eljárással további tömörítés érhető el. **A reprezentáció váltás célja: hogy a jelet olyan tartományba képezze le, ahol bizonyos tulajdonságai folytán veszteséges tömörítés végezhető el rajta.** A reprezentáció váltáshoz tartoznak például a következő eljárások:

- részsáv kódolás,
- transzformációs kódolás,
- mozgáskompenzált kódolás.

A veszteséget a rendszerbe a kvantálás viszi be, ennek két nagy csoportja különböztethető meg:

- skalár kvantálás: minden egyes mintát külön kvantálnak,
- vektor kvantálás: minták egy csoportjának tömbbe, azaz vektorba rendezése.

Az emberi szem látási tulajdonságai objektív paraméterekkel igen nehezen mérhetők, ugyanis előfordulhat az az eset, hogy míg egy kódolási eljárás egy objektív paraméter, például jel/zaj viszony szempontjából jobb, mint egy másik, addig szubjektív megítélés alapján rosszabb is lehet. Azért érdemes definiálni néhány objektív paramétert:

$$\text{jel/zaj viszony: } \text{SNR} = 10 \cdot \lg \frac{\frac{1}{XY} \sum_{i=1}^X \sum_{j=1}^Y x_{ij}^2}{MSE}$$

$$\text{csúcs jel/zaj viszony: } \text{PSNR} = 10 \cdot \lg \frac{255^2}{MSE}$$

ahol $MSE = \frac{1}{XY} \sum_{i=1}^X \sum_{j=1}^Y (x_{ij} - y_{ij})^2$ a négyzetes hiba, X és Y a kép vízszintes és függőleges mérete, $x_{i,j}$ az eredeti, $y_{i,j}$ a rekonstruált pixel.

Érdeemes még bevezetni egy paramétert, amely a tömörítés hatékonyságát hivatott kifejezni, ez pedig az 1 pixelre jutó bitek száma:

$$\text{BR} = \frac{\text{a képet leíró bitek száma}}{\text{a képet kifejező összes pixel}}$$

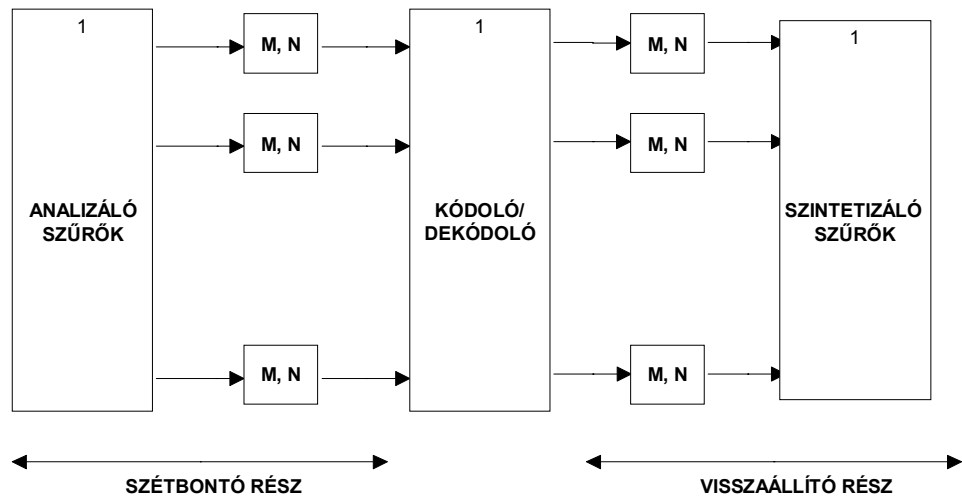
A következőkben a fent említett eljárásokról szólok néhány szót, különös tekintettel a DCT transzformációs eljárásra.

Részsávkódolás

A részsávkódolást Crochiere és társai fejlesztették ki elsősorban hangfrekvenciás jelek átvitelére, de sikeresen alkalmazható képek kódolása esetén is. **Az alap gondolat az, hogy a képek kétdimenziós frekvenciasávját meghatározott számú részsávra osztják, a részsávokat külön-külön kódolják.** A részsávok kialakításánál figyelembe veszik az emberi szem látási tulajdonságait. Ennek előnye, hogy különböző fontosságú sávok hozhatók létre. A jelenergia a kisfrekvenciás tartományban koncentrálódik, és az emberi szem erre a tartományra a legérzékenyebb. Azokban a részsávokban, ahol az emberi szem érzékenysége kisebb, nagyobb tömörítés érhető el. Előnye még ennek az eljárásnak, hogy segítségével többszörös felbontású képkezelés valósítható meg, így például alacsonyabb szinten egy alapszolgáltatás nyújtható, felsőbb szinten pedig egy nagy felbontású, jobb minőségű szolgáltatás.

Egy részsávkódoló három jól elkülöníthető részből áll:

- Szétbontó rész. Itt egy analízáló szűrősor a bejövő jelet részsávokra választja és mindegyik részsávot alulmintavételezi.
- Kódoló/dekódoló, amely mindegyik részsávot külön-külön kódolja valamilyen ismert kódolási technikával.
- Visszaállító rész, ahol egy szintetizáló szűrősor az egyes részsávok mintavételezés utáni visszaállítását és összegzését végzi.



4.1. ábra A részsávkódoló felépítése

Az analízáló és szintetizáló folyamat során az egyes részsávok átlapolódhatnak, és ez torzítást okozhat. Torzításmentes visszaállítást megfelelően választott és méretezett szűrők biztosítanak.

Transzformációs kódolások

A transzformációs kódolások jelentős szerepet töltenek be különböző digitális jelfeldolgozások során. Alkalmazásuk oka: olyan területre transzformálható át az eredeti jel, ahol a jel bizonyos jellemző tulajdonságai élesen megmutatkoznak. *A kódolás során a bejövő jelet blokkokra osztjuk, mindegyik blokkon elvégezzük a transzformációt.* Ezek után kerül sor a kvantálásra. Ez az a pont, ahol a veszteség bejön a rendszerbe, mert a transzformáció önmagában veszteségmentes művelet. A kvantálás során veszik figyelembe az emberi szem látási tulajdonságait, ugyanis a transzformált blokk együtthatói más-más térbeli frekvenciákhoz tartoznak. Felállítható az együtthatók között bizonyos fontossági sorrend a látási tulajdonságok alapján.

A transzformációs kódolások egyik csoportja az ortogonális jeltranszformációk. Ebben az esetben a bejövő jelet az alapfüggvények súlyozott összegével helyettesítjük:

$$x(t) = \sum_{n=0}^{\infty} a_n \cdot u_n(t) \quad (5.1)$$

ahol a_n az együttható, u_n pedig a bázisfüggvény.

Az ortogonális transzformációkat még két csoportra oszthatjuk:

harmonikus bázisfüggvényű	nem harmonikus bázisfüggvényű
DFT	Walsh-Hadamard
FFT	Haar
DCT	Slant
DST	Wavelet

A harmonikus bázisfüggvényűek közül a DFT és az FFT komplex számítást igényel, míg a diszkrét koszinusz- illetve szinusztranszformáció csak tisztán valós.

Az alábbiakban én a gyakorlati fontosságuk miatt a DFT, FFT és bővebben a DCT eljárással fogok foglalkozni.

Matematikai alapok [1]

Fourier sor

Ismert összefüggés, hogy egy $f(t)$ periodikus időfüggvény felbontható egy konstans tag, továbbá végtelen sok harmonikus összetevő összegére. A harmonikus összetevőket általában szinuszos és koszinuszos időfüggvényekkel írják le. A periodikus jel T periódusidejének reciproka adja az alapharmonikus frekvenciáját, és a további összetevők frekvenciája ennek az alapfrekvenciának az egész számú többszöröse. Matematikai formában:

$$f(t) = c_0 + \sum_{n=1}^{\infty} (a_n \cos(\omega n t) + b_n \sin(\omega n t)) \quad \text{ahol } \omega = 2\pi \frac{t}{T} \quad (6.1)$$

A c_0 , stacionárius egyenáramú összetevő: $c_0 = \frac{1}{T} \int_0^T f(t) dt$. (6.2)

A harmonikus összetevők amplitúdója :

$$a_n = \frac{2}{T} \int_0^T f(t) \cos(\omega n t) dt \quad (6.3)$$

$$b_n = \frac{2}{T} \int_0^T f(t) \sin(\omega n t) dt \quad (6.4)$$

Ismeretes a sor komplex amplitúdóval felírt alakja is:

$$f(t) = \sum_{n=-\infty}^{\infty} C_n e^{jn\omega t} \quad (6.5)$$

$$C_n = \int_0^T f(t) e^{-jn\omega t} dt \quad (6.6)$$

ahol $C_n = \frac{a_n - jb_n}{2}$ a komplex amplitúdó, amire érvényes a következő összefüggés: $C_{-n} = C_n^*$ és $C_0 = c_0$

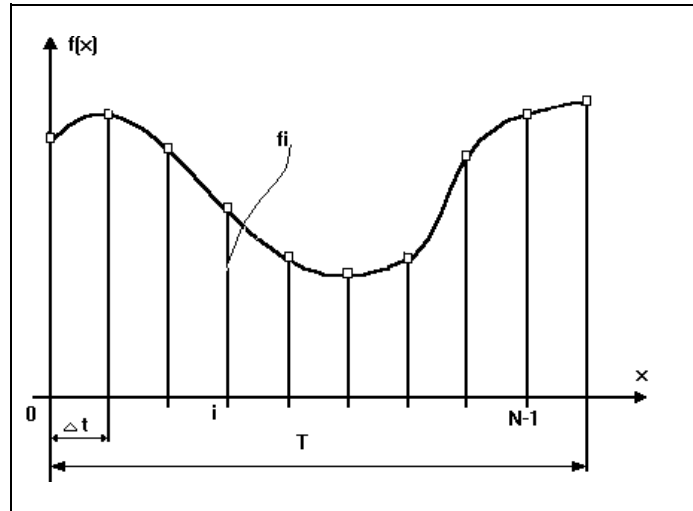
Némi levezetéssel belátható (amit itt most nem teszek meg), hogy fenti összefüggések akkor is alkalmazhatók ha az $f(t)$ függvény komplex értékű.

Diszkrét Fourier transzformáció (DFT)

A mérnöki gyakorlatban és a természettudományokban sokkal gyakoribb az az eset, amikor a folytonos időtartományú jeleket a mintavételezett értékeiből előállított diszkrét idejű jelekkel helyettesítjük.

Legyen a helyzet például az (6.1 ábra) szerinti. A vizsgált függvény továbbra is T szerint periodikus, de értékét csak véges számú x helyen ismerjük, tehát csak véges számú minta áll rendelkezésünkre. Legyen ezek jele f_i . A T hosszúságú periódusra essék N darab minta.

Ekkor a minták távolsága: $\Delta x = \frac{T}{N}$.



6.1. ábra

Ha a mintákat a mintavételi tétel szerint elegendően sűrűn vesszük (az „elegendően sűrűn” azt jelenti hogy mintavétel olyan gyakori kell hogy legyen, hogy a jelben előforduló legmagasabb frekvenciájú összetevő egy periódusára legalább két minta essék, Shannon mintavételi törvénye alapján), akkor az eredeti jelet információvesztés nélkül vissza tudjuk állítani, tehát tökéletesen reprezentálja azt.

Próbáljuk meg közelítőleg, a függvény alatti téglalapok segítségével meghatározni a Fourier sor elemeit. Az integrált, szummával közelítve:

$$C_n = \frac{1}{T} \sum_{i=0}^{N-1} f_i \Delta x e^{-j2\pi \frac{1}{T} i \Delta x} \quad (6.7)$$

Helyettesítsük be a $\Delta x = \frac{T}{N}$ összefüggést :

$$D_n = \frac{1}{T} \sum_{i=0}^{N-1} \frac{T}{N} f_i \Delta x e^{-j2\pi \frac{1}{T} i \frac{T}{N}} = \frac{1}{N} \sum_{i=0}^{N-1} f_i e^{-j \frac{2\pi}{N} i n}$$

Az így kapott összefüggés a DFT formulája.

Bizonyítható hogy ennek a transzformálnak csak N darab egymástól különböző értéke lehet, mert,

$$D_{n+N} = \frac{1}{N} \sum_{i=0}^{N-1} f_i e^{-j \frac{2\pi}{N} i (n+N)} = \frac{1}{N} \sum_{i=0}^{N-1} f_i e^{-j \frac{2\pi}{N} i n} e^{-j2\pi i} = D_n$$

(mivel i csak egész lehet, és az $e^{-j2\pi i}$ mindig 1).

Most hogy már sikeresen elő tudjuk állítani a D_n transzformált értékeket, próbáljuk meg ugyanezt hasonló gondolatmenettel visszafelé, vagyis visszaállítani az eredeti $f(i)$ értékeket úgy, hogy csak N darab ($0 \leq n < N$) D_n transzformált értéket használunk fel.

$$f_k = \sum_{n=0}^{N-1} D_n e^{j\frac{2\pi}{N}kn} \quad (6.8)$$

A művelet helyességéről könnyen meg lehet győződni. Ha visszahelyettesítenénk a képletbe $D_n = \frac{1}{N} \sum_{i=0}^{N-1} f_i e^{-j\frac{2\pi}{N}in}$ -et, akkor azonosságot kapnánk. Tehát ez valóban a visszatranszformálás összefüggése. (A levezetés a szakirodalomban [1] részletesen megtalálható.)

Összefoglalva : az N elemből álló $f(i)$ értéksorozat diszkrét Fourier transzformáltja: $f = \frac{1}{2\Delta t}$

$$D_n = \frac{1}{N} \sum_{i=0}^{N-1} f_i e^{-j\frac{2\pi}{N}in} \quad 0 \leq n \leq N-1 \quad (6.9)$$

A visszatranszformálást az alábbi szummával végezzük:

$$f_k = \sum_{n=0}^{N-1} D_n e^{j\frac{2\pi}{N}kn} \quad 0 \leq k \leq N-1 \quad (6.10)$$

Mind a transzformált mind a visszatranszformált függvény sor N -nel periodikus:

$$D_{n+N} = D_n, \quad f_{k+N} = f_k \quad (6.11)$$

Ha az $f(i)$ értéksorozat valós elemekből áll, a D_n transzformált egyes elemei között bizonyos szimmetria összefüggések állnak fenn:

D_0 valós,

$D_{-n} = D_n^*$, ami így is írható: $D_{N-n} = D_n^*$

$\frac{D_N}{2}$ valós.

