

# DOKTORI ÉRTEKEZÉS

## ÚJ RÁCSKVANTÁLÓ ALGORITMUSOK ÉS ALKALMAZÁSAIK A VIDEÓ- ÉS A BESZÉDKÓDOLÁSBAN

Készítette:  
Lois László

Témavezető:  
Dr. Pálinszki Antal

Budapesti Műszaki és Gazdaságtudományi Egyetem

Híradástechnikai Tanszék

Budapest

2003

# Bevezetés

Az elmúlt évtizedben a mozgókép, állókép, hang és beszéd átvitelére és tárolására kidolgozott kódolási algoritmusok mind az alkalmazott számítástechnikai megoldásokban, mind pedig a kutatások terén látványos és fontos szerepet töltek be és töltenek be ma is.

A kezdeti digitális rendszerek fő eleme az analóg jel mintavételezése és skalár kvantálása volt, ez az idők folyamán egyre több új, a hatékonyságot növelő algoritmussal egészült ki. Az algoritmusok egy része már más tudományág által ismert és használt megoldás volt, mind például a lineáris predikció vagy lineáris transzformáció, de a tömörítés fontosságának köszönhetően több új algoritmus is létrejött, mind például a spektrális burkoló különböző kódolási módjai. Ezen felül rengeteg új ismeret és alkalmazás is született a régi módszerekkel kapcsolatosan, így jött létre például a lineáris predikciós kódolás, a transzformációs kódolás, a rácskvantálás alapú kódolás stb.

A kezdeti rendszerek skalár kvantáláson alapultak mind a kép, mind pedig a hang és a beszéd esetében, a kvantálás hatékonyságát pedig kezdetben predikcióval, lineáris transzformációval növelték, és ahol ezt az alkalmazás lehetővé tette, a tömörítési arányt gyakoriságfüggő kódolással javították. Az egyre több ismeret és egyre nagyobb processzorteljesítmény eredményeképpen egyre bonyolultabb algoritmusokat is kidolgoznak és megvalósítanak, és ez nemcsak az elméleti kutatások irodalmában, hanem a gyakorlatban alkalmazott szabványokban is testet ölt. A bonyolultság ára azonban a magasabb komplexitás, és így az implementáció költségesebb, ezért a bonyolultabb algoritmusoknál a komplexitás csökkentése is fontos szempont lett.

A már kidolgozott vektorkvantálás elméletére alapozva, illetve a processzorok számítási és memóriakapacitásának rohamos növekedésének köszönhetően az 1990-es években jelentek meg nagy számban olyan rendszerek, ahol a skalár kvantálást vektorkvantálás váltotta fel, a kódolási hatékonyság jelentős növekedésével. A vektorkvantálás szinte minden esetben képes átvenni a skalár kvantálás szerepét, és ezen felül olyan lehetőségeket is nyújt, amely a skalár kvantálás esetében nem létezhet: blokk kódolás, mintánkénti bitek száma törtszám is lehet, hatékonyabb torzításkritérium stb. A hibrid skalár kvantálók helyén többnyire alkalmazható a vektorkvantálás is (spektrális kódolás vektorkvantálással, prediktív vektorkvantálás stb.) az eredeti rendszernél hatékonyabb kódolást nyújtva, de a komplexitást jelentősen megnövelve. A vektorkvantálás esetében mind a legközelebbi kódszó megtalálása (kódszó index megtalálása), mind a kódkönyvtár tárolása nagy komplexitást jelent, ezen több módon lehet javítani, néha a hatékonyság rovására. A komplexitás csökkentésének egyik jellegzetes eszköze rács-vektorkvantálás, ahol azonban a kódkönyvtárban lévő nagyfokú szimmetria a specialitás rovására is mehet, emiatt a rácskvantálást gyakran ki kell egészíteni különböző megoldásokkal (pl. transzformáció, entrópiakódolás), amelyekkel a kódolás hatékonysága javítható.

## Célkitűzések

Doktori munkám célja az volt, hogy a többdimenziós jelek kvantálásában, ezen belül is elsősorban az eredetileg térkitöltésre használt rácsok [1] vektorkvantálóként [2] való felhasználásával dolgozzak ki új algoritmusokat a digitális jeltömörítés területén. A rácskvantálás [3] témakörén belül a kódkönyvtárban való indexelés és az ehhez kapcsolódó elemszámítás terén dolgoztam ki új algoritmusokat a skálázott és bináris vektorokkal eltolt  $D_k$  alap-rácsok unióira, amelyek a legtöbbször alkalmazott rácsok a rácskvantálás esetében.

Az irodalomból már eddig is ismertek voltak a skálázott  $D_k$  alap-rács és  $N$  darab  $K$  dimenziós  $\underline{b}_i$  bináris kódszó mellékalmazainak unióiból álló  $(\bigcup_{i=0}^{N-1} (\underline{b}_i + 2 \cdot D_k))$  alakú) rácsokra kódoló, indexelő és elemszámító algoritmusok [3][4][5][6][7][8]. A  $D_k$  rács azon  $K$  dimenziós vektorok halmaza, ahol az együtthatók egészek és az együtthatók összege páros, valamint  $\underline{b}_i$  valamilyen  $N$  elemű bináris kód egyik kódszava, ilyen bináris kód például a 16 dimenziós Barnes-Wall rács [1] esetén az 16 dimenziós elsőrendű Reed-Muller kód. Ezek az algoritmusok nem, vagy csak korlátozott mértékben alkalmasak arra, hogy olyan vektorokat is kódoljanak, ahol bizonyos koordináták ki vannak zárva.

Céлом volt olyan új indexelési és elemszámítási algoritmusok konstruálása, amelyek képesek olyan korlátozásokat is kezelni, mind a vektorok koordinátaiban lévő felülről és/vagy alulról korlátos értékkészlet, ide értve a csak nemnegatív vagy csak pozitív értékeket. Ezekben túlmenően célként szerepelt a 0 értékek kizárására is képes elemszámítási és indexelési algoritmus kidolgozása, amivel olyan források kódolását lehet hatékony hibrid rácskvantálással megvalósítani, ahol skalár kvantálás esetén nagyon sok 0 és kevés nem nulla érték a jellemző, mint például egy MPEG-1 videó AC együtthatói. Ahhoz, hogy nem-0 együtthatójú kódvektorokra lehessen dönteni, új kvantáló algoritmust is ki kellett fejleszteni, amely képes a legközelebbi ilyen tulajdonságú rácsvektor megtalálására is. Ezeket az eredményeket egy elméleti jellegű téziscsoport, az 1. téziscsoport tartalmazza.

Az előbb említett új algoritmusokkal kapcsolatos követelmények gyakorlati alkalmazásokból származnak, ilyen gyakorlati alkalmazás például az MPEG-1 videó AC együtthatóinak újrakvantálása. Az eddigi tudományos eredmények az MPEG-1 videó AC együtthatóinak kvantálására, illetve újrakvantálására [9][10] skalár kvantáláson és a 0 futamok hatékony kódolásán alapultak, és a sok nulla közeli AC együttható miatt a vektorkvantálás ezen a területen eddig nem bizonyult eredményesnek. Ennél a feladatnál a céлом az volt, hogy az általam kidolgozott, a koordináta korlátokat figyelembe venni és 0 értékeket kizárni képes rácskvantálót alkalmazni tudjam MPEG-1 videó AC együtthatóinak újrakvantálása az eddig ismert tudományos

eredményeknél nagyobb hatékonysággal, vagyis azonos bitsebességen kisebb torzítást, illetve azonos torzítás mellett alacsonyabb bitsebességet elérve.

Mivel az általam kidolgozott algoritmusok többféle korlátozást is képesek kezelni, ezért meg kellett határozni azt is, hogy mely korlátozásokat célszerű figyelembe venni, illetve ezt milyen módon kell megtenni. A koordinátákban lévő korlátos értékészlet figyelembe vételének alkalmazása esetén meg kellett határozni, hogy lehetséges-e ezt az értékészletet a priori meghatározni, illetve el kell-e küldeni a dekódernek ezt az információt a sáv szélesség rovására, és ez utóbbi esetben a sáv szélesség veszteséget képes-e az új algoritmus által biztosított nyereség ellensúlyozni. Hasonlóan meg kellett határozni azt a döntéshívő függvényt is, amely megmondja, hogy mely együtthatók fontosak, illetve nem fontosak (nullának). A fontos együtthatók esetében alkalmazzuk a rácskvantálást, ekkor azt is meg kellett határozni, hogy a 0 értékekre való kvantálást engedélyezzük-e, illetve ezt csak az indexelésnél vegyük-e figyelembe. Meg kellett továbbá vizsgálni az előbb említett döntéshívő függvény és a rácskvantálás paramétereinek közötti kapcsolatot is. A feladat megoldására egy többszintű rácskvantálót konstruáltam, amelyhez megadtam a jelentőséget meghatározó döntéshívő függvényt, amely kódolását fontossági térkép kódolással oldottam meg (az MPEG-1-ben futamhossz kódolás helyett), valamint meghatároztam az indexelő algoritmust, továbbá entrópia kódolással is kiegészítettem a kódolót. Ezeket az eredményeket a 2. téziscsoport tartalmazza.

E mellett célom volt a rácskvantálás transzformációs kódolóban is alkalmazni a beszédjel spektrális kódolására használt LSF vektorok [2][12] kódolására. Az irodalomból már ismert volt a prediktív többléptékű rácskvantálás algoritmus, amely a súlyozott négyzetes hiba távolságkritériumot alkalmazva hatékony LSF kvantálást tett lehetővé [13]. Célom volt ennek a sémának a továbbfejlesztése, elsősorban a lineáris transzformációs kódolás bevezetésével és a rácskvantálás hatékonyságának növelésével. Feladatul tűztem ki a bitsebesség vezérlés és a bitkiosztás hatékonyabbá tételét, ezt általam kidolgozott pakolt indexelésnek elnevezett algoritmussal oldottam meg. A lineáris transzformációhoz való jobb illeszkedést koordinátánkénti zsugorítással oldottam meg, amely szintén új elem az LSF rácskvantálása terén. Ezeket az eredményeket a 3. téziscsoport tartalmazza.

Az általam kidolgozott új módszerek mindegyikét kipróbáltam teszt adatokon is, és összehasonlítottam az eddig ismert eredményekkel. Az összehasonlítás eredményeképpen az esetek döntő többségében jelentős javulás figyelhető meg.

# Új tudományos eredmények

## 1. Téziscsoport: Új indexelési és elemszámítási algoritmusokat dolgoztam ki rácskvantálás céljára [K2][K5]

Az új elemszámítási, kódolási és indexelési algoritmus a skálázott és bináris vektorokkal eltolt  $D_k$  alap-rácsok unióiból álló rácsokra az alábbi feltételeket is figyelembe képes venni:

- Képes kezelni a koordináták amplitúdó korlátait (például ha a vektor olyan nagy sugarú piramison\* vagy gömbön\*\* van rajta, amelyben előfordulhatnak olyan vektorok is, amelyek bizonyos koordinátákban lévő együtthatói nagyobbak az egyébként lehetséges értéknél).
- Képes kizárni a rácsból azokat a vektorokat, amelyek valamelyik komponense 0 (például ha már eleve olyan értékeket válogatott össze, amelyek egy küszöbszint felett vannak, és azt szeretném, hogy ezek a vektorkomponensek ne legyenek 0 értékűek a kvantálás után sem).

### 1.1. Tézis: Kvantáló eljárást konstruáltam a legközelebbi nem-nulla komponensű rácspontra

A  $D_K$  és az  $\bigcup_{i=0}^{N-1} (\underline{b}_i + 2 \cdot D_K)$  alakú rácsokra Conway és Sloane dolgozta ki a kvantálási algoritmusokat, vagyis a legközelebbi rácspontot, illetve több legközelebbi esetén az egyik legközelebbi rácspontot megtaláló algoritmust [4]. Ezeket az algoritmusokat módosítottam úgy, hogy a nem-nulla együtthatójú rácspontok közül keresse meg az algoritmus a legközelebbi rácsvektort.

A  $D_K$  rács esetén a Conway-Sloane kvantálási algoritmust azon a ponton kellett módosítani, hogy minden olyan koordinátában, ahol a *páros szomszéd* értéke 0 lenne, a 0 helyett a 2 vagy -2 értéket vesszük számításba a következő módon:

- ha az együttható 0-nál nagyobb, akkor 2 lesz a páros szomszéd értéke.
- ha az együttható 0-nál kisebb, akkor -2 lesz a páros szomszéd értéke

Hasonló módon kizárható bármely más érték, illetve értékek is, így páros/páratlan érték kizárása esetén értelemszerűen a legközelebbi érvényes páros/páratlan szomszédot kell megkeresni.

Megmutattam továbbá, hogy ahhoz, hogy a nem-0 együtthatójú vektorra való kvantálás megvalósítható legyen az  $\bigcup_{i=0}^{N-1} (\underline{b}_i + 2 \cdot D_K)$  alakú rácsokra, Conway és Sloane erre a rácsra vonatkozó algoritmusát az alábbi két ponton kell módosítani az  $i$ -dik bináris vektor esetén:

---

\* piramis (hiperpiramis): azonos  $L_1$  normájú vektorokból álló részhalmaz

\*\* gömb (hipergömb): azonos  $L_2$  normájú vektorokból álló részhalmaz

- Meg kell határozni, hogy  $\underline{b}_i$  mely koordinátaiban van 0, csak ezekben a koordinátákban kell kizárni a 0 koordinátákat.
- A  $D_K$  rácsra vonatkozó kvantálási algoritmust úgy kell módosítani, hogy csak azokban a koordinátákban zárja ki a 0 értéket, ahol a  $\underline{b}_j$  vektor megfelelő koordinátájában 0 van.

## 1.2. Tézis: Elemszámító algoritmust dolgoztam ki a hiperpiramison lévő vektorok számának meghatározására a véges koordinátájú kockarácsban

A véges koordinátájú kockarács ebben az esetben azt jelenti, hogy a kódkönyvtár vektoraira a következő megszorítást tesszük: minden koordináta egy adott intervallumba esik, ahol a minimum negatív, a maximum pedig pozitív.