Az N valós számból álló értéksorozat diszkrét Fourier transzformáltját tehát egyértelműen megadhatjuk a D_n transzformált értéksor felével. Pontosabban az alábbiakkal:

$$D_0, D_1, D_2, \dots, D_{\frac{N}{2}-1}, \frac{D_N}{2}$$

A DFT kapcsolata a vizsgált függvény harmonikus összetevőivel.

Tegyük fel, hogy valós értékű f_x jelből Δx lépésközzel N darab mintát veszünk, és ezekből az f_i mintákból előállítjuk a diszkrét Fourier transzformáltat, vagyis a D_n értéksorozatot. Kérdés az, hogy a D_n sorozatból vissza tudjuk-e állítani az eredeti mintákat, és azokból az eredeti függvényt. Az előző pont összefüggéseit figyelembe véve alakítsuk át az inverz transzformáció szummáját:

$$f_k = D_0 + \sum_{n=1}^{\frac{N}{2}-1} D_n e^{j\frac{2\pi}{N}kn} + \frac{D_N}{2} e^{j\pi k} + \sum_{n=\frac{N}{2}+1}^{N-1} D_n e^{j\frac{2\pi}{N}kn}.$$

Ebből rendezés után:

$$f_k = D_0 + D_{\frac{N}{2}} e^{j\pi k} + \sum_{n=1}^{\frac{N}{2}-1} \left(D_n e^{j\frac{2\pi}{N}kn} + D_{-n} e^{-j\frac{2\pi}{N}kn} \right) =$$

$$= D_0 + D_{\frac{N}{2}} e^{j\pi k} + \sum_{n=1}^{\frac{N}{2}-1} \left(D_n e^{j\frac{2\pi}{N}kn} + D_n^* e^{-j\frac{2\pi}{N}kn} \right)$$

Végül, felhasználva a trigonometrikus függvények exponenciális kifejezéseit, hogy

$$\cos(x) = \frac{1}{2} (e^{jx} + e^{-jx}) \quad \text{és} \quad \sin(x) = \frac{1}{2j} (e^{jx} - e^{-jx})$$

$$f_k = D_0 + D_{\frac{N}{2}} \cos(\pi k) + \sum_{n=1}^{\frac{N}{2}-1} 2 \left(\operatorname{Re}(D_n) \cos\left(\frac{2\pi}{N}kn\right) - \operatorname{Im}(D_n) \sin\left(\frac{2\pi}{N}kn\right) \right) \quad (6.12)$$

Mivel egyenértékű átalakításokkal jutottunk eddig, ez a kifejezés k index egész értékeinél ugyanúgy szolgáltatja az f_k értékeket, mint az inverz transzformáció egyenlete. Vajon ezekből az f_k értékekből következik-e az az f_x függvény, amiből a mintákat vettük? Könnyen megvizsgálhatjuk ezt, ha teszünk egy formális lépést. A k érték legyen index helyett egy folytonos független változó úgy, hogy a mintavételi értékek helyein értéke éppen egész szám legyen:

$$k = \frac{x}{\Delta x}, \quad x = k\Delta x$$

Ezzel :

$$f_{(x)} = D_0 + D_{\frac{N}{2}} \cos\left(\frac{N}{2} \frac{2\pi}{N\Delta x} x\right) +$$

$$+ \sum_{n=1}^{\frac{N}{2}-1} 2 \left(\operatorname{Re}(D_n) \cos\left(n \frac{2\pi}{N\Delta x} x\right) - \operatorname{Im}(D_n) \sin\left(n \frac{2\pi}{N\Delta x} x\right) \right)$$

Ez egy olyan függvény, ami az $x = 0, \Delta x, 2\Delta x$, stb. helyeken az eredeti f_k mintasorozat értékeit adja. Tehát egy olyan függvényre jutottunk, ami a mintavételi pontokban egybeesik az eredetivel. De az még nem biztos, hogy az eredeti f_x függvényt kaptuk vissza. A DFT egyenletében ugyanis az exponenciális kifejezések kitevőjében n helyén bármikor állhat $n+N$, $n+2N$ stb., anélkül hogy a kifejezés értéke megváltozna. Akkor pedig a levezetett egyenletünkben sem biztos, hogy

$$\cos\left(n \frac{2\pi}{N\Delta x} x\right), \quad (6.13)$$

áll. Ha ennek helyén például

$$\cos\left((n+N) \frac{2\pi}{N\Delta x} x\right) \quad (6.14)$$

szerepel, a kapott függvény a mintavételi pontokban továbbra is egybe esik az eredetivel, de mégsem lesz azonos az $f_{(x)}$ -szel. Vagyis számtalan olyan függvényt konstruálhatunk, amelyek a mintavételi pontokban azonos értékűek az eredeti $f_{(x)}$ -szel, de máshol különböznek tőle. Egy egyszerű kikötéssel mégis áthidalhatjuk ezt a problémát. Feltételezzük hogy a mintavétel előtt az $f(x)$ jelet szűrtük, úgy, hogy a

$$\cos\left(2\pi\frac{1}{2\Delta x}x\right)$$

összetevőnél magasabb felharmonikus biztosan nem fordul elő benne. Ez azt jelenti, hogy benne, $2\Delta x$ -nál rövidebb periódushosszú összetevő nem fordul elő. Ez a szűrés kűszöböli ki az aliasing (spektrum-átlapolódás) jelenség kellemetlen hatásait, ezért antialiasing szűrésnek nevezik. Ebben az esetben az (6.14) komponensek nem fordulhatnak elő.

Összefoglalva:

Vizsgáljuk az $f(t)$ valós értékű függvényt,

kikötjük hogy az eredeti függvény előzetes aluláteresztő szűrésen esett át és ezért $f = \frac{1}{2\Delta t}$ -nél magasabb frekvenciájú összetevő biztosan nem fordul elő benne, a függvény T hosszúságú periódusán Δt lépésközzel N darab mintát veszünk, a mintákból előállítjuk a $D_0, -D_{\frac{N}{2}}$ elemeket.

Ebben az esetben az $f(x)$ függvény vizsgált szakaszát egyértelműen visszaadja az alábbi kifejezés:

$$f_{(t)} = D_0 + \sum_{n=1}^{\frac{N}{2}} 2\left(\operatorname{Re}(D_n)\cos\left(\frac{2\pi}{T}tn\right) - \operatorname{Im}(D_n)\sin\left(\frac{2\pi}{T}tn\right)\right)$$

A fenti feltételek mellett a DFT elemei egy véges felharmonikus számon végzett Fourier sorfejtés együtthatóival azonosíthatók úgy, hogy a

D_0 adja a stacionárius összetevőt,

D_n adja az n -edik felharmonikus amplitúdóját.

Transzformációs kódolások

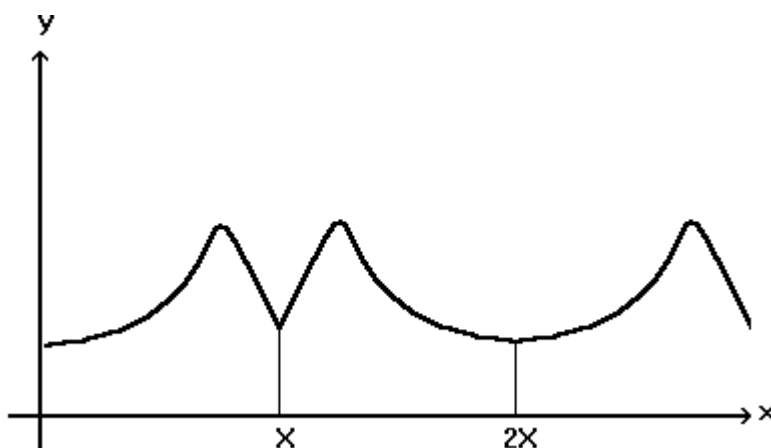
A Karhunen-Loeve transzformáció

A Karhunen-Loeve transzformáció az optimális transzformáció, ugyanolyan PCM minőség mellett a legkisebb bitsebesség érhető el vele. A bemenő jelet dekorrelálja, úgy, hogy a bejövő jelből meghatározza az $N \times N$ -es korrelációmátrixát, ennek meghatározza a sajátvektorait, s ezek lesznek az optimális transzformációs mátrix sorai. Ez az eljárás nagyon számításigényes, a gyakorlatban nem használják, de mint optimális transzformáció, referenciául szolgál a többi transzformációs kódolás jóságának megállapításához.

Diszkrét koszinusz transzformáció

Mind a képfeldolgozás gyakorlatában, mind a térszámítás egyes feladatainál szívesen használják a diszkrét Fourier transzformáció egyik ágát: a diszkrét koszinusz transzformációt. A diszkrét koszinusz transzformációnak amellett, hogy a legjobban közelíti az optimális KLT transzformációt, gyors implementációs lehetőségei vannak.

Az eljárást a folytonos függvények Fourier sorba fejtésére vonatkozó összefüggések segítségével vezetem be. Az $f(x)$ függvényt akarjuk sorral közelíteni, a 0 és x közötti tartományon. Mivel e tartományon kívül nem cél, hogy a közelítés jó legyen (lehet, hogy a függvény további szakaszát nem is ismerjük), tulajdonképpen tetszőleges feltételezést tehetünk a függvény további lefutására vonatkozóan. Legyen olyan a függvény (7.1. ábra), hogy a $[x, 2x]$ tartományban tükrözve ismétlődjön $[2x, 3x]$ -nél ismét a normál pozícióban, stb. A $[0, x]$ szakasz ilyen módon történő folytatásából két állítás következik:



7.1. ábra

a kapott függvény szimmetrikus minden $t = kx$ függőlegesre.

Írjuk fel az (6.3) és az (6.4) szerint a_n és b_n együtthatókat (figyelembe véve hogy $T=2x$).

$$a_n = \frac{2}{X} \int_0^{2X} f(x) \cdot \cos\left(2\pi n \frac{x}{2X}\right) dx$$

$$b_n = \frac{2}{X} \int_0^{2X} f(x) \cdot \sin\left(2\pi n \frac{x}{2X}\right) dx$$

Látható, hogy b_n minden esetben 0, ugyanis az $f(x)$ függvény az integrálási tartomány közepére szimmetrikus, a szinuszfüggvény n bármely értékére antiszimmetrikus, ezért a $[0, X]$ és $[X, 2X]$ tartományon vett integrálok egymást kiejtik. Az a_n együtthatók integrálja alatt levő függvények minden esetben szimmetrikusak az integrálási tartomány közepére, ezért a $[0, 2X]$ -en vett integrál a $[0, X]$ -en vettnek a kétszerese.

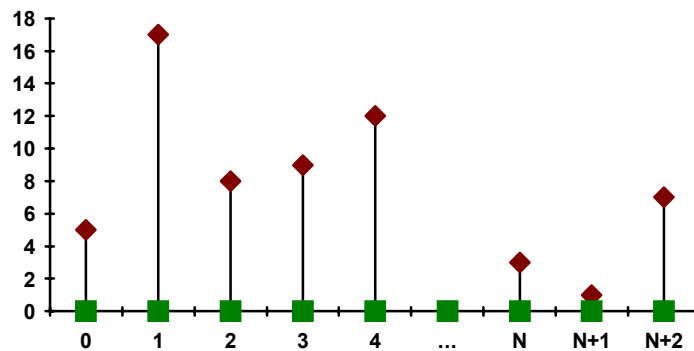
Tehát a koszinusz sorfejtés összefüggései az előzőeket figyelembe véve:

$$f(x) = c_0 + \sum_{n=1}^{\infty} a_n \cdot \cos\left(2\pi n \frac{x}{X}\right) \quad (7.1)$$

$$a_n = \frac{2}{X} \int_0^X f(x) \cdot \cos\left(\pi n \frac{x}{X}\right) dx \quad (7.2)$$

$$c_0 = \frac{1}{X} \int_0^X f(x) dx \quad (7.3)$$

Ahogy a Fourier sornak, úgy a koszinusz sornak is definiálhatjuk a diszkrét értékű függvényekre (6.3. ábra) vonatkozó megfelelőjét.



7.2. ábra

Diszkrét koszinusz transzformáció

$$F_n = \sum_{i=0}^{N-1} f_i \cos\left(\frac{\pi}{N} n \left(i + \frac{1}{2}\right)\right) \quad 0 \leq n \leq N-1 \quad (7.4)$$

$$f_k = \frac{2}{N} \sum_{i=0}^{N-1} F_n \cos\left(\frac{\pi}{N} n \left(k + \frac{1}{2}\right)\right) - \frac{F_0}{N} \quad 0 \leq k \leq N-1 \quad (7.5)$$

ahol f_i az i . függvényérték az N vizsgáltból.

N : ennyi elemből áll az f_i értéksorozat

Látható hogy az $F(x)$ függvényt most pusztán koszinuszos tagokból álló sor összegeként tudtuk visszaállítani. Innen a koszinusz sorba fejtés elnevezés. A transzformációt leellenőrizhetjük, ha F_n összefüggését k -be helyettesítjük, elemi trigonometriai számítások után azonoságra jutunk.

A DCT-nek vannak előnyös vonásai a DFT-vel szemben. Határozott előny hogy a valós f_i bemeneti vektorból valós F_n transzformált vektort kapunk; a számítások valós aritmetikával végigvihetők. Megmaradt az a fontos vonás, hogy a függvényt felharmonikusokra bontottuk, tehát az F_n vektor a frekvenciák szerinti leírást adja. Ez az a tulajdonság amiért előszeretettel alkalmazzák a különböző képfeldolgozási eljárásoknál.

A DCT „kétirányú” transzformáció: „előrefelé” FDCT (Forward DCT), és „visszafelé” IDCT (Inverse DCT), és teljesen reverzibilis. Amiből következik hogy, (ideális esetben) $IDCT(FDCT(x))=x$.

Legyen a minták lépésköze Δx akkor i helyébe $x/\Delta x$ -et írva:

$$f(t) = \frac{2}{N} \sum_{u=0}^{N-1} F_u \cos\left(\frac{2\pi}{2N\Delta x} u \left(x + \frac{\Delta x}{2}\right)\right)$$

látható, hogy a periódushossz $2N\Delta x$ -től majdnem $2\Delta x$ -ig sorakozik a sorfejtésben. Tehát ha a mintákat olyan függvényből vettük, amelynek $2\Delta x$ -nél kisebb periódushosszú összetevője nincsen, úgy az F_n együtthatók egyértelműen a harmonikus összetevők amplitúdóját adják.

Egy fekete-fehér kép kétváltozós $f(x,y)$ függvénnyel írható le, amely a kép világosságát adja meg a hely függvényében. A képfeldolgozásban a koszinusz sorfejtést kétdimenziós függvényre kell alkalmazni, így kétdimenziós koszinusz sorra van szükség.