Jelölje  $0 \leq i_{a,j}$  a legkisebb, illetve  $0 \leq i_{f,j}$  a legnagyobb érték amplitúdóját a kockarács  $j$ -dik koordinátájában, vagyis  $-i_{a,j} \leq c_j \leq i_{f,j}$ . Jelölje  $N(d,m)$  a  $d$  dimenziós  $m$  sugarú rácsvektorok számát a  $Z_d$  kockarácson, ahol  $d$  és  $m$  egész, valamint  $d > 0$ . Ekkor meghatároztam, hogy tetszőleges  $d > 0$  érték esetén  $N(d,m)$  kiszámítható az alábbi rekurzióval

$$N(d,m) = \sum_{i=-\min\{m,i_a\}}^{\min\{m,i_f\}} N(d-1, m-|i|) \quad (1)$$

A rekurzió mellett megadtam egy rövidebb formát is, ez a következő:

$$N(d,0) = 1$$

$$N(d,m) = \left. \begin{cases} 0 & \text{ha } ((d=1) \text{ és } (\max\{i_{a,1}, i_{f,1}\} < m)) \text{ vagy } (m < 0) \\ 1 & \text{ha } d=1 \text{ és } \min\{i_{a,1}, i_{f,1}\} < m \leq \max\{i_{a,1}, i_{f,1}\} \\ 2 & \text{ha } d=1 \text{ és } m \leq \min\{i_{a,1}, i_{f,1}\} \\ N(d, m-1) + N(d-1, m) + N(d-1, m-1) - \\ - N(d-1, m-1-i_{a,d}) - N(d-1, m-1-i_{f,d}) & \text{ha } d > 1 \end{cases} \right\} \quad (2)$$

Megadtam, hogy a piramison lévő vektorok száma véges nemnegatív koordinátájú kockarácsban:

$$N(d,0) = 1$$

$$N(d,m) = \left. \begin{cases} 0 & \text{ha } ((d=1) \text{ és } (i_{f,1} < m)) \text{ vagy } (m < 0) \\ 1 & \text{ha } d=1 \text{ és } m \leq i_{f,1} \\ N(d, m-1) + N(d-1, m) - N(d-1, m-1-i_{f,d}) & \text{ha } d > 1 \end{cases} \right\} \quad (3)$$

Megmutattam, hogy a piramison lévő vektorok száma nem-0 koordinátájú kockarácsban:

$$N(d,m) = \left. \begin{cases} 0 & \text{ha } m < 0 \text{ vagy } d=1 \text{ és } \max\{i_{a,1}, i_{f,1}\} < m \\ 1 & \text{ha } d=1 \text{ és } \min\{i_{a,1}, i_{f,1}\} < m \leq \max\{i_{a,1}, i_{f,1}\} \\ 2 & \text{ha } d=1 \text{ és } m \leq \min\{i_{a,1}, i_{f,1}\} \\ N(d, m-1) + 2N(d-1, m-1) - \\ - N(d-1, m-1-i_{a,d}) - N(d-1, m-1-i_{f,d}) & \text{ha } d > 1 \end{cases} \right\} \quad (4)$$

Megmutattam továbbá, hogy ha  $N_n(d,m)$  jelölni a nemnegatív koordinátájú pontok számát a  $d$  dimenziós,  $m$  sugarú piramison, ahol a koordináta-tartomány a  $j$ -dik koordinátában  $[-i_{a,j}+1, i_{f,j}-1]$ ,

ráadásul a koordinátalimit szimmetrikus (azaz  $i_{a,j}=i_{f,j}$ ), akkor a piramison lévő vektorok száma nem-0 koordinátájú kockarácsban

$$N(d, m) = 2^d \cdot N_n(d, m - d) \quad (5)$$

Az  $N_n(d, m)$  számításához a (3) egyenlőséget kell alkalmazni.

### 1.3.Tézis: Új indexelési algoritmust készítettem az $\bigcup_{i=0}^{N-1} (\underline{b}_i + 2 \cdot D_K)$ rács egy piramisára

Az általam kidolgozott kódolás az előbbiek alapján a következő lépésekből áll:

- *Előfeldolgozás:* vegyük a  $\underline{c}$  bemeneti vektort, határozzuk meg az  $r_b$  mellékalmazt megadó bináris vektort és a  $\underline{c}$  vektor sugarát,  $r(\underline{c})=m$ -et.
- *Hasítás:* válasszuk szét a  $\underline{c}$ -t két vektorra attól függően, hogy  $r_b$  bináris vektorban a megfelelő koordináta 0 vagy 1. Legyen a két vektor rendre  $\underline{c}_0$  és  $\underline{c}_1$ , dimenziójuk  $K_0$  és  $K_1$ . Ekkor nyilván

$$r(\underline{c}_0) + r(\underline{c}_1) = r(\underline{c}) = m \quad (6)$$

- *Páros komponensek ( $\underline{c}_0$ ) átalakítása:* alakítsuk át  $\underline{c}_0$ -t koordinátánként a következőképpen:

$$\underline{u}_{0,i} = \frac{\underline{c}_{0,i}}{2}, \quad \text{vagyis a mindig 0 értékű, és ezért redundáns LSB-t eltüntettük} \quad (7)$$

- *Páratlan komponensek ( $\underline{c}_1$ ) átalakítása:* jelölje  $\underline{s}$  a  $\underline{c}_1$  minden koordinátájára az előjelbiteket, így  $\dim(\underline{s})=K_1$ . Alakítsuk át  $\underline{c}_{1,i}$ -t úgy, hogy eltüntetjük a mindig 1 értékű, így redundáns LSB-t:

$$\underline{u}_{1,i} = \frac{(-1)^{s_i} \cdot \underline{c}_{1,i} - 1}{2}. \quad (8)$$

Látható, hogy az  $\underline{s}$  előjelbitekből,  $\underline{u}_0$  és  $\underline{u}_1$  vektorokból egyértelműen előállítható  $\underline{c}_0$  és  $\underline{c}_1$ .

Bebizonyította továbbá, hogy egy előjelbit redundáns. A kapott vektorok sugarára:

$$r_0 = r(\underline{u}_0) = \frac{r(\underline{c}_0)}{2} \quad r_1 = r(\underline{u}_1) = \frac{r(\underline{c}_1) - K_1}{2} \quad r_0 + r_1 = \frac{m - K_1}{2} \quad (9)$$

A bemeneti  $\underline{c}$  vektor páros és páratlan komponensekre való felosztását az 1. ábra szemlélteti.

Bináris vektor ( $r_b$ )	1	1	0	0	1	1	0	0
Input vektor ( $\underline{c}$ )	3	-1	2	0	-3	1	4	-2
A szétválasztott vektorok ( $\underline{c}_0$ és $\underline{c}_1$ )	2	0	4	-2	3	-1	-3	1
Előjel vektor (csak $\underline{c}_1$ esetében)					+	-	-	+
Kódolandó vektorok ( $\underline{u}_0$ és $\underline{u}_1$ )	1	0	2	-1	1	0	1	0

1. ábra Piramis alapú indexelés: a (7) és (8) lépések szemléltetése

- *A rész-kódok indexelése:*  $\underline{u}_0$  vektort az általános, és  $\underline{u}_1$  vektort pedig - amely koordinátái nemnegatív egészek - a nemnegatív koordinátájú indexelési algoritmussal kódoljuk:

$$\begin{aligned} index_0 &= \text{általános\_index}(\underline{u}_0) \\ index_1 &= \text{nemnegatív\_index}(\underline{u}_1) \end{aligned}$$

- **Multiplexálás:** az  $\underline{u}_0$  vektor és  $\underline{u}_1$  vektor indexeit, valamint az  $\underline{s}$  előjelbiteket az utolsó bit kivételével multiplexáljuk össze az alábbi módon:

$$a) \quad kód_0 = \sum_{i=0}^{r_0-1} N_{\text{általános}}(K_0, i) \cdot N_{\text{nemnegatív}}\left(K_1, \frac{m-K_1}{2} - i\right) \quad (10)$$

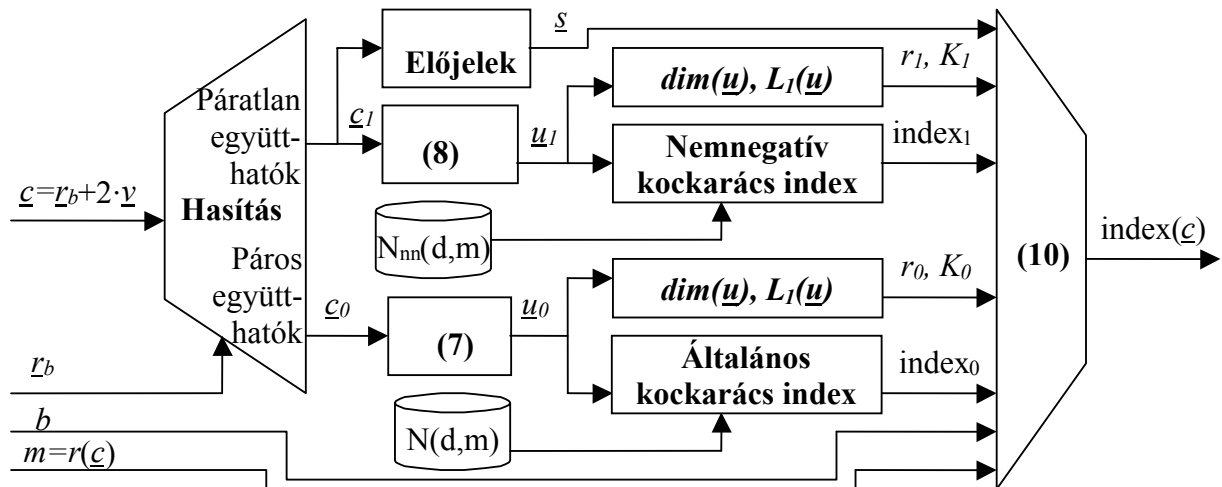
$$b) \quad kód_1 = kód_0 + index_0 + N_{\text{általános}}(K_0, r_0) \cdot index_1$$

$$c) \quad kód_2 = 2^{K_1-1} \cdot kód_1 + (s \& (2^{K_1-1} - 1))$$

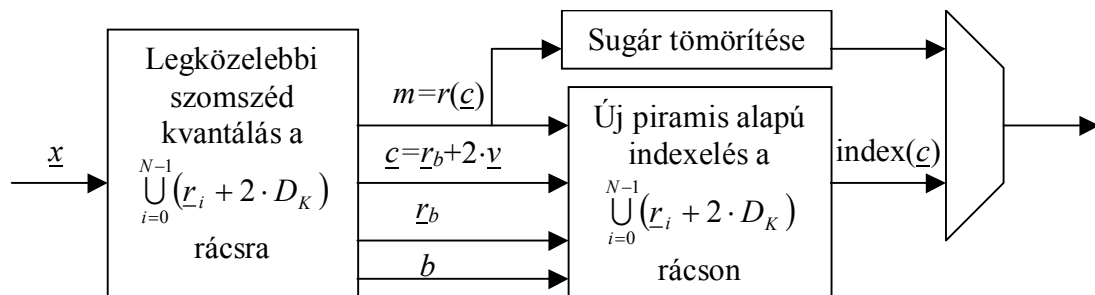
$$d) \quad index = kód_2 + \sum_{a=0}^{b-1} N_{r_a}(m)$$

ahol  $N_r(m)$  az  $r$  bináris kódszóhoz tartozó pontok száma az  $m$  sugarú piramison.

Az általam kidolgozott új indexelési algoritmus blokkdiagramját a 2. ábra, míg az algoritmus rácskvantálóba való beillesztését a 3. ábra mutatja be.



**2. ábra** Új piramis alapú indexelési algoritmus felépítése;  $N(d,m)$  és  $N_{nn}(d,m)$  jelöli a kockarácson való indexeléshez szükséges táblázatokat



**3. ábra** Új piramis alapú kvantálási és indexelési algoritmusokból felépített rácskvantáló

A kódoláshoz megadtam a dekódolást is, amely a fenti pontok inverze, ami triviális a multiplexálás kivételével. A multiplexálás inverzének részletes kifejtésével a dekódolást a következő módon határoztam meg:

- a) A kapott indexből keressük meg  $b$ -t a következő módon:

$$\text{for}(b=0; \sum_{a=0}^b N_{r_a}(m) \leq index; ++b);$$

- b) Ha megvan  $b$ , akkor ismert  $r_b$ ,  $K_0$  és  $K_1$ , valamint a hasítás, és  $kód_2$  is számítható:



$$kód_2 = index - \sum_{a=0}^{b-1} N_{r_a}(m)$$

c) Az utolsó előjelbitet kivéve megkapjuk az előjeleket, ezeket eltávolítva kapjuk  $kód_1$ -et:

$$s = kód_2 \& (2^{K_1-1} - 1) \quad \text{valamint} \quad kód_1 = \left\lfloor \frac{kód_2}{2^{K_1-1}} \right\rfloor$$

d) Keressük meg  $r_0$ -t, és  $r_0$  alapján  $r_1$  és  $kód_0$ -t is meghatározható:

$$for(r_0=0; \sum_{i=0}^{r_0} N_{\text{általános}}(K_0, i) \cdot N_{\text{nemnegatív}}\left(K_1, \frac{m-K_1}{2} - i\right) \leq index; ++r_0);$$

$$kód_0 = \sum_{i=0}^{r_0-1} N_{\text{általános}}(K_0, i) \cdot N_{\text{nemnegatív}}\left(K_1, \frac{m-K_1}{2} - i\right)$$

e)  $index_1 = \frac{kód_1 - kód_0}{N_{\text{általános}}(K_0, r_0)} \Rightarrow \underline{u}_1 = \text{nemnegatív\_index\_dekóder}(index_1, r_1)$

$$index_0 = (kód_1 - kód_0) \bmod N_{\text{általános}}(K_0, r_0) \Rightarrow \underline{u}_0 = \text{általános\_index\_dekóder}(index_0, r_0)$$

f) Végrehajtjuk  $\underline{u}_0$  és  $\underline{u}_1$  vektorokkal a hasítás inverzét úgy, mintha a hiányzó előjelbit pozitív lenne.