A kétdimenziós diszkrét koszinusz transzformáció: kétdimenziós bemenő diszkrét értékű függvény transzformációja:

$$f_{xy} \quad 0 \leq x < N \quad 0 \leq y < M$$

A DCT együtthatók kétdimenziós mátrixot alkotnak:

$$F_{uv} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f_{xy} \cos\left(\frac{\pi}{2N} u(2x+1)\right) \cos\left(\frac{\pi}{2M} v(2y+1)\right)$$

$$0 \leq u < N-1$$

$$0 \leq v < M-1$$

vagyis a transzformált egy $N \times M$ méretű kétdimenziós mátrix. Ha u vagy v nagyobb, mint N vagy M , akkor ezek az információk már nem mondanak újat, ugyanis

$$F_{u+N,v} = F_{uv} \text{ és } F_{u,v+M} = F_{uv}$$

Ez azt jelenti, hogy F mátrix elemei az u, v síkon mindkét irányba ciklikusan ismétlődnek.

Az inverz transzformáció képlete:

$$f_{xy} = \frac{2}{N \cdot M} \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} F_{uv} \cos\left(\frac{\pi}{2N} u(2x+1)\right) \cos\left(\frac{\pi}{2M} v(2y+1)\right) - \frac{F_{00}}{N \cdot M}$$

Bontsuk fél a koszinuszos szorzatot a $\cos \alpha \cdot \cos \beta = \frac{1}{2} \cos(\alpha + \beta) + \frac{1}{2} \cos(\alpha - \beta)$ ismert trigonometrikus összefüggés alapján:

$$\begin{aligned} & \cos\left(\frac{\pi}{N}u\left(x + \frac{1}{2}\right)\right)\cos\left(\frac{\pi}{M}v\left(y + \frac{1}{2}\right)\right) = \\ & = \frac{1}{2}\cos\left(\frac{2\pi}{2N\Delta x}u\left(x' + \frac{\Delta x}{2}\right) + \frac{2\pi}{2M\Delta y}v\left(y' + \frac{\Delta y}{2}\right)\right) + \\ & + \frac{1}{2}\cos\left(\frac{2\pi}{2N\Delta x}u\left(x' + \frac{\Delta x}{2}\right) - \frac{2\pi}{2M\Delta y}v\left(y' + \frac{\Delta y}{2}\right)\right) \end{aligned}$$

ahol x' és y' az x illetve y irányú folytonos koordináta, Δx illetve Δy a mintavételi lépésköz.

Látható, hogy $u, v=0$ esetén legnagyobb a periódushossz, tehát a frekvencia ekkor a legkisebb, u és v növelésével a periódushossz csökken, tehát ekkor a frekvencia nő. Elmondható tehát hogy az F_{00} az „egyenáramú” összetevő, míg F_{NM} a legnagyobb frekvenciás összetevő.

A bevezetőben említett fiziológiai okok miatt a frekvenciatartományban végzett módosítások, adatok elhagyása, felbontás durvítása nagyobb frekvenciákon számottevően nem rontja az eredeti kép minőségét, a képadatokban tömörítés érhető el, amely entrópia kódolással még tovább tömöríthető.

Érdekes megfigyelni, hogy a diszkrét koszinusz transzformációs kódolás esetén hogy alakul az együttható-amplitúdók eloszlása. A diszkrét koszinusz transzformációnál azt tapasztaltuk, hogy a nagy amplitúdójú együtthatók a transzformált mátrix (0,0) pontja körül sűrűsödnek. Ennél a transzformációnál, egy átlagos kép 8x8-as blokkjánál csak kevéssé változnak a térbeli frekvenciák, tehát jogos az, hogy az együtthatók a (0,0) együttható körül sűrűsödnek. Az előbbieket alátámasztására mutatok példát a következőkben. Diszkrét koszinusz transzformációt és a transzformációs tartományban végzett módosításokat végeztek a TRANSZ nevű program segítségével. A két kiindulási tesztkép a következő:



7.1. kép: CAMMAN

7.2. kép: BABOON

Látható, hogy a CAMMAN kép nagy homogén mezőket tartalmaz, ezzel szemben a BABOON egy nagyon részletdús kép. Ennek a két képnek a vizsgálatával lehet alátámasztani a fentebbi elméleti eredményeket. Először a CAMMAN kép vizsgálatával kezdem. A transzformációt elvégezve, azt tapasztaljuk, hogy a DC együttható aránya 80%. Kiszemelve a kép bal felső 8x8-as blokkjának - amely ránézésre homogénnek tűnik, és jól jellemzi a kép egészét - transzformáltjára a következőket kapjuk:

285	-4	6	0	-1	-4	-3	5	
-8	1	-2	-2	-1	2	-3	-1	
4	-2	2	-2	0	4	-1	-5	
-1	3	1	-1	-1	-3	4	0	
-1	3	0	-1	-1	-1	0	4	
-1	-1	-1	0	-1	-1	0	1	
3	2	-1	-1	2	2	0	-1	
0	1	3	0	0	3	4	0	

7.1. mátrix

Jól látható a DC tartalom jellemző volta. Azonban ha a transzformációs tartományban csak a DC együtthatókat tartjuk meg, akkor a kép részletei elvesznek, a blokkok átlagvilágosságát kapjuk meg:

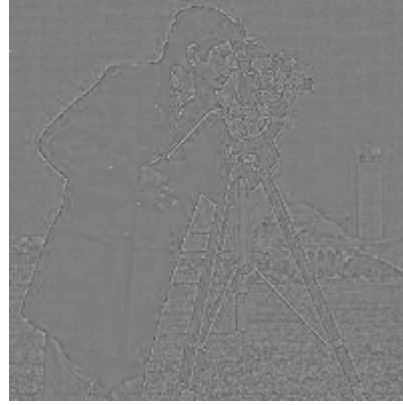


7.3. kép: CAMMAN, csak DC együtthatók

Ha a transzformációs tartományban az emberi szem látási tulajdonságait figyelembe vevő kvantálási mátrix segítségével kvantálunk, a következő eredményeket kapjuk (az alkalmazott mátrix az MPEG szabványban leírt *intra_quantizer_matrix*):

8	16	19	22	26	27	29	34	
16	16	22	24	27	29	34	37	
19	22	26	27	29	34	34	38	
22	22	26	27	29	34	37	40	
22	26	27	29	32	35	40	48	
26	27	29	32	35	40	48	58	
26	27	29	34	38	46	56	69	
27	29	35	38	46	56	69	83	

7.2. mátrix



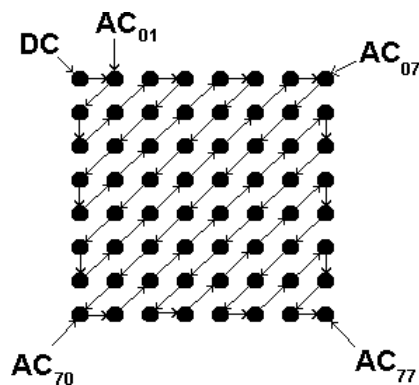
7.4. kép: CAMMAN kvantálva az adott kvantálási mátrixszal, és a különbségi kép

tehát a kép minősége nem romlik jelentősen, de a blokkosodás már megfigyelhető és az objektív paraméterek romlanak. A bal felső blokk alakulása a transzformációs tartományban a kvantálás után:

$$\begin{vmatrix} 35 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

7.3. mátrix

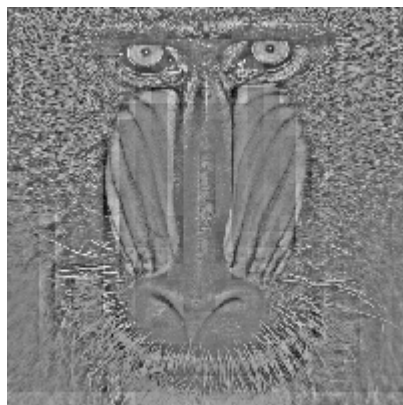
Látható, hogy ez már valóban megfelelő alapot szolgáltat a tömörítésre a nullák nagy száma miatt. Egy lehetséges megoldás a nullák futamhossz-kódolása, majd ezek Huffman- vagy aritmetikai kódolása. A kérdés az a futamhossz-kódolás kapcsán, hogy az együtthatókat milyen sorrendben vegyük figyelembe. Erre találták ki az ún. zig-zag letapogatást, amely tulajdonképpen növekvő térbeli frekvencia szerint állítja sorba az együtthatókat, így a képet kevésbé jellemző és durvábban kvantált, s emiatt gyakran nulla értékű nagyfrekvenciás együtthatók kerülnek az így nyert lineáris struktúra végére:



7.3. ábra: a zig-zag letapogatás

Természetesen a dekóder oldalon szükséges ugyanezzel a mátrixszal való felszorzás inverz transzformáció előtt.

A következő vizsgált kép a BABOON. Ha ennél vizsgáljuk a DC együtthatók súlyát, akkor azt tapasztaljuk, hogy az lényegesen kevesebb, mint az előző képnél volt, 48%. Ez egyezik azon fentebbi nézettel, hogy a részletdús képek több nagy 0 tér-frekvenciás összetevőt tartalmaznak. Érdeemes megvizsgálni tehát a nem nulla frekvenciás komponensek milyen arányban szólnak bele a kép megjelenésébe:



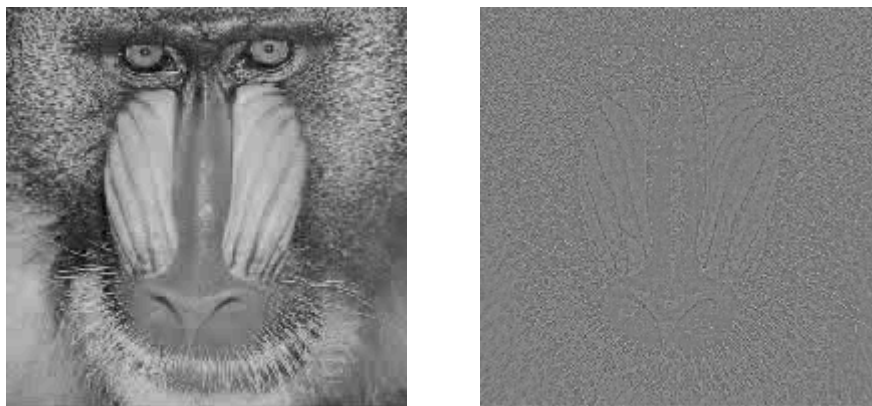
7.5. kép: BABOON csak AC együtthatókkal

Valóban észrevehető, hogy a részletek a nagyfrekvenciás tartományban vannak. A kép bal felső 8x8-as blokkjának transzformáltja:

327	-59	-58	-30	46	-37	-43	10
-61	39	65	-21	-82	-42	22	-24
-17	-23	85	51	-13	-38	14	31
31	-27	-1	-6	-4	52	-21	-75
-40	-23	76	-49	80	-28	66	-50
-90	-9	-16	31	47	24	-23	41
-1	18	22	-4	41	-13	13	23
18	-13	6	-32	6	-10	37	-22

7.4. mátrix

Ez is alátámasztja azt, hogy itt a nagyfrekvenciás összetevők jobban beleszámítanak a kép kialakulásába, a DC együttható súlya valóban kevesebb. Ha a transzformációs tartományban ugyanúgy, mint az előbb, az emberi szem látási tulajdonságait figyelembe vevő kvantálási mátrixszal kvantálunk, a következő eredményeket kapjuk:



7.6. kép: BABOON kvantálva az adott mátrixszal és a különbségi kép

Látható egy kis minőségromlás, de ez még nem zavaró mértékű. Ha vizsgáljuk a bal felső blokk transzformáltjának alakulását:

$$\begin{vmatrix} -40 & -3 & -3 & -1 & 1 & -1 & -1 & 0 \\ -3 & 2 & 2 & 0 & -3 & -1 & 0 & 0 \\ 0 & -1 & 3 & 1 & 0 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 1 & 0 & -1 \\ -1 & 0 & 2 & -1 & 2 & 0 & 1 & -1 \\ -3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

7.5. mátrix

azt figyelhetjük meg, hogy a nullák száma kevesebb, mint az előző képen volt, de még így is elégséges marad arra, hogy ha a fent említett lépéseket végrehajtjuk, tömörítést érzünk el, különösen ha figyelembe vesszük, hogy nem áll az egész kép ilyen nagyon részletdús blokkokból, habár kétségtelenül ez jellemzi a képet.

Az előzőekből kiderül, hogy diszkrét koszinusz transzformáció alkalmazása esetén megfelelő kvantálási mátrixot alkalmazva hatékony tömörítés érhető el. Ezt a transzformációs kódolást alkalmazzák például az állóképekre kidolgozott JPEG szabványban leírt eljárások, és a mozgóképek kódolására kidolgozott MPEG szabványban, valamint a képtelefonoknál.

A DCT definíciója a JPEG szabvány szerint

A DCT transzformációs eljárás annyira jónak bizonyult, hogy a CCITT a JPEG ajánlásában szabványosította. Sok képfeldolgozással, képátvitellel kapcsolatos területen ezt a transzformációt alkalmazzák, és sokan a szabványosított alakját tekintik alapnak egy-egy rendszer specifikációjának elkészítésénél. Várhatóan, ahogy kezdenek megjelenni a távközlő és adatátviteli hálózatokon az álló- és mozgóképek-átviteli szolgáltatások, újabb követelmények fognak megjelenni a fogyasztói piacon is.

A JPEG és a képtelefon, a DCT-nek és az IDCT-nek ugyanazt a formáját használja. A jelentőségére és a fontosságára való tekintettel a továbbiakban ismertetem a DCT és az IDCT JPEG ajánlásban definiált matematikai alakját.

Az egydimenziós DCT és IDCT

Bemenete az $f(x)$ függvényből vett N darab mintavételi érték, ami tulajdonképpen egy N elemű egydimenziós mátrix- és a kimenete ugyancsak egy N elemű transzformált mátrix.