Ha a kapott vektor sugara nem  $m$ , akkor korrigálunk az előjelbit koordinátájában a negatív előjelnek megfelelően.

A halmazok uniójából álló rács indexelését az ennél egyszerűbb, kockarácsban történő indexelésekre vezettem vissza. A kockarács indexelési algoritmusokat a (7) és (8) alapján az  $\underline{u}_0$  vektorra az általános, az  $\underline{u}_1$  vektorra pedig nemnegatív kockarácsban kell megvalósítani.

#### 1.4. Tézis: Új elemszámítási algoritmust adtam az $\bigcup_{i=0}^{N-1} (\underline{b}_i + 2 \cdot D_K)$ rács egy piramisára

Az 1.3 tézisben bemutatott új indexelési algoritmus alapján az  $\bigcup_{i=0}^{N-1} (\underline{b}_i + 2 \cdot D_K)$  alakú rácsokra készíthetem elemszámítási algoritmus is, mégpedig arra az ötletre alapozva, hogy itt csak azt kell megszámlolni, hogy hányféle  $(\underline{u}_0, \underline{u}_1, \underline{s})$  hármas alakulhat ki.

A számítást rögtön koordinátalimittel együtt végeztem el (ha nem szükséges a koordinátalimit figyelembe vétele, akkor a limitek helyére egy nagyon nagy pozitív számot kell érteni) – ez egy új problémát is be fog vezetni, ami az 1.3 tézisben a transzformációs lépésben az abszolút értékkel kapcsolatos.

Jelölje  $N_{r,s}(m)$  az  $r$  bináris kódszóhoz tartozó pontok közül azon pontok számát, amelyeknél a páratlan koordináták előjelbitjei megegyeznek  $s$ -sel. Ezzel a jelöléssel

$$N(m) = \sum_{i=0}^{N-1} N_{r_i}(m) \quad \text{és} \quad N_r(m) = \sum_{s=0}^{2^{K_1(r)-1}-1} N_{r,s}(m) \quad (11)$$

Megmutattam, hogy az  $N_{r,s}(m)$  értékének meghatározásához az alábbi lépéseket kell megtenni:

1) Az  $r$  bináris kódszó alapján meghatározzuk  $K_0$ -t és  $K_1$ -et. Az egy redundáns bit miatt:

$$\dim(\underline{s}) = K_1 - 1.$$

2) *Transzformálás a kockarácsra*: a  $\underline{c}$  vektorra vonatkozó koordinátalimiteket át kell számítani a  $\underline{c}_0$ , illetve a  $\underline{c}_1$  vektorokhoz a (7) és (8) transzformációs összefüggést felhasználva. Ha nincs 0 koordináta, akkor  $\underline{u}_0$  vektorban sincs (7) alapján, viszont (8)-ből látható, hogy ettől még  $\underline{u}_1$  vektorban lehet 0.

3) *Rész-kódok számossága*: (9) szerint  $\underline{u}_0$  és  $\underline{u}_1$  sugara függ egymástól, és mindkettő a  $[0, \frac{m-K_1}{2}]$  intervallumból vehet fel egész értéket, és ha az egyik sugár  $i$ , akkor a másik

$\frac{m-K_1}{2} - i$ . Ezek szerint nagyon hasonlóan (10.a)-hoz:

$$N_{\underline{r},s}(m) = \sum_{i=0}^{\frac{m-K_1}{2}} N_{\text{általános}}(K_0, i) \cdot N_{\text{nemnegatív}}\left(K_1, \frac{m-K_1}{2} - i\right) \quad (12)$$

Így (11) és (12) felhasználásával számítható a rács egy piramisának számossága:

$$N(m) = \sum_{\underline{r} \in \{\underline{r}_0, \underline{r}_1, \dots, \underline{r}_{31}\}} \sum_{s=0}^{2^{K_1(\underline{r})}-1} \sum_{i=0}^{\frac{m-K_1(\underline{r})}{2}} N_{\text{általános}}(K_0(\underline{r}), i) \cdot N_{\text{nemnegatív}}\left(K_1(\underline{r}), \frac{m-K_1(\underline{r})}{2} - i\right) \quad (13)$$

Megmutattam továbbá, hogy abban az esetben, ha nincs koordinátalimit, vagy pedig a koordinátalimit szimmetrikus az  $\underline{u}_1$  vektorra nézve minden  $\underline{r}$  vektor esetében, akkor az előjelvektornak valójában nincs szerepe, és ekkor a fenti képlet sokkal egyszerűbb lesz:

$$N(m) = \sum_{\underline{r} \in \{\underline{r}_0, \underline{r}_1, \dots, \underline{r}_{31}\}} 2^{K_1(\underline{r})-1} \cdot \sum_{i=0}^{\frac{m-K_1(\underline{r})}{2}} N_{\text{általános}}(K_0(\underline{r}), i) \cdot N_{\text{nemnegatív}}\left(K_1(\underline{r}), \frac{m-K_1(\underline{r})}{2} - i\right) \quad (14)$$

A 0 koordináták kizárására pedig azt a megoldást találtam, hogy ez az  $\underline{u}_0$  vektorban jelent csak új feltételt, ekkor ezen vektorok számát nem az általános, hanem a nem-nulla táblázatból kell meghatározni, így a (13) az alábbiak szerint módosul:

$$N_{\text{nemnulla}}(m) = \sum_{\underline{r} \in \{\underline{r}_0, \underline{r}_1, \dots, \underline{r}_{31}\}} \sum_{s=0}^{2^{K_1(\underline{r})}-1} \sum_{i=0}^{\frac{m-K_1(\underline{r})}{2}} N_{\text{nemnulla}}(K_0(\underline{r}), i) \cdot N_{\text{nemnegatív}}\left(K_1(\underline{r}), \frac{m-K_1(\underline{r})}{2} - i\right) \quad (15)$$

és hasonlóan (14)-hez, a koordinátalimit nélküli, valamint a szimmetrikus limit esetében a nem-nulla komponensű vektorok számát az alábbiak szerint meghatároztam meg:

$$N_{\text{nemnulla}}(m) = \sum_{\underline{r} \in \{\underline{r}_0, \underline{r}_1, \dots, \underline{r}_{31}\}} 2^{K_1(\underline{r})-1} \cdot \sum_{i=0}^{\frac{m-K_1(\underline{r})}{2}} N_{\text{nemnulla}}(K_0(\underline{r}), i) \cdot N_{\text{nemnegatív}}\left(K_1(\underline{r}), \frac{m-K_1(\underline{r})}{2} - i\right) \quad (16)$$

Végezetül megemlítem azt az esetet is, amikor a koordinátalimit nem szimmetrikus valamelyik koordinátában: ebben az esetben a negatív és pozitív limit közül a nagyobb abszolút értékűt lehet alkalmazni a koordinátalimit vektorra a páratlan komponensek esetében (a (8) transzformációs lépésnek megfelelően). Ezzel a megoldással az indexek tartománya már nagyobb lesz a vektorok tényleges számánál, azonban ez értelemszerűen nem haladja meg a koordinátalimit nélküli számosságot, tehát némi nyereség még elkönnyvelhető.

## **2. Téziscsoport: Az új rácskvantáló algoritmusokat alkalmaztam MPEG-1 videó transzkódolásra [K2][K5]**

Az MPEG-1 transzkódolás esetén a cél egy tömörített MPEG-1 videó anyag újrakódolása új paraméterekkel ugyanabba a formátumba, illetve akár más formátumba is. Ez az átkódolás (transcoding) szélsőséges esetben történhet teljes dekódolással, majd egy újra elvégzett kódolással. Az átkódolására az a legcélszerűbb megoldás, hogy a dekódolást csak olyan, kódoló és dekódoló eszközkészlet szempontjából közös fázisig végezzük el, ameddig a kódolás és a dekódolás komplexitása közelítőleg egyforma. Ismert az eddigi kutatási eredményekből, hogy az MPEG-1 esetén a mozgáskompenzáció az a pont, ami sokkal nagyobb komplexitású a kódolóban, mint a dekóderben [9][10].

Ebben a dolgozatban a GOP (Group Of Pictures – képcsoport struktúra) változatlan marad úgy, ahogy eredetileg adott volt a tömörített videóban. A továbbiakban ennek a problémának az előző tézisben bemutatott új rácskvantálási algoritmusokkal való megoldását mutatom be, ahol a rácskvantálót az MPEG-1 videó frekvenciatartománybeli együtthatóinak újrakvantálására alkalmaztam.

### **2.1. Tézis: AC együtthatók vektorba rendezését fontossági térkép kódolással oldottam meg**

Ahhoz, hogy a kvantálást ne koordinátánként végezzük el skalár kvantálással, hanem vektorokat tudjunk kvantálni, nem elég a skalár kvantálót kicserélni vektorkvantálással, hanem több ponton is módosítani kellett az eredeti kódoló algoritmust. A DCT együtthatók vektorokba rendezése alapvetően kétféle módon lehetséges:

- Statikus vektorba rendezés: a kódoló képmátrixot kisebb téglalapokra osztja fel, és ezek lesznek a vektorok.
- Dinamikus vektorba rendezés: a kódoló a képpontok halmazát jelentős és nem jelentős pixelekre bontja, és a jelentős együtthatókat egy folyamba fűzi, és ezt a folyamatot kódolja vektorkvantálással. Természetesen ekkor a jelentős-nem jelentős döntést is kódolni kell még.

Megmutattam, hogy alacsony bitsebesség esetén a dinamikus felosztás esetében sokkal kevesebb a kvantálandó vektor, mint statikus felosztással, ezért a dinamikus felosztásra kerestem megoldást, erre a fontossági térkép kódolást javasoltam. A fontossági térkép a dinamikus vektorba rendezést írja le oly módon, hogy a dekódolásakor minden együtthatóra eldönthető, hogy fontos-e, és a letapogatási irány ismeretében pedig az is, hogy hányadik vektor hányadik koordinátájában szerepel.

Szimulációs módszerrel megmutattam, hogy a fontossági térkép kódoló algoritmusra a következő új módszer adja a legjobb eredményt, vagyis a legkisebb bitszámot ugyanazon kvantált együtthetők esetében:

- A kvantált együtthetők letapogatásának sorrendje blokk alapú: az Y, Cr és Cb képeket külön képkomponensként kezeljük, egy kép képkomponensen belül egy DCT pozícióban olvassuk végig az összes blokkot, és ezt mind a 64 vagy 63 pozícióra megteesszük.
- A térképet leíró szimbólumsorozat szimbólum-ABC-je 3 elemű: a nem fontos együtthetők 0 szimbólum jelzi, a fontos együtthetők esetében 2 jelzi a blokkban lévő utolsó jelentős együtthetőt (ekkor a blokk következő előfordulásakor már kihagyható a letapogatásból), 1 a nem utolsó jelentős együtthetőt.
- A 3 szimbólumból álló folyam entrópiakódolása adaptív aritmetikai kódolóval a leghatékonyabb.
- A teljesen üres blokkot a kódolt blokk minta (CBP), illetve a makroblokk típusa jelzi, ezért ezeket nem kell külön jelezni a fontossági térkép kódolásakor.

## **2.2. Tézis: Új algoritmust adtam a vektorba rendezett AC együtthetők kvantálása és indexelése a 0 együtthetők kizárásának szempontjából**

Empirikus úton megállapítottam a bitsebesség-torzítás diagramok vizsgálatával, hogy ha a jelentőség küszöböt a  $\lambda$  újrakvantálási lépésköz konstansszorosára ( $c \cdot \lambda$ ) választjuk ki, akkor a 0 együtthetők engedélyezése (a rácsvektor lehet piramis sarokpont) esetén  $c \approx 0.75..1$ , míg a 0 együtthetők kizárása esetén az optimális jelentőségküszöb nő, tipikusan  $c \approx 1.25..1.5$ .

Megállapítottam továbbá szintén a bitsebesség-torzítás diagramok vizsgálatával, hogy a nulla együtthetők kizárása kvantálásnál nem előnyös, vagyis hogy az új, nem-0 együtthetőkra való kvantáló algoritmus a fenti jelentőség küszöb megválasztásának javító hatása ellenére is rosszabb eredményt ad a 0-t is engedélyező Conway-Sloane algoritmusnál.

Megállapítottam azonban azt is, hogy nulla együtthetők kizárása indexelésnél már előnyös még olyan áron is, hogy minden vektor előtt 1 jelzőbittel kell jelezni azt, hogy a következő kódszó a 0-t kizáró vagy a 0-t is engedélyező indexelési algoritmusból származik. Szimulációs módszerrel igazoltam, hogy a jelzőbitek által okozott bitsebesség növekedést bőségesen kompenzálta az indexelésnél bekövetkező bitsebesség csökkenés.

## **2.3. Tézis: Vektorba rendezett AC együtthetők többszintű vektorkvantálással való kvantálását és indexelését az új rácskvantáló algoritmusokkal oldottam meg**

Az AC együtthetők vektorainak rácskvantálásánál megmutattam, hogy egy gyakorlati realizációban az aritmetikai pontosság adta korlát miatt többszintű rácskvantálást kell használni. A

koordinátalimit és a 0 együtthatók kizárása szempontjából a többszintű kvantálás javító szintjei alapvetően különböznek az első szinttől:

- Az első szinten kizárható a 0 érték, a javító szinten azonban nem.
- Az első szinten a koordinátánkénti limit a kvantálási lépésköztől és a bejövő jel értéktartományától függ, a javító szinten a limit meghatározható az alkalmazott rács és az előző szint kvantálási lépésközének ismeretében (tehát általában független a jeltől!) – a javító szinteken tehát mindenképpen az új koordinátalimmal való indexelést kell használni, hiszen az indexek tartománya így kisebb lesz.

A többszintű vektorkvantáláshoz egy külön  $s$  skálafaktort használtam ( $s$  pozitív egész szám), amellyel az újrakvantálási lépésközt megszorozva megkapható az adott szint újrakvantálási lépésköze ( $s \cdot \lambda$ ). Ha  $s > 1$ , akkor van további szint, ha pedig  $s = 1$ , akkor értelemszerűen nincs. Az  $s$  skálafaktort entrópiakódolóval tömörítve írtam ki a bitfolyamba.