FDCT:

$$F_{(u)} = \frac{C_{(u)}}{2} \sum_{x=0}^7 f_{(x)} \cos\left((2x+1)\frac{u\pi}{16}\right)$$

IDCT:

$$f_{(x)} = \sum_{u=0}^7 \frac{C_{(u)}}{2} F_{(u)} \cos\left((2x+1)\frac{u\pi}{16}\right)$$

Ahol:

$$C_{(u)} = \frac{1}{\sqrt{2}} \quad \text{ha } u = 0$$

$$C_{(u)} = 1 \quad \text{ha } u > 0$$

$f_{(x)}$ = a mintavételezett értékekkel

$F_{(u)}$ = az egydimenziós DCT transzformáltakkal

A kétdimenziós FDCT és IDCT (ezt használják a képfeldolgozásban)

FDCT:

Bemenete egy kétdimenziós $\mathbf{M}_{(x,y)}$ mátrix, amit egy egyváltozós folytonos valós függvény mintavételi értékeinek is felfoghatunk úgy, hogy az \mathbf{M} mátrix soronkénti kiolvasásából kapott értékek az $f(i)$ függvény i helyen vett értékei ahol i tart $x*y$ -ig $i = 0$ -tól $i = x*y$ -ig

$$F_{(u,v)} = \frac{C_{(u)}}{2} \frac{C_{(v)}}{2} \sum_{y=0}^7 \sum_{x=0}^7 f_{(x,y)} \cos\left((2x+1)\frac{u\pi}{16}\right) \cos\left((2y+1)\frac{v\pi}{16}\right)$$

IDCT:

Bemenete az előző transzformáció kimenetként kapott transzformált mátrix (a szakirodalomban $\mathbf{S}_{(v,u)}$ -nak szokták jelölni) és kimenetként megkapom az eredeti bemenő $\mathbf{M}_{(x,y)}$ mátrixot.

$$f_{(y,x)} = \frac{1}{4} \sum_{v=0}^7 \sum_{u=0}^7 C_{(v)} C_{(u)} F_{(v,u)} \cos\left((2x+1)\frac{u\pi}{16}\right) \cos\left((2y+1)\frac{v\pi}{16}\right)$$

Ahol:

$$C_{(u)} = \frac{1}{\sqrt{2}}, \quad \text{ha } u = 0$$

$$C_{(u)} = 1, \quad \text{ha } u > 0$$

$$C_{(v)} = \frac{1}{\sqrt{2}}, \quad \text{ha } v = 0$$

$$C_{(v)} = 1, \quad \text{ha } v > 0$$

$f_{(y,x)}$ = a kétdimenziós mintavételezett értékek

$F_{(v,u)}$ = pedig a kétdimenziós DCT transzformáltak

A DCT gyakorlati megvalósítása [1]

A műszaki leírások és szabványok (mint pl. a JPEG ajánlás) legtöbbször nem tartalmaznak konkrét ajánlást a DCT algoritmus megvalósítására vonatkozóan. Nehéz is lenne olyan algoritmust találni amely minden rendszeren egyformán kedvező paraméterekkel futna. Sok kutató és kutatócsoport dolgozott és dolgozik azon, hogy egy-egy hardverre minél gyorsabb algoritmust találjon. Egy sokkal elterjedtebb és általánosabb algoritmuson a J. W. Cooley és J. W. Tukey által kidolgozott FFT-n próbálok meg megmutatni azokat a pontokat, ahol egy DFT (DCT) típusú transzformáció műveleti igénye látványosan csökkenthető anélkül, hogy az eredeti definíciós egyenlettől bármiben is eltérnénk.

Az FFT alap algoritmus

Vizsgáljuk meg közelebbről a koszinusz transzformált összefüggését:

$$F_n = \sum_{i=0}^{N-1} f_i \cos\left(\frac{\pi}{N} n \left(i + \frac{1}{2}\right)\right)$$

Ha az F_n minden elemét ki akarjuk számolni akkor a \sum alatti műveleteket N^2 -szer kell elvégeznünk (mivel $0 \leq i \leq N-1$ és $0 \leq n \leq N-1$). Nyilvánvalóan ez N^2 szorzást és $N(N-1)$ összeadást jelent. A műveletek időigénye tehát kb.

$$T_{op} = N^2(T_{mul} + T_{add})$$

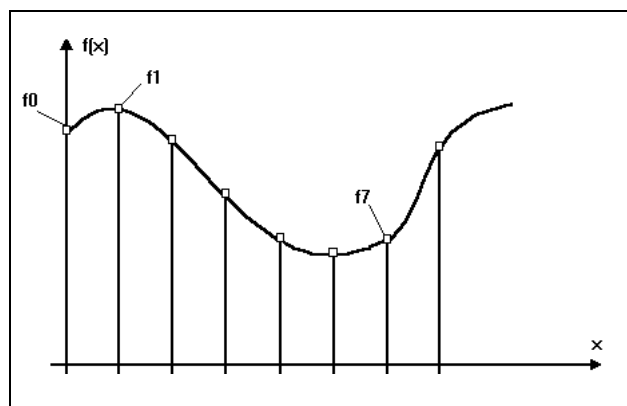
ahol T_{mul} a szorzás és T_{add} az összeadás időigénye. (A harmonikus időfüggvény kiértékelését külön nem számoltam mert, ez összesen N féle argumentummal fordul elő, táblázatosan előre elkészíthető.) Jól látható hogy az alappontok számának növelésével a műveletek száma négyzetesen növekszik. Ez egy határ fölött már irreálisan nagy számítási időt jelent.

Vegyünk például egy $f(t)$ általános időfüggvényt, amit a mintavételezés után egy rövid szakaszán nyolc alappontjával adunk meg, (8.1. ábra). Ezek után kíváncsiak vagyunk az alábbi képlet szerint számítható diszkrét Fourier transzformált értékekre.

$$D8_r = \sum_{i=0}^7 f_i e^{-j2\pi \frac{1}{8} ir} \text{ ahol } 0 \leq r \leq 7.$$

(A számítások egyszerűsítése miatt a továbbiakban mellőzöm az $1/N$ szorzó kiírását.)

A $D8_r$ -nek összesen nyolc független értéke lesz ($r = 0 \dots 7$), a többi ezek periodikus ismétlése. Ezt a nyolc értéket kell kiszámolunk.

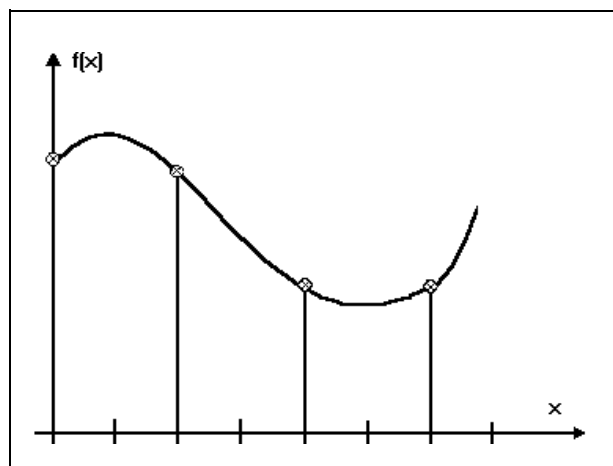


8.1. ábra

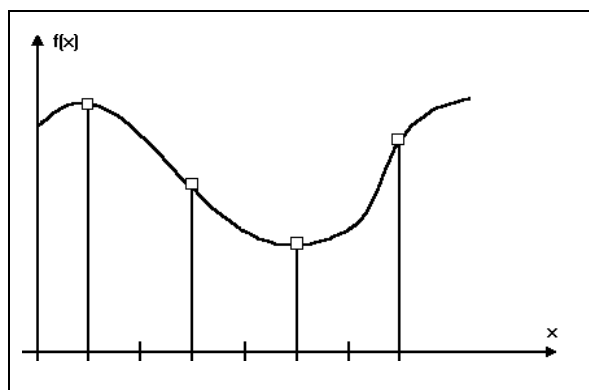
Tegyük fel hogy fel hogy már ismerjük egy durvább, négy alappontra támaszkodó Fourier transzformáltját az eredeti $f(t)$ függvénynek. Ez nyilván kevesebb információt tartalmaz, mint a nyolc alappontra támaszkodó transzformált, hiszen belőle az eredeti $f(t)$ függvénynek csak négy helyen vett értéke állítható vissza inverz transzformációval. Próbáljuk meg pótolni úgy az információhiányt, hogy az eredeti függvényt nem a nyolc alappontra támaszkodó transzformáltjával, hanem két egymással nem egybeeső négy alappontos transzformálttal jellemzem. Ha ezt a kettőt egyenként visszatranszformáljuk és „egymásra illesztjük”, ugyanúgy az $f(t)$ függvényből nyolc különböző helyen vett értéket kapunk, mint az első esetben. Mivel a mintavett értékek között a távolság nem biztos hogy állandó lesz, ezért tegyük ezt a következőképpen: mintavételezzünk lassabban, feleannyi $(1/2f_1)$ frekvenciával az $f(t)$ függvénynek ugyanazt a szakaszát kétszer, úgy hogy először csak a páratlan f_i értékekre essék mintavétel, utána pedig a párosokra. számítsuk ki a transzformált értékeket külön a páros és külön a páratlan mintavételi pontokra, és jelöljük ezt ((8.2. ábrán) f_0, f_2, f_4, f_6) $D4_r^{\otimes}$ -el, és ((8.3. ábrán) f_1, f_3, f_5, f_7) $D4_r^{\oplus}$ -el. Vegyük észre hogy ennek a két transzformálnak az eredményei megegyeznek az eredeti (f_1) frekvenciával mintavételezett függvényből számolt $D8_r$ értékekkel. úgy hogy:

$$\begin{aligned}
 D4_0^{\otimes} &= D8_0 & D4_0^{\oplus} &= D8_1 \\
 D4_1^{\otimes} &= D8_2 & D4_1^{\oplus} &= D8_3 \\
 D4_2^{\otimes} &= D8_4 & D4_2^{\oplus} &= D8_5 \\
 D4_3^{\otimes} &= D8_6 & D4_3^{\oplus} &= D8_7
 \end{aligned}$$

és



8.2. ábra páros alappontokra (f_0, f_2, f_4, f_6) $D4_r^{\otimes}$ -el



8.3. ábra páratlan alappontokra (f_1, f_3, f_5, f_7) $D4_r^{\oplus}$ -el.

Ez könnyen belátható, hiszen az egyes f_i értékekkel ugyanazt a műveletet hajtottuk végre, a képlet nem változott, csupán a műveletek száma. A páros $D4_r^{\otimes}$ és páratlan $D4_r^{\oplus}$ alappontokhoz tartozó transzformáltak matematikai alakja a következőképpen néz ki:

$$D4_r^{\otimes} = \sum_{i=0}^3 f_{2i} e^{-j2\pi\frac{1}{4}ir} \quad 0 \leq r \leq 3 \quad (8.1)$$

$$D4_r^{\oplus} = \sum_{i=0}^3 f_{2i+1} e^{-j2\pi\frac{1}{4}ir} \quad 0 \leq r \leq 3 \quad (8.2)$$

mindkét négy alappontú transzformáltra igaz, hogy 4-el periodikus, azaz

$$D4_{r+4} = D4_r$$

Vizsgáljuk meg a fenti összefüggéseket a műveleti idő szempontjából. Amint már rámutattam, hogy ha az eredeti módon számoljuk a $D8_r = \sum_{i=0}^7 f_i e^{-j2\pi\frac{1}{8}ir}$ összefüggést akkor N^2 műveletre, ha

pedig az előbb levezetett összefüggés szerint számolok akkor $\left(\frac{N}{2}\right)^2 * 2$ műveletre van szükség.

Könnyen belátható hogy $\left(\frac{N}{2}\right)^2 * 2 < N^2$ mindig igaz.

Ugyanezen gondolatmenetet rekurzíve alkalmazva a négy alappontos transzformáltakat két darab két alappontos transzformáltból, a két alappontos transzformáltakat két egy alappontos transzformáltból állíthatjuk elő. Az egy alappontos transzformált ahogy az a későbbiekben kiderül nem más, mint a függvény f_i helyen vett értéke.

Ezek alapján már érthető az általános résztranszformált megfogalmazása:

$$D_r^{p,q} = \sum_{i=0}^{\frac{N}{p}-1} f_{ip+q} e^{-j2\pi\frac{p}{N}ir} \quad 0 \leq r \leq \frac{N}{p}-1 \quad (8.3)$$

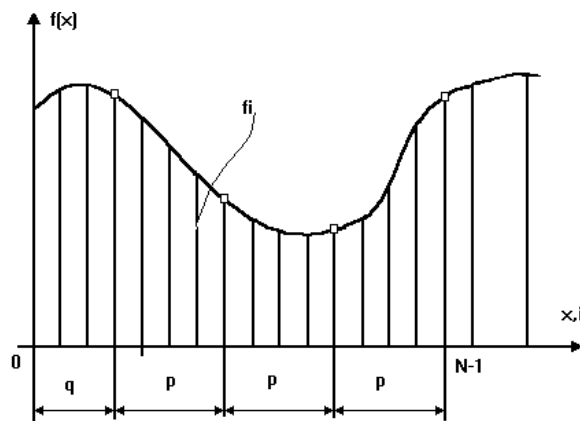
$$0 \leq q \leq p-1$$

Ahol r rész transzformált elemeinek futó indexe, a fölülre írt indexek jelentése pedig:

q a résztranszformált alappont sorozata az f_q alapponton kezdődik,

p a résztranszformált alappont sorozata f_i egymástól p -re levő elemeiből áll (p mindig 2 hatványa).

Az értelmezésben segít a 8.4. ábra.



8.4. ábra

DCT algoritmus az FFT mintájára

A DCT algoritmus megtervezéséhez induljunk ki a koszinusz transzformált definíciójából:

$$F_n = \sum_{i=0}^{N-1} f_i \cos\left(\frac{\pi}{N} n \left(i + \frac{1}{2}\right)\right)$$

Felhasználva hogy $\cos(\alpha + \beta) = \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta)$, ez is írható:

$$F_n = \sum_{i=0}^{N-1} f_i \left(\cos\left(\frac{\pi}{N} in\right) \cos\left(\frac{\pi}{N} \frac{n}{2}\right) - \sin\left(\frac{\pi}{N} in\right) \sin\left(\frac{\pi}{N} \frac{n}{2}\right) \right)$$

Ez rendezés után így alakul:

$$F_n = \cos\left(\frac{\pi}{N} \frac{n}{2}\right) \sum_{i=0}^{N-1} f_i \cos\left(\frac{\pi}{N} in\right) - \sin\left(\frac{\pi}{N} \frac{n}{2}\right) \sum_{i=0}^{N-1} f_i \sin\left(\frac{\pi}{N} in\right)$$

Most pedig írjuk át a fenti összefüggést.

$$K_n = \sum_{i=0}^{N-1} f_i \cos\left(\frac{\pi}{N} in\right) \quad \text{és}$$

$$S_n = \sum_{i=0}^{N-1} f_i \sin\left(\frac{\pi}{N} in\right) \quad -t \text{ behelyettesítve.}$$

$$F_n = \cos\left(\frac{\pi}{N} \frac{n}{2}\right) K_n - \sin\left(\frac{\pi}{N} \frac{n}{2}\right) S_n \quad 1 \leq n \leq N-1 \quad (8.3)$$

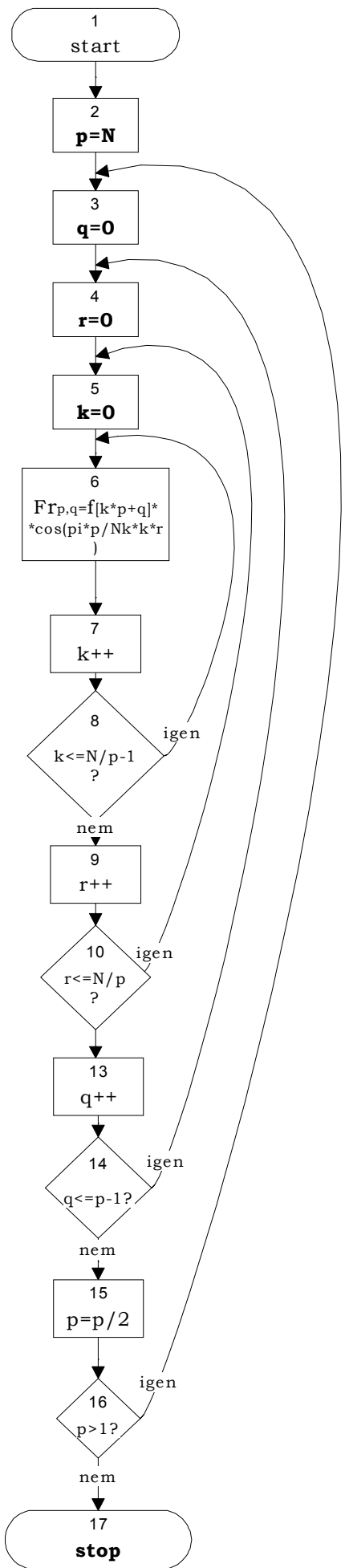
A számítás alapjául célszerűen a K_n és S_n szummák előállítását érdemes választani, és ekkor a feladat lényegi részét már meg is oldottuk, a transzformáció számításigénye jelentősen lecsökkent. Látható, hogy az algoritmus menete az előbb bemutatott FFT algoritmushoz hasonló lesz.

Az általános transzformáltakat az alábbi módon definiálhatjuk:

$$K_n^{p,q} = \sum_{i=0}^{N-1} f_{ip+q} \cos\left(\pi \frac{p}{N} kr\right) \quad \text{ahol} \quad 0 \leq n \leq \frac{N}{p} \quad \text{és} \quad 0 \leq q \leq p-1 \quad (8.4)$$

$$S_n^{p,q} = \sum_{i=0}^{N-1} f_{ip+q} \sin\left(\pi \frac{p}{N} kr\right) \quad \text{ahol ugyancsak} \quad 1 \leq n \leq \frac{N}{p}-1 \quad \text{és} \quad 0 \leq q \leq p-1 \quad (8.5)$$

A következő folyamatra az összeg előállításának algoritmusát mutatja.

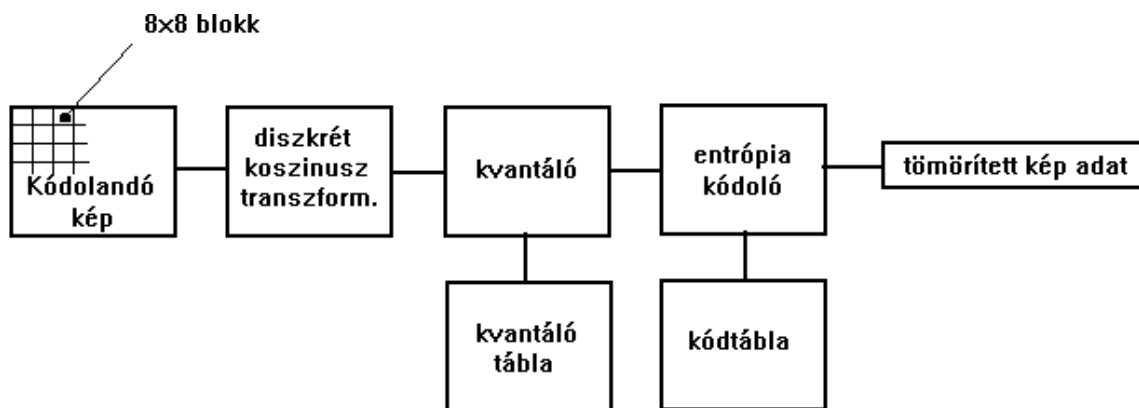


A képtelefonok tömörítésének bemutatása

A tömörítés (9.1. ábra) három lépésben történik. Az eredeti képet az optimális műveleti idő miatt 8x8-as (esetenként 16x16-os) blokkokra bontják fel, ahol a DC és az AC összetevőket egymástól elkülönítve kezelik. Legyen a példának okáért egy fekete-fehér képünk, amely csak világosságinformációt tartalmaz. A kép minden 8x8-as blokkja a következő három lépésen megy keresztül:

- DCT (sík tartományból frekvenciatartományba történő transzformáció)
- Normalizálás (vagy másképpen kvantálás)
- Kódolás

Az információvesztés a 2. lépésnél keletkezhet, ahol a külső normalizálási táblázat az adatvesztés fokát és ezzel együtt a tömörítési hatékonyságot és a képminőséget is meghatározza. A kódoláshoz ugyancsak kell egy táblázat, amely a kódok létrehozását irányítja. (Ha egy ilyen módon tömörített képet tárolni szeretnék, a két táblázatot is le kell tárolni, különben a kitömörítés nem lesz lehetséges).



9.1. ábra

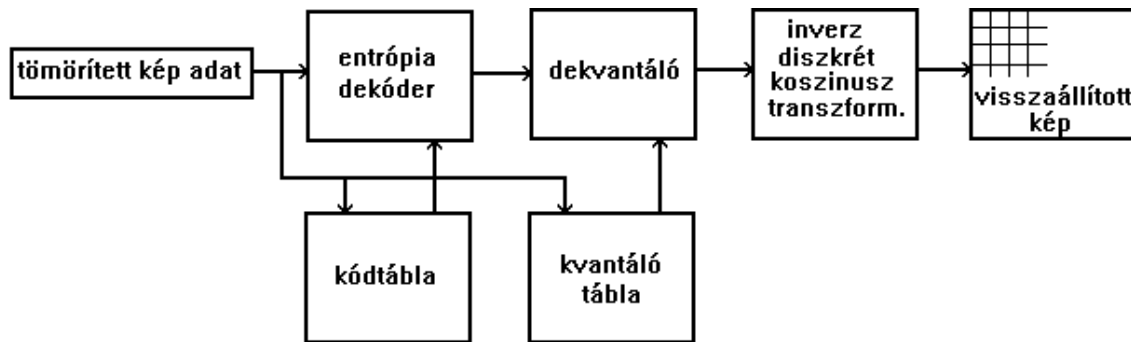
A transzformáció blokkonként történik, eredménye egy 8x8-as mátrix. Ennek bal felső elemei a DC együtthatók, a többi 63 pedig AC együtthatók. Ha a bemenő adatok nem túl erősen és rendszertelenül változnak (ez igaz a természetből vett képeknél) akkor legtöbb információt az alacsony frekvencián található összetevők hordozzák - ezek rendelkeznek a legnagyobb energiával - (a DC-koefficiensnek van a legnagyobb információtartalma). Ilyen esetben sok AC-koefficiensnek nulla, vagy majdnem nulla az értéke, ezért nem kell őket kódolni.

A kvantálás során alkalmazható olyan kvantáló mátrix, amely alkalmazkodik az emberi szem látási tulajdonságaihoz.

A kitömörítés (9.2. ábra) éppúgy három lépésben történik:

- Bináris dekódolás
- Denormalizálás
- IDCT

A DCT alapú dekóder:



9.2. ábra

A normalizálás (kvantálás)

Miután a DCT a 64 együtthatót kiszámította, a kvantálási eljárás következik. Ennek során a 64 együtthatót egy kvantálási mátrix megfelelő elemeivel elosztják. A viszonyítási értékek egy 64 elemű táblázatban találhatóak. A feladattól függően minden alkalmazáshoz más táblázatot választanak. Minden szoftvernek lehet egy saját normalizálási táblázata (elemei 1 és 255 közé esnek). A normalizálás egy erős tömörítést céloz meg, amelynél a DCT összetevők pontosságát csak annyira csökkenti le, hogy a képminőség még megfelelő legyen. Ennek a feldolgozási lépésnek az a célja, hogy azokat az információkat, amelyek vizuálisan értéktelennek bizonyulnak, törölje.

Ebben az összefüggésben ez egy olyan kvantálás tulajdonképpen, amely egyes tényezőknek az értékét egy bizonyos értékre csökkenti.

A JPEG szerint a normalizálás az, ha az S transzformált mátrix elemeit a Q normalizálási táblázat megfelelő elemeivel elosztom.

$$\text{Matematikailag: } N_{i,j} = \text{Integer} \left(\frac{S_{i,j}}{Q_{i,j}} \right) \quad i,j=0,1,\dots,7$$

Az Integer művelet tulajdonképpen egy kerekítést takar (pl.: az 4,67-ből 5 lesz).

A denormalizálás (dequantization) a fenti képlet megfordításából adódik: $S'_{i,j} = N_{i,j} * Q_{i,j}$, ahol $i,j=0,1,\dots,7$.

Az S mátrix nem egyezik az S' mátrixszal, mert a normalizálás folyamán információvesztés történik.

Ha az összetevőknek nagy a frekvenciájuk (tehát az i és a j is nagy), relatíve kevés az információtartalmuk, ezért a normalizálási táblázat megfelelő elemei nagy értékűek. (9.1-es és 9.2-es táblázat) A JPEG bizottság a DIS 10918-1-ben két normalizálási táblázatot ajánl, amelyeket a képfüggő táblázatok helyett fel lehet használni. Az alábbi táblázatokat tapasztalati kísérletek alapján állította össze a JPEG.

9.1. táblázat: Normalizálási táblázat a luminancia-komponensekre.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	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

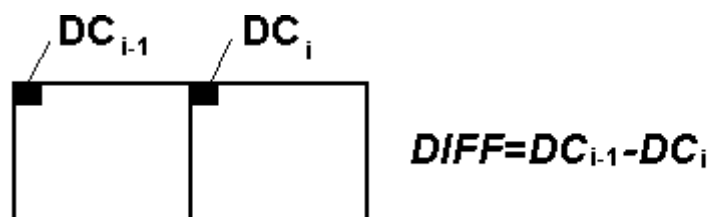
9.2. táblázat: Normalizálási táblázat a krominancia-komponensekre.

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	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

Látható hogy a táblázatokból hogy a krominancia komponensekre sokkal nagyobb tömörítés érhető el mint a luminancia komponensekre. A megfelelő normalizálási táblázat megtalálása egyedül a szoftver feladata. A választás függ a tömörítendő képtől és a kép minőségétől. Az 9.1-es táblázat egy szín luminancia-komponensének a normalizálási táblázatát mutatja. A 9.2-es táblázatban a krominancia-komponensekre vonatkozó értékek találhatóak.

DC-kódolás és cikk-cakk szekvencia

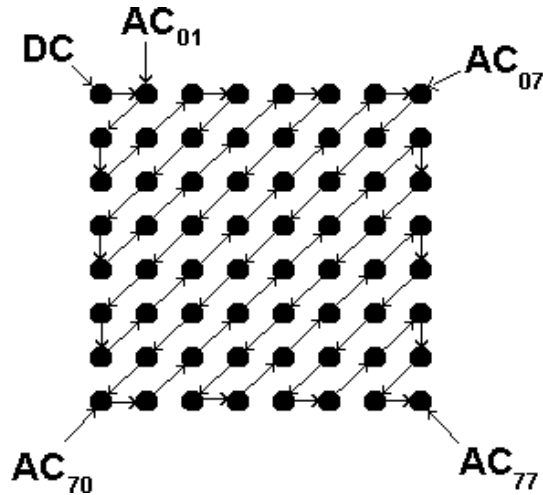
A DC-összetevők szoros kapcsolatban állnak egymással, az előző blokk különbségeként kódolják őket a jobb tömöríthetőség érdekében (lásd a 9.3. ábrát).



9.3. ábra

A normalizálás után az \mathbf{N} mátrix elemeit lineárisan sorba kell rakni. A DC-összetevőket az AC-összetevőktől elkülönítve kezelik.

Az AC-összetevőket a cikk-cakk elrendezés segítségével állítják sorba. Így a sorban először az alacsony frekvenciájú AC-összetevők jönnek, majd a magasabb frekvenciájúak. Maga az eljárás a következő ábrán látható. Erre a bonyolult elrendezési módszerre azért van szükség, mert így jobban lehet majd tömöríteni a bináris kódolás során az adatokat. Ennek az az oka, hogy sok magasabb frekvenciájú összetevő értéke 0 lesz a normalizálás után.



Ez az egydimenziós szekvencia kerül kódolásra.

Entrópia kódolás

Látható hogy a normalizálás során, az emberi látás frekvenciaérzékenysége miatt az információk egy részét elhanyagolhatjuk. Az így nyert adatokat további veszteségmentes tömörítés alkalmazásával, még hatékonyabban lehet tárolni és átviteli csatornán továbbítani.

Az entrópia

Egy meglehetősen fiatal tudományág, az informatika foglalkozik a problémával, amely leírja az üzenet útját a forrásból, a csatornán át, a vevőig. Az üzenet nagyon lényeges tulajdonsága az információtartalma. Például két ember egymás közötti beszélgetése is üzenetek küldése, vétele. Egyértelműleg mindketten igyekeznek a közlésekre szolgáló csatornát jól kihasználni, nem feleslegesen beszélni. Általános a törekvés tehát, hogy az üzenetek információtartalma magas legyen. C. E. Shannon (az információelmélet megalapozója) munkája alapján vált objektíven mérhetővé az egyes üzenetek információtartalma.

Az üzenet N lehetséges elemből (jelkészletből) kiválasztott n elemből áll. Az i -edik elem előfordulási valószínűsége p_i (ahol $0 \leq i \leq N$) tehát $\sum p_i = 1$). A digitalizálás során egy adatállomány (vagy leendő adatállomány), ami byte-okból épül fel, 256 lehetséges értéket tartalmaz ($N=256$). Shannon javasolta a kettes alapú logaritmust, a mérendő információ egységéhez - a bithez - igazodva.