Megállapítottam továbbá, hogy a piramis alapú indexelés (amikor egy vektorhoz a piramis sugarat és a piramis indexet két külön kódszóval kódoljuk, az előbbit entrópiakódolóval is) előnyösebb a piramis korlátozású térbeli kódolásnál (amikor a legkülső piramison és az azon belül lévő összes piramison lévő bármely rácsvektort egyetlen kódszó írja le), valamint a skalár együtthatók Huffman kódolásánál is. Szimulációs módszerrel azt tapasztaltam, hogy a piramis alapú kódolás (amely mellet a sugár kódolását entrópiakódolással végeztem el) nyeresége a másik kettő legjobbjához képest  $\lambda = 1$  esetében alig 7% volt, de  $\lambda \geq 1$ -nél már 15 % feletti, ami  $\lambda$  növekedésével fokozatosan növekedett.

A 0 együtthatók engedélyezésére, illetve kizárására, valamint a koordinátánkénti limit kihasználására 3 különböző új indexelő algoritmust vezettem be, az alábbi sorszámozás egyúttal mértéket is ad arra, hogy az új algoritmusok mekkora arányban szerepelnek (1=legkevésbé):

**1-es indexelési algoritmus:** Nem küldünk szélsőérték információt, de a többszintű kvantálás esetén a javító szinteken használjuk a szélsőértékeket. Használjuk továbbá a piramis sarok jelzését, illetve a nem-sarokpontok (belső pontok) indexelő algoritmusát.

**2-es indexelési algoritmus (szimmetrikus koordinátalimit):** 1-es algoritmus kiegészítve azzal, hogy a többszintű kvantáló első szintjén az abszolút értékek maximumát használjuk szélsőérték információként (minden cikcakk pozícióra egy szélsőérték információt kell küldeni).

**3-es indexelési algoritmus (általános koordinátalimit):** 1-es algoritmus kiegészítve azzal, hogy a többszintű kvantáló első szintjén az amplitúdók minimumát és maximumát használjuk szélsőérték információként (az elküldött szélsőérték információ a 2-es algoritmusénak a kétszerese: minden cikcakk pozícióban két érték).

Szimulációval megmutattam, hogy általában az 1-es és 2-es indexelő algoritmus a legjobb, de amikor a 2-es felülmúlja a legegyszerűbb 1-es indexelési algoritmust, akkor is érdektelen a javulás gyakorlati szempontból nézve. Megállapítottam továbbá, hogy az eddig ismert rácskvantáló indexelő algoritmusok (a dolgozatban a 4-es indexelő algoritmus) bitsebességéhez képest az 1-es indexelés csupán a bitsebesség 94%-96%-át igényli, ami jelentős javulás. Ez a javulás főleg a 0 együtthatók kizárásán alapuló indexelésnek és kismértékben a javító szinteken lévő koordinátalimit használatának köszönhető. Ebben az esetben tehát egy olyan sémát sikerült találnom, ahol mindkét új indexelő algoritmus hatékonyabbnak bizonyult az eddig ismertekhez képest.

Összefoglalva az eddigieket, a következő új kvantáló algoritmust dolgoztam ki az MPEG-1 videó AC együtthatóinak újrakvantálására:

- Kvantálás:
  - Többszintű kvantálás, egyetlen szint esetén a 0 együtthatók is engedélyezettek
  - A jelentőség küszöbnél  $c=1$
  - 16 dimenziós Barnes-Wall rács a többszintű vektorkvantáló minden szintjén
  - Minden szinthez elküldjük az alábbi információkat:
    - A piramis sugara: adaptív aritmetikai kódolóval kódolva.
    - Piramis index bitek: nincs további kódolás, egész bitre kerekítve írjuk ki.
    - Skálázó faktor (ha 1, akkor nincs további szint), adaptív aritmetikai kódolással.
    - Sarok jelzése jelzőbittel: szintén adaptív aritmetikai kódolóval kódolva.
  - Felsőbb szinteken szélsőérték használta az előző szint skálázó faktora alapján.
- Indexelési algoritmus: 1-es indexelési algoritmus, vagyis az adott piramis sugár mellett a sarok jelzőbittől függően vagy belső rácspontra, vagy általános rácspontra indexelünk.
- Fontossági térkép kódoló algoritmus: 3 szimbólum, aritmetikai kódolóval kódolva.
- Az MPEG bitfolyamból csak az AC együtthatókat vesszük ki.
- Az MPEG bitfolyamban a blokk mintája (Coded Block Pattern) változhat és makroblokkok válhatnak üressé.

Szimulációs módszerrel megmutattam, hogy az új algoritmus PSNR értékekben tipikusan 0.5dB-1dB-vel jobb kódolási minőséget ad azonos bitsebességen, ami jelentős minőségjavulás. Az új kódoló inkább a mozgáskompenzált differenciakép tömörítésében hatékony, hosszú GOP esetén a PSNR érték akár több mint 2 dB-t is javult, ami nagyon jelentős eredmény.

### **3. Téziscsoport: Új transzformációs kódolási eljárást dolgoztam ki beszédkeretek LSF paramétereinek kvantálására [F1][K3]**

A beszédjelek LSF paramétereit elsősorban kis bitsebességű kódolásban használják, a szabványokban többnyire prediktív osztott vektorkvantálással és súlyozott négyzetes hiba (WSE – Weighted Squared Error) távolságkritériummal [12-17]. Az LSF kódolók torzítását általában nem a négyzetes hiba, hanem a logaritmusos spektrális torzítás (LSD – Log Spectral Distortion) alapján minősítik, ezt a torzítást próbálják a kódolás során a súlyozott négyzetes távolságkritériummal közelíteni.

Az LSF vektorok között mind a szomszédos vektorok között, mind egy vektoron belül nagy korrelációt tapasztalható, az előbbit a prediktív kódolás, az utóbbit pedig a vektorkvantálás használja ki. A kereten belüli korreláció másik kiaknázási lehetősége a lineáris transzformáció alkalmazása, amely azonban az LSF tartományban érvényes (de a transzformált tartományban már nem) súlyozott négyzetes hiba alkalmazhatóságát nehezíti meg.

Az első altézisben a lineáris transzformáció és a súlyozott négyzetes hiba távolságkritérium együttes alkalmazására mutatok új, az eddigiéknél hatékonyabb algoritmust. A második altézis az osztott vektorkvantálásnál sokkal kisebb komplexitású prediktív többléptékű rácskvantálás (Predictive Multi-Scale Lattice Vector Quantization - PMSLVQ [12]) és a transzformációs kódolás együttes használatára ad új módszert, a harmadik altézisben pedig új, a logaritmusos spektrális torzítást az eddigiéknél jobban közelítő hibakritériumot vezetek be, amely szintézis általi analízis technikát használ. Továbbá bevezetem a PMSLVQ sémában a léptékenkénti különböző korlátot és az ehhez kapcsolódó pakolt indexelést, és általam kidolgozott újdonság a koordinátánkénti zsugorítás is.