Az üzenet információtartalmát az entrópiája jellemzi. Amekkora az üzenet információtartalma (amit a következő képlettel kapunk meg), jelenként annyi bittel lehet(ne) kódolni, minimálisan annyi bit fejezi ki azt az információt.

$$H = -(p_1 \log_2 p_1 + p_2 \log_2 p_2 + \dots + p_n \log_2 p_n)$$

másképpen:

$$H = -\sum p_i \log_2 p_i$$

Ha valamelyik elem előfordulási valószínűsége 1, azaz biztosan tudható, hogy melyik elem fog következni, akkor az üzenet entrópiája (információtartalma) 0, hiszen ha egy olyan üzenet érkezik amelyről biztosan tudni, hogy mi lesz a tartalma - tehát minden jel előre ismert -, akkor annak nincs információértéke. Például ha megkérdezzük egy éppen felébredt embertől, hogy felébredt-e, akkor a $H=0$.

Az üzenet információtartalma akkor a legnagyobb, ha minden egyes lehetséges elem egyenlő valószínűséggel fordul elő. Ha az érkező üzenet egyes elemei teljesen véletlenszerűen oszlanak el, vagyis nem tudjuk előre, hogy mi lesz a következő karakter, akkor annak az üzenetnek az információtartalma lesz a lehető legnagyobb ($p_1=p_2=\dots=p_n$). Példa rá a pakliból kapott kártya vagy a jövő heti lottószámok.

Fontos kapcsolódó fogalmak a *relatív entrópia*, ez az aktuális és a maximális entrópia viszonya, valamint a *redundancia*, ami $(1 - \text{relatív entrópia}) \times 100\%$, és azt jelenti, hogy az üzenet hány százalékát lehetne elhagyni anélkül, hogy annak információtartalma csorbulna.

Tehát egy üzenetnek akkor a maximális az információtartalma, ha az egyes karakterek (jelek) egyenlő valószínűséggel fordulnak elő. Az előbbi egyenlet átalakítva:

$$H_{\max} = -(\sum p_i) \log_2 p_i$$

Mert itt $p_1=p_2=\dots=p_n$, $p_i=1/n$ és $p=1/256$, hiszen egyenlőek a 256-os készletből választott egyes jelek előfordulásának valószínűsége.

Az előbbieket a képletbe behelyettesítve azt kapjuk, hogy 256 különböző jelből felépülő üzenet maximális információtartalma jelenként 8 bit: tehát egy jel (esetünkben egy byte) 8 bitnyi információt tartalmaz.

A tömörítés mértékét két számmal szokták jellemezni: a zömítési aránnyal, ez az eredeti és a tömörített méret hányadosa (pl.: 5:1) vagy pedig a gyakoribb módszer alkalmazásával, a tömörítéssel, ez pedig $(1 - (1/\text{zömítési arány})) \times 100\%$ (pl. 80%).

A Huffman kódolás

A kódolás általában kétféle: Huffman kódolás vagy aritmetikai kódolás. Mindkét tömörítési eljárást az ISO szabvány DIS 10918-1-es szekciója részletesen leírja. A *Baseline JPEG system* csak a Huffman kódolást támogatja.

A bináris kódolási eljárást ebben az esetben kétlépcsős eljárásnak is lehet tekinteni. Az első lépésben a cikk-cakk módszerrel adatfolyammá alakított összetevők ideiglenes szimbólumok sorozatává alakulnak. A második lépésben ezeket a szimbólumokat bináris adatfolyammá konvertálják, amelyben a kódok változó hosszokkal szerepelnek.

A Huffman kódoláshoz több olyan táblázatra van szükség, amelyek a kódok szimbólumokhoz való rendelését tartalmazza. Ugyanúgy, mint a normalizálásnál, a táblázatokat a szoftver biztosítja. A DIS 10918-1 sok ilyen táblázatot említ, amelyeket azonban nem kötelező használni.

Az aritmetikus kódolás külön táblázat használatát nem igényli, mert az adatok be- és kitömörítésénél, az adatok statisztikai jellemzőit vizsgálva építi fel a táblázatot, ezért 5-10%-kal jobbák a tömörítési eredményei, mint a Huffmané. Az aritmetikai kódolást ritkán használják,

mert sokkal összetettebb és sokkal több számítási időt igényel mint a Huffman. A következőkben a Huffman-féle módszerrel történő tömörítés követhető végig egy egyszerű példán.

Az ideiglenes szimbólumok sorozatában minden nem-nulla AC együtthatóhoz a cikk-cakk szekvenciában az öt megelőző nullák számát (RUNLENGHT) és az együttható értékének ábrázolásához szükséges bitek számát (SIZE) rendelik. Így a cikk-cakk sorozatban szereplő együtthatók az alábbi szimbólumpárokkal reprezentálhatók:

(szimbólum-1) (szimbólum-2)
(RUNLENGHT, SIZE) (AMPLITUDE)

A 2. típusú szimbólum tulajdonképpen az AC összetevő amplitúdóját adja meg. A 9.3-as táblázat mutatja meg, hogy az egyes *Amplitude* tartományok milyen *Size* értékek használatát kívánják meg.

9.3. táblázat: Az egyes *Amplitude*-tartományokhoz a *Size* értékének mekkorának kell lennie

Size	Amplitude
1	-1,1
2	-3,-2,2,3
3	-7...-4,4...7
4	-15...-8,8...15
5	-31...-16,16...31
6	-63...-32,32...63
7	-127...-64,64...127
8	-255...-128,128...255
9	-511...-256,256...511
10	-1023...-512,512...1023
11	-2048...-1024,1024...2048

Mind az AC, mind pedig a DC együtthatókhoz tartozó 1. típusú szimbólumot változó hosszúságú kóddal (VLC, Huffman-kód) kódolják.

Ha több Huffman-táblázat áll rendelkezésre, előfordulhat, hogy a kép különböző komponenseit (pl. luminancia, krominancia) más-más táblázat felhasználásával kódolják. A 2. szimbólumokhoz szintén változó hosszúságú kódot rendelnek (variable-length integer code, VLI), de ez nem azonos a Huffman-kóddal. A legfontosabb különbség a két módszer között, hogy a Huffman-kódolt (RUNLENGHT, SIZE) szimbólum hossza a dekódolás pillanatáig nem ismert, míg az (AMPLITUDE) szimbólum hosszát az öt megelőző VLC már meghatározza.

A pozitív számokat egyenes bináris kód szerint ábrázolják, a negatív értékeket pedig az 1-es komplement kód szerint.

Fontos, hogy csak annyi bitet használnak fel, amennyi feltétlenül szükséges (a *Size* és az *Amplitude* közötti összefüggés). A dekódolásnál a hibák ki vannak zárva, mivel az előző 1-es szimbólum a *Size* komponens *VLI* kódjának hosszát tartalmazza. A következő példák megmagyarázzák a *VLI* kódok felépítését:

Érték:	1	-1	2	-2	3	-3	4	-4
<i>VLI</i> kód:	1	0	10	01	11	00	100	011

Látható, hogy a pozitív kódok 1-gyel kezdődnek, míg a negatív *VLI* kódok 0-val. A 8x8-as blokkok DC együtthatóját (differenciális) hasonló szimbólumokkal kódolják, de itt természetesen a RUNLENGTH információra nincs szükség, mert a cikk-cakk sorozatban a DC együttható az első elem és így nem szerepelhetnek előtte zérus elemek.

Az entrópia kódolás hatékonyságának szemléltetésére tételezzük fel, hogy a kvantálás után kapott blokkban a bal felső 3x3-as sarok kivételével az összes együttható zérus.

14	0	-2
-3	-1	0
-2	-1	0

Mivel a DC együtthatót differenciálisan kell kódolni, tegyük fel, hogy a megelőző blokkban a DC értéke 12 volt. Így a DC együtthatót reprezentáló közbenső szimbólumpár: (SIZE=2) és (AMPLITUDE=2). A cikk-cakk irányt követve az első nem nulla AC együttható a -3, amit 1 darab nulla elem előz meg. Erre az együtthatóra a (1,2) (-3) szimbólumpár írható fel. Tovább lépve a cikk-cakk sorozatban, három nem nulla elem következik, amelyekhez a (0,2) (-2), (0,1) (-1), és (0,2) (-2) szimbólumok tartoznak. Az utolsó nem-nulla együtthatót a (1,1) (-1) szimbólumok képviselik, mivel a -1-et egy nulla előzi meg. Egy 8x8-as blokk végét (EOB) a (0) (0) szimbólumpár jelenti. Így a példában szereplő blokkra az alábbi szimbólumok írhatók fel:

(2) (2), (1,2) (-3), (0,2) (-2), (0,1) (-1), (0,2) (-2), (1,1) (-1), (0,0)

A következő lépés a szimbólumokhoz tartozó VLC-k és VLI-k kikeresése a táblázatokból. Tegyük fel, hogy a differenciális együtthatóhoz tartozó VLC az alábbi:

(2) 011

A luminancia blokkok AC együtthatóihoz pedig a következő VLC-k tartoznak:

(0,1) 1010
 (0,2) 11100
 (1,1) 00
 (1,2) 11011

A VLI-kódolás kettes komplementes ábrázolással történik:

(-3) 11
 (-2) 01
 (-1) 0

A példában szereplő 8x8-as blokk kódolása után kapott bitfolyam a következő:

0111101111111000110100111000100000

Látható, hogy a 64 együttható ábrázolásához elegendő 34 bit, ami közel 0.5bit/minta tömörítési arányt jelent.

Mozgáskompenzáció

Igaz hogy az állóképek tömörítéséhez nem kapcsolódik szorosan, de mert a képtelefonoknál a mozgó képek esetén szükség van a hatékonyság miatt erre az adatcsökkentő lépésre, kell ejteni róla néhány szót.

A mozgó képek kódolása esetén kihasználható az egymást követő képek közötti időbeli redundancia. Ennek megvalósítására alapvetően két módszert dolgoztak ki. Ezek

- a pel-rekurzív,
- és a blokkillesztő algoritmusok.

A blokkillesztő eljárásoknál egy meghatározott méretű blokkhoz rendelünk egy mozgásvektort, míg pel-rekurzív eljárásoknál minden pixelhez egy vektor rendelődik.

A mozgáskompenzált kódolók általában a következő fokozatokból állnak [3]:

1. mozgásdetektor, amely kiválasztja a képen belül a mozgást tartalmazó részeket;
2. mozgásbecslő, amely az elmozdulásvektor értékét határozza meg valamilyen stabil becslő algoritmus segítségével;
3. az adattömörítést végző kódoló, amely felhasználja a meghatározott mozgásvektort.

A mozgáskompenzáció tulajdonképpen egy olyan eszköz, amellyel javítható az időbeli kompresszió minősége, ha mozgás van a képen. A mozgáskompenzációnál meg kell határozni hogy mi mozgott a képen és hova mozgott. Ha ezt az információt a dekóder ismeri, akkor ennek segítségével az előző kép eltolható úgy, hogy ezáltal a következő kép sokkal jobb predikcióját kapjuk meg. A kóder végrehajtja ugyanazt a predikciót, amit a dekóder végrehajt, így meg tudja határozni a becsült kép és a tényleges kép közötti különbséget. Ha a mozgás megegyezik a becsült mozgással, akkor a különbségi jel nulla lesz [3].

Az előző kép elmozdulása lehet teljes képes vagy részképes, esetleg pixel alapú. Ennek megfelelően egy-egy egyedi mozgásvektor generálható a teljes képre a kép részeire és a pixelekre. A teljes képre vonatkozó egyetlen mozgásvektor generálásának nem sok értelme van, kivéve azt az esetet amikor a teljes kép mozdul el. Az lenne az ideális ha minden egyes pixelre egy mozgásvektort hoznánk létre. De mivel a mozgásbecslés egy komplex folyamat és megkívánja az előző kép ismeretét, ezért csak a kóder valósíthatja meg. Ezeket a pixel-mozgásvektorokat át kell vinni a dekóder számára, ez azonban nagyon bonyolult és időigényes lenne. Ezért a mozgásvektor-becslés részképekre vonatkozik. Ezt a részképet superblokknak nevezték el, amely vízszintesen 4 DCT blokkból, míg függőlegesen 2 DCT blokkból áll [3].

A mozgásbecslés lehetséges folyamatainál a mozgásbecslési algoritmusokat két osztályba soroljuk [3]: az egyik a háromdimenziós mozgásparamétereket a kétdimenziós egymást követő képekből nyeri ki, a másik a területről területre pixelről pixelre ad becslést a sebességre, függetlenül attól, hogy a tárgyak hogyan mozognak. Az elsőben az általános szabály az, hogy a képet három dimenziós tekintve, azon apró részek vannak, amelyek leírhatók elfordulás-forgás- és zoom paraméterekkel, egy középpontot feltételezve. A képtelefonoknál általában a képek egyszerű felépítésűek, nem követik egymást túl nagy képváltozások, így itt alkalmazhatóak ezek az eljárások, míg egy hagyományos tv-kép sokkal bonyolultabb felépítésű ahhoz hogy ezzel az egyszerű modellel leírható lenne. A második megoldás az előző mozgásleírással szemben unintelligens. Általában csak elmozdulást lehet vele leírni. Ha azonban a terület, amelyre a mozgás meghatározása folyik, kicsi, akkor sokkal bonyolultabb mozgásformákat is lehet írni. Ez a módszer minden ún. superblokkra egy 9 bites mozgásvektor átvitelét teszi lehetővé (0.018bit/pixel).

Nemzetközi szabvány a képtelefon megvalósítására

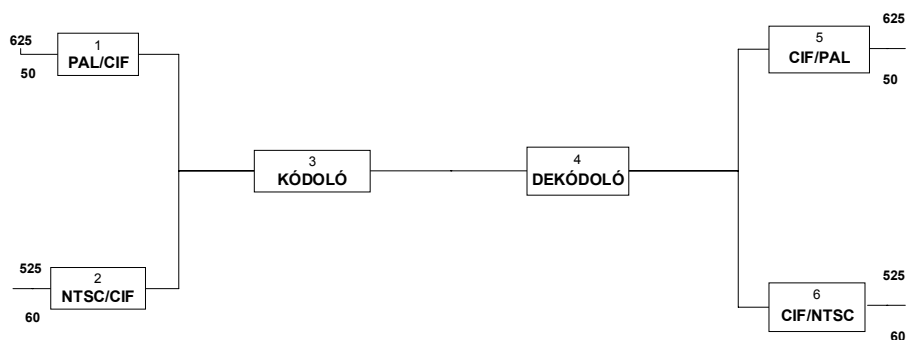
A jelenlegi szabványosítási tevékenység fő jellemzőit, illetve a meglévő és jövőbeli szabványait álló és mozgóképek esetére az 10.1. táblázat foglalja össze [3]. A táblázatban fel vannak tüntetve a szabványosítással foglalkozó csoportok és a fontosabb alkalmazási területek.

10.1. táblázat

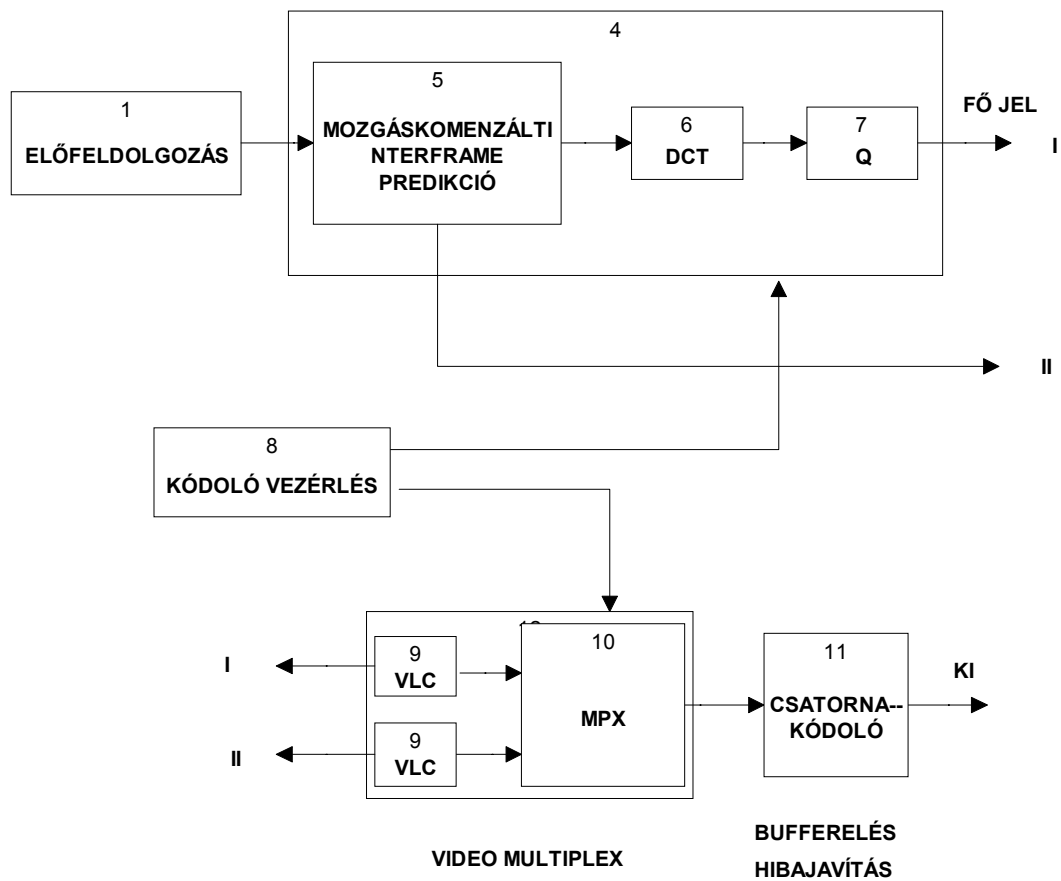
fő alkalmazások	faximile	audio-grafikus	hírközlés, videotext színes fax	video-konferencia, videotelefon	digitális tárolás	televízió-átvitel
szabványosító csoport	CCITT SG VIII	ISO/IEC JTC1/SG2/WG9 (JBIG)CCITT SG VIII	ISO/IEC JTC1/SC2/WG10 (JPEG)CCITT SG VIII	CCITT SG XV	ISO/IEC JTC1/SG2/WG11 (MPEG)	CCIR/CCITT CMTT
szabványok (év)	Rec.T.4 (1980) Rec.T.6 (1984)	IS (1992) rec.T.82 (1992)	IS (CD10918) (1991) Rec.T.81	Rec.H.261 (1990)	IS (CD 11172) (1991)	Rec.721, Rec.723 (1990)

A CCITT Study Group XV felvázolt egy kódoló elrendezést a H.261-es ajánlásban a videotelefon és videokonferencia-szolgáltatás számára. Ez a kodek kis adatsebességeken működik: videotelefon esetében $n \times 64$ kbit/sec ($n=1,2$), videokonferencia-alkalmazásban $m \times 384$ kbit/sec ($m=1-6$)

A világon két televízió szabvány létezik: 625sor/50félkép/sec és 525sor/60félkép/sec. Mivel a videotelefon illetve a videokonferencia terminálok a helyi szabványnak megfelelő kamerákat és monitorokat használnak ezért szükség van egy egységesen használható szabványra amely a helyi TV jeleket konvertálja az egységes 360 pel x 288 sor x 29.97 kép/sec nem váltott soros alakba a forráskódolás céljára. Ez a H.261 ajánlásban definiált (CIF) (Common Intermediate Format) szabvány. A H.261 szerinti videokódoló tömbvázlatát [3] az 1. ábra mutatja. A bemenő CIF kép időfüggő redundanciáját mozgáskompenzált interframe predikció csökkenti, míg a predikciós hibajel térbeli redundanciáját DCT és az azt követő kvantáló minimalizálja. Ennek a mozgáskompenzált hibrid kódolónak a két kimenő adatjelét (kvantált DCT együtt-hatók, mozgásvektorok járulékos információja) változó szóhosszúságú kódoló tömöríti. A legvégén a csatornakódoló a hibajavítás lehetőségét biztosítja.



10.1. ábra CIF szabvány szerinti rendszer



10.2. ábra

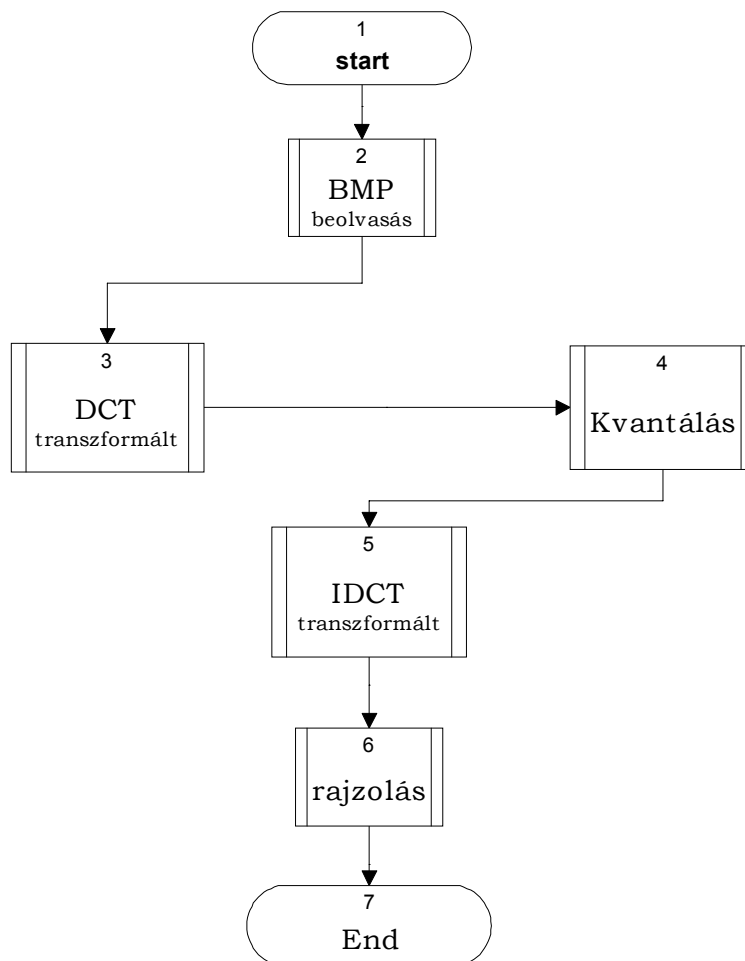
Látható az előzőek alapján, hogy képtelefon esetén (64 kbit/sec) a felbontás 1/4CIF-nek (180x144), míg a CCIR Rec.601 szerinti videojel (625 soros) 4 CIF-nek (750x576 európai szabvány) felel meg [3].

A szimulációs szoftver

Egy nagyon egyszerű programmal megpróbáltam vizuálisan is érzékeltetni az ISDN képtelefon működése közben a kép minőségén bekövetkezett változásokat. A képinformációt a képtelefon kamerája az adott minőségi követelményeket figyelembe véve rögzíti. Ezután a képet DCT segítségével átranszformálják frekvencia-tartományba és kvantálják. A kvantálás után következik a veszteségmentes kódolás, ami nem változtat a kép minőségén, csak a képet leíró numerikus adatok számát csökkenti.

A szoftver ennek megfelelően beolvas először egy tetszőleges, maximum 16 színű BMP képet, azután a bal alsó sarokból indulva 8x8-as blokkonként a képet leíró numerikus adatokon végrehajtja a DCT-t. Az így kapott eredményeket elosztja a JPEG-ben a luminancia adatokra definiált kvantáló mátrixszal, és utána visszatranszformálja. Az így kapott 8x8-as blokkot azután visszateszi a képernyőre. Így a képernyőn először az átvinni kívánt, „hibamentes” kép jelenik meg, azután pedig a transzformáció végrehajtása során fokozatosan az átvitel során keletkező kép jelenik meg, megmutatva a veszteségmentes tömörítés során várhatóan bekövetkező minőségromlásokat. A szoftver a bmp3v.exe-vel indul. Ezután be kell gépelni a vizsgálni kívánt bmp kép nevét az elérési útvonallal együtt. Ezek után már beavatkozás nélkül az előbb leírt folyamatok lefutnak, és a végén, tetszőleges billentyűt megnyomva kiléphetünk a programból.

A program sematikus folyamatábrája a következőképpen néz ki:



A program teljes algoritmusát és forráslistáját valamint külön a DCT rutin algoritmusát a mellékletben található meg.

Összefoglaló

A szakdolgozat során feladatom a képtelefonokban alkalmazott veszteséges tömörítési eljárás bemutatása volt.

A különböző hálózatok, (távközlési-, adat-, számítógép-,) egységesítésére való törekvés egy nemzetközi vállalkozáshoz vezetett, amelynek célja egy XXI. századbeli, a megnövekedett igényeket maximálisan kiszolgálni képes, korszerű digitális távközlési hálózat kiépítése volt. Ennek az új rendszernek elsődleges célja, hogy integrálja mind a beszéd-, mind a nem beszéd-célú átviteli szolgáltatásokat. Ennek a neve ISDN (Integrated Services Digital Network). Az alap-hozzáférésű ISDN-ben definiáltak egy kis - 2x64 kbit/sec-os - átviteli sebességű képtelefon szolgáltatást. Itt a képinformáció átviteléhez a kis sebesség miatt elengedhetetlenül szükséges valamilyen hatékony tömörítési eljárás alkalmazása. A képadattömörítésre a **JPEG** (Joint Photographic Experts Group) társaság által állóképekre kidolgozott veszteséges tömörítési eljárást alkalmazzák. A szakdolgozatomban tulajdonképpen ezt a **kódolási technikát próbáltam bemutatni, különös figyelmet fordítva a diszkrét koszinusz transzformációra és ennek algoritmussal történő megvalósításának alapproblémájára. Igyekeztem érzékelteni azt, hogy milyen manipulációkat lehet eszközölni a képen a frekvenciatranszformáció után, és hogy ez hogyan ad lehetőséget hatékony tömörítésre.**

Végül az egész illusztrálásaként írtam C nyelven egy rövid programot, ami kirajzol egy tetszőleges bmp képet, végrehajtja rajta a transzformációs tömörítést és újra kirajzolja, megmutatva így a transzformáció által okozott változásokat.

Summary

The task of my thesis is to present the compression losses in picture-telephones. In the basic ISDN they defined a 2X64 kbit/sec transmission speed for picture-telephone service. It is of prime importance to use an efficient compression if such slow-speed is used for picture transmission by telephone. The JPEG (Joint Photographic Experts Group) suitably compression is that used. The thesis highlights the problems in this coding technique with special emphasis on cosine transforms and their problems in different algorithms. I tried to show the use of different methods after frequency-transformation and to efficiently utilize compression. Finally to demonstrate everything I wrote a program in „C” which draws the bmp picture and efficiently uses the transformation compression.

Irodalomjegyzék

- [1] Székely Vladimír: Képporrekción, hanganalízis és térszámítás PC-n. Gyors Fourier transzformációs módszerek ComputerBooks Kiadó, 1994.
- [2] Álló Géza - Főglein János - Hegedűs Gy. Csaba-Szabó József: Bevezetés a számítógépes adatfeldolgozásba; BME Mérnöktovábbképző Intézet, Budapest 1993.
- [3] Fazekas Kálmán - Csillag Péter: Képek és képszekvenciák adattömörítő kódolása. Híradástechnika 1991 dec.
- [4] William B. Pennebaker & James A. Storer: JPEG Still Image Data Compression Standard. 1992 New York
- [5] Sólyom Csaba: Digitális képfeldolgozás és képátvitel. PKI Tudományos Napok 1993 november 23-24.
- [6] dr. Ferenczy Pál: Video- és hangrendszerek. Műegyetemi Kiadó, 1994.
- [7] JPEG Joint Photographic Experts Group
ISO/IEC JTC1/SC2/WG8 1990.
CCITT SGVIII
- [8] Thomas W. Lippe: Die Welt der Graphik - Formate, 1993
- [9] Hirling Endre: Televízió műsorszórás

Melléklet

A program algoritmusja és forráslistája

```

/*****
/*
/* Keptelefon szimulacios program
/*
*****/

#include <math.h>
#include <float.h>
#include <conio.h>
#include <dos.h>
#include <stdio.h>
#include <stdlib.h>
#include <graphics.h>

#define X_DEF 200
#define Y_DEF 200
#define WIDTH 18      /* A kep szelessege (X iranyu kiterjedes) a 18-ik byte-tol*/
#define HEIGHT 22     /* A kep szelessege Y iranyu kiterjedes a 22-ik byte-tol*/
#define BPP 28        /* Szinmelyseg (bitek szama bit/pixel)(1,2,4,8,24) a 28 byte-tol*/
#define PAL 54        /* A paletta kezdete a file-ban*/
#define N 8
#define CU 0.7071067812
#define CV 0.7071067812

typedef unsigned char byte;
typedef unsigned int word;

struct PAL_TYPE {byte blue;      /* paletta tipusu struktura*/
                 byte green;    /* a harom alapszin +1 hasznalt*/
                 byte red;
                 byte not_used; };
typedef struct PAL_TYPE PAL_REC; /*paletta record*/

byte szin_hossz[]={0,2,0,0,16}; /*egyenlore csak monokrom es*/
                             /*16 szinu keppel foglalkozunk*/
long word maxx,maxy,maxxold;
double dct_in[N][N];        /*DCT bemeneti tomb*/
double dct_out[N][N];       /*DCT kimeneti tomb*/

int kvant_matrix[N][N]={ {16,11,10,16,24,40,51,61}, /*JPEG szerinti kvantulasi matrix*/
                        {12,12,14,19,26,58,60,55},
                        {14,13,16,24,40,57,69,56},
                        {14,17,22,29,51,87,80,62},
                        {18,22,37,56,68,109,103,77},
                        {24,35,55,64,81,104,113,77},
                        {49,64,78,87,103,121,120,101},
                        {72,92,95,98,112,100,103,99} }

```



```
/******DCT transzformacio******/
```

```
void dct(void) {  
int i,j,u,v;  
double cu,cv;  
  
for(u=0;u<N;u++) {  
for(v=0;v<N;v++) {  
cu=cv=1;  
for(i=0;i<N;i++) {  
for(j=0;j<N;j++) {  
dct_out[v][u]+=(dct_in[j][i])*cos((2*i+1)*u*M_PI/(2*N))*cos((2*j+1)*v*M_PI/(2*N));  
}  
}  
if(u==0) cu=CU;  
if(v==0) cv=CV;  
dct_out[v][u]*=0.25*cu*cv;  
}  
}  
}
```

```
/******IDCT transzformacio******/
```

```
void idct(void) {  
int i,j,u,v;  
double cu,cv;  
  
for(i=0;i<N;i++) {  
for(j=0;j<N;j++) {  
for(u=0;u<N;u++) {  
for(v=0;v<N;v++) {  
cu=cv=1;  
if(u==0) cu=CU;  
if(v==0) cv=CV;  
dct_in[j][i]+cu*cv*dct_out[v][u]*cos((2*i+1)*u*M_PI/(2*N))*cos((2*j+1)*v*M_PI/(2*N));  
}  
}  
dct_in[j][i]*=0.25;  
}  
}  
}
```

```
/******Kvantalas******/
```

```
void kvant(void){  
int x,y;  
for(y=0;y<N;y++){  
for(x=0;x<N;y++){  
dct_out[y][x]=kvant_matrix[y][x];  
}  
}
```

```

}
}

```

```

/*****bmp beolvaso rutin*****/

```

```

void bmpload(void) {
FILE *fp;
long word *mutat;
long word bmp_size;      /*bmp merete*/
int gd,gm;
byte be,munka,m1;
int x,y,i,j;
char file[100];          /*ide kerul beolvasasra a file nev*/
PAL_REC szinek;         /*paletta rec tip., szinek nevu struktura*/

scanf(„%s”,file);      /*bmp nevenek es utvonalanak a beolvasasa*/
detectgraph(gd,gm);     /*agrafikus kepernyo megnyitasa*/
initgraph(gd,gm,„c:\\output”);
setgraphmode(VGAHI);

fp=fopen(file,„rb”);    /*file megnyitasra binaris olvasasra*/
if(fp==NULL) {         /*hibaellenorzes*/
printf(„Hat ez kesz bukas...\n”);
exit(0);
}
fseek(fp,WIDTH,SEEK_SET); /*a elejetol kezdve*/
fread(&maxx,sizeof(long int),1,fp); /*a file mereetinek*/
fread(&maxy,sizeof(long int),1,fp); /*a beolvasasa*/
maxxold=maxx;
while((maxx%32)!=0) maxx++; /*oszthato-e 32-vel*/
fseek(fp,BPP,SEEK_SET); /*kivancsi vagyok,*/
fread(&szinmely,sizeof(word),1,fp); /*a szinmelysegre*/
fseek(fp,PAL,SEEK_SET); /* a paletta elejere allunk*/
for(i=0;i<szin_hossz[szinmely];i++) { /*1 palettarecordot feltolt,*/
fread(&szinek,sizeof(PAL_REC),1,fp); /*es ennek segitsegevel a palettat*/
setrgbpalette(i,szinek.red&63,szinek.green&63,szinek.blue&63); /*beallitja*/
setcolor(i);
}

x=0;
y=getmaxy(); /*az x es y valtozokat a kepernyo bal ,also sarkaba allitja*/
do {
j=0;
do {
fread(&be,sizeof(byte),1,fp); /*beolvassa az elso kepadatot*/
for(i=0;i<(8/szinmely);i++) {
if(szinmely==1) { /*ha a kep csak monokrom*/
munka=be&0x80;
munka>>=7; /*nem kivancsi az also 7 bitre*/
be<<=szinmely;
}
if(szinmely==4) {
munka=be&0xf0;

```

```

    munka>>=4;
    be<<=szinmely;
  }
  if(j<maxxold) putpixel(x,y,munka); /*itt rajzol*/
  x++;
  j++;
}
} while(j<maxx);           /*a sorokat rakja ki*/
x=0;
y--;
} while(!feof(fp));      /*addig megy el amig el nem jut a file vegeig*/
fclose(fp);
}

```

/******A foprogram*****/

```

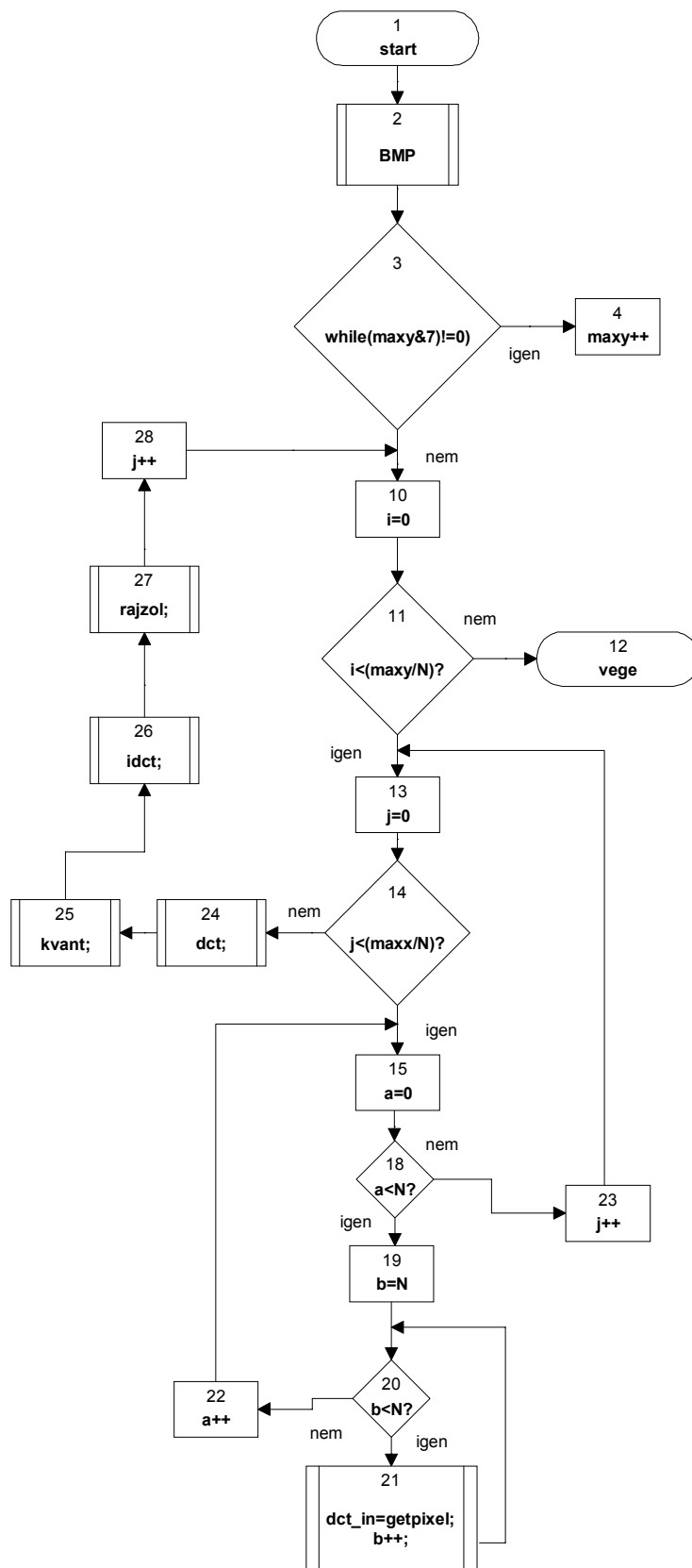
void main(void) {
int i,j,a,b;
double matrix[N][N];
double dct_out[N][N];

bmpload();           /*egy tetszoleges kep beolvasasa*/
while((maxy&7)!=0) maxy++;
for(i=0;i<(maxy/N);i++) { /*8x8-as blokkot olvas be*/
for(j=0;j<(maxx/N);j++) {
for(a=0;a<N;a++) {       /*a blokkon belül mozog*/
for(b=0;b<N;b++) {
dct_in[a][b]=getpixel(j*N+a,getmaxy()-i*N+b); /*beolvassa a pontok szinet*/

putpixel(j*N+a,getmaxy()-i*N+b,0);
} /*b++*/
} /*a++*/
putpixel(getmaxx(),getmaxy()-i*N+b,1);
dct();
kvant();
idct();
for(a=0;a<N;a++) {
for(b=0;b<N;b++) {
putpixel(j*N+a,getmaxy()-i*N+b,(int)dct_in[a][b]);
}
}
}
}
}
}
}
}

```

A program folyamatábrája:



A DCT lineáris algoritmus

A DCT lineáris, lényegében a képlet alapján történő programozását az alábbiakban ismertetem. Induljunk ki a kétdimenziós koszinusz transzformált, JPEG szerinti definíciójából, ahol a transzformációt 8x8-as mátrixokra végzik el:

$$F_{(u,v)} = \frac{C_{(u)}}{2} \frac{C_{(v)}}{2} \sum_{y=0}^7 \sum_{x=0}^7 f_{(x,y)} \cos\left((2x+1)\frac{u\pi}{16}\right) \cos\left((2y+1)\frac{v\pi}{16}\right)$$

Ahol:

$$C_{(u)} = \frac{1}{\sqrt{2}}, \quad \text{ha } u = 0$$

$$C_u = 1, \quad \text{ha } u > 0$$

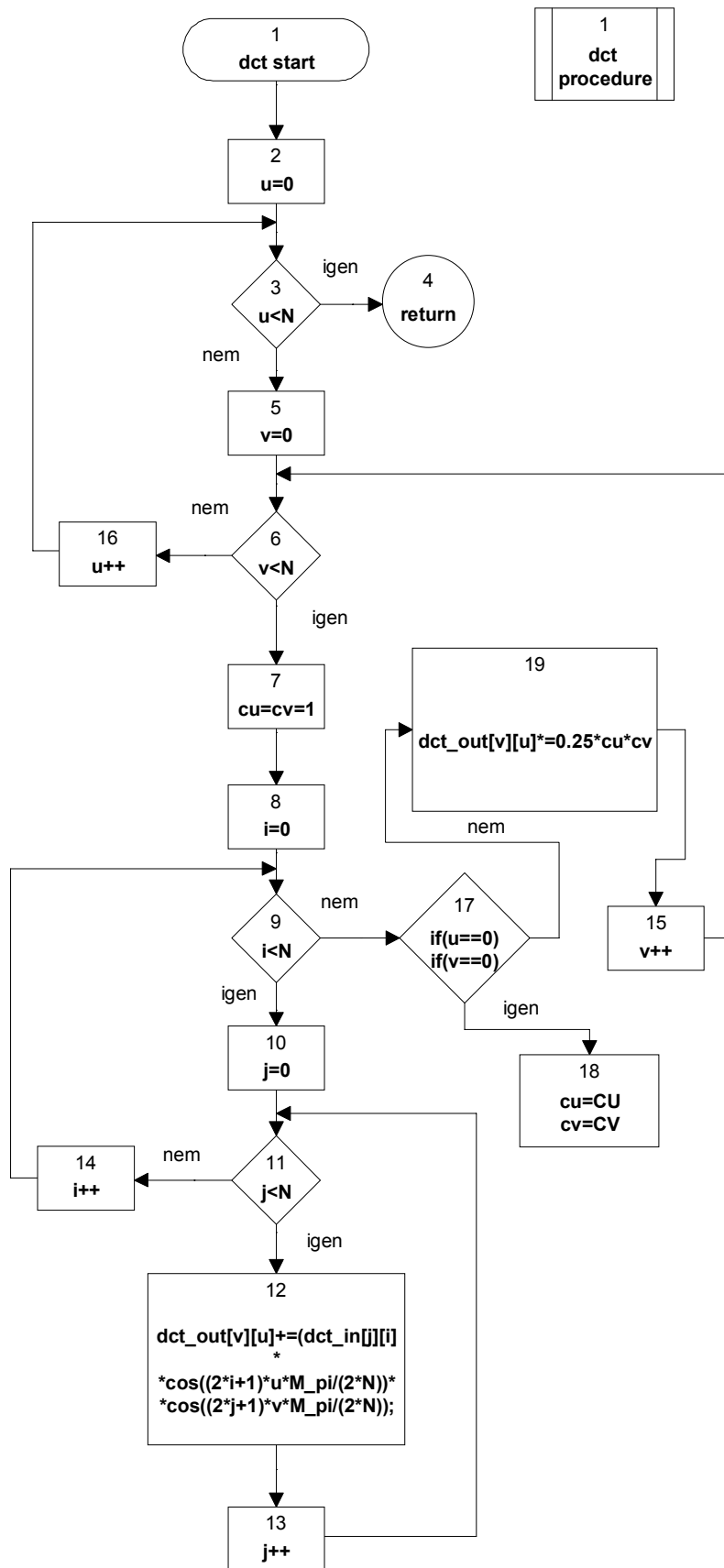
$$C_{(v)} = \frac{1}{\sqrt{2}}, \quad \text{ha } v = 0$$

$$C_{(v)} = 1, \quad \text{ha } v > 0$$

$f_{(y,x)}$ = a kétdimenziós mintavételezett értékek

$F_{(v,u)}$ = pedig a kétdimenziós DCT transzformáltak

A rutin folyamatábrája a következőképpen fog kinézni:



1
dct procedure