#### **3.1. Tézis: Továbbfejlesztett sémát készítettem lineáris transzformációval transzformált LSF paraméterek súlyozott négyzetes hibakritériummal való osztott vektorkvantálására [F1][K3]**

Osztott vektorkvantálás alkalmazása esetén a kódolási komplexitás azáltal csökken, hogy a vektorok szegmensekre oszlanak, és a keretenkénti bitszám is eloszlik a szegmensek között, így kisebb kódkönyvtárakat kell alkalmazni kisebb dimenziójú vektorok kvantálására. Megmutattam a dolgozatban, hogy amennyiben a  $K$  szegmensű osztott vektorkvantálást ortogonális lineáris transzformáció is megelőzi, és WSE távolság-kritériumot alkalmazunk az eredeti (transzformáció előtti) tartományban, akkor az  $\underline{u}$  vektor transzformálása és a transzformált vektor osztott vektorkvantálása esetén az optimális  $\underline{c}_1 \in C_1, \underline{c}_2 \in C_2, \dots, \underline{c}_K \in C_K$  kódszavak nem határozhatóak meg egymástól függetlenül – ellentétben a súlyozás nélküli esettel.

A WSE használatához lineáris transzformációt alkalmazó osztott vektorkvantálás esetére új algoritmust készítettem, amely lényege, hogy – feltételezve, hogy a transzformáció utáni koordináták fontossági sorrendben állnak (pl. Karhunen-Loeve transzformáció vagy DCT) –  $c_j$  meghatározásához figyelembe kell venni az összes  $j < i$  sorszámú szegmensre a már megkapott  $c_j$  kódszavakat, és a  $j > i$  sorszámú szegmensekre pedig 0-t veszünk. Megmutattam továbbá, hogy ez az algoritmus elvileg csak szuboptimális megoldást ad, de jobb megoldást jelent, mint az eddig ismert módszerek, ahol az összes  $j < i$  sorszámú szegmensre is 0-t vesznek az  $i$ -dik szegmens kódolásakor. Szimulációs vizsgálatokkal megmutattam, hogy mind a tanító adatokra, mint a tanító adatoktól független teszt adatokra ez az eljárás WSE esetén ugyanakkora minőségjavulást eredményez a transzformáció bevezetésével, mint amekkora javulást a súlyozás nélküli esetben jelentett a transzformáció bevezetése, ellentétben az eddig ismert módszerrel, ahol WSE esetében a transzformáció bevezetése ugyan javított a minőségen, de nem akkorát, mind amekkorát a WSE nélküli esetben tapasztalhatunk.

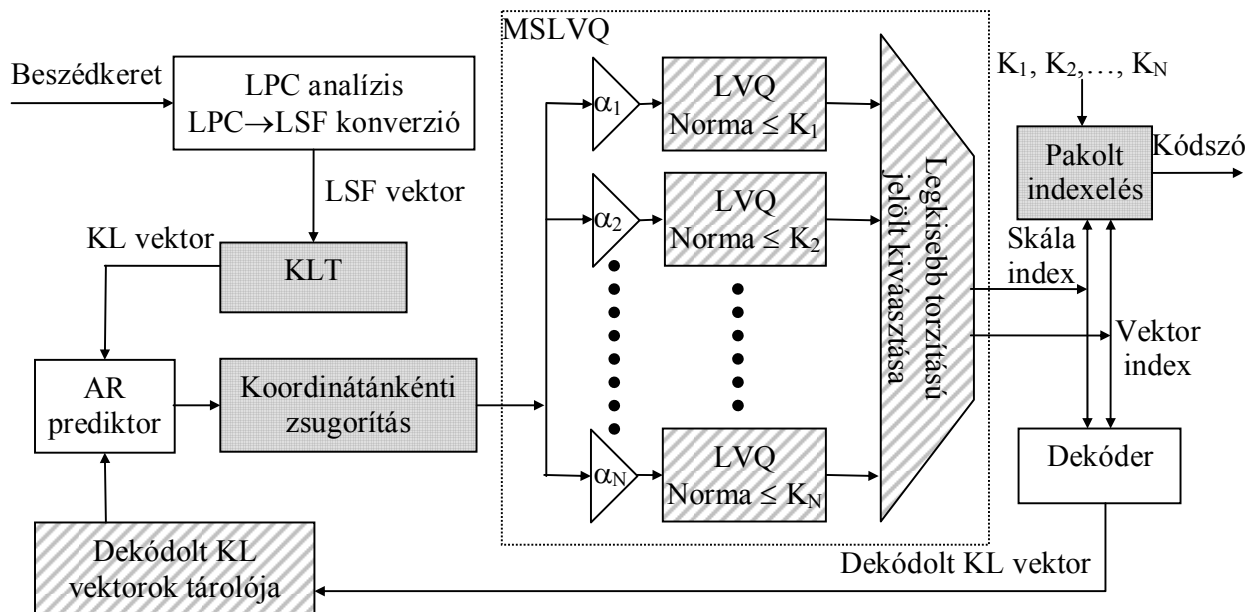
### **3.2. Tézis: PMSLVQ alapú LSF kódolási eljárást kiegészítettem lineáris transzformációval**

A lineáris transzformációt az irodalomból ismert PMSLVQ alapú LSF kódolóba alapvetően két helyre lehet beilleszteni a jelútba, ez a prediktor előtti (transzformáció-predikció sorrend), illetve a prediktor és a rácskvantáló közötti részre (predikció-transzformáció sorrend). Lineáris transzformációként a Karhunen-Loeve transzformációt (KLT) és a diszkrét koszinusz transzformációt (DCT) használtam, mind az irodalomban erre a célra szinte kizárólagosan használt két módszert.

A KL transzformációt a transzformáció-predikció sorrend esetében a tanító adatbázis 0 várható értékűvé tett LSF vektorainak empirikusan megállapított autokorrelációs mátrixából származtattam, majd pedig a kapott KLT mátrixot, illetve a DCT-t használva tanítottam be a keretek közötti 4-edrangú autoregresszív (AR) prediktort. A predikció-transzformáció sorrend esetében először az AR prediktort kellett meghatározni a tanító LSF vektorokra, ezután pedig a kvantálás nélküli predikciós hibavektorra kellett a KLT mátrixot meghatározni.

Szimulációs vizsgálatokkal megmutattam, hogy mind a tanító adatokra, mint a tanító adatoktól független teszt adatokra a KLT vagy DCT bevezetése legalább 0.1 dB átlagos LSD javulást jelent, és a transzformáció-predikció sorrend minden vizsgált esetben előnyösebb a predikció-transzformáció sorrendnél, habár a különbség gyakorlati szempontból nem lényeges (néhány század dB).





4. ábra Új KLT-PMSLVQ séma LSF vektorok transzformációs kódolására, szürkével a teljesen új részeket, szürke-fehér csíkozással a funkcionálisan módosított részeket jelöltem

### 3.3. Tézis: LSF kvantálási eljárás új lineáris transzformációs PMSLVQ-val

A lineáris transzformáció - AR predikció - többléptékű rácskvantáló sémán további változtatásokat vezettem be (4. ábra), amely további jelentős minőségjavulást jelentett LSF vektorok kódolása esetén:

- Szintézis általi analízis alapú torzításkritérium alkalmazása:  
Az általam készített rendszerben a  $D_{10}^+ = D_{10} \cup ((\frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2}) + D_{10})$  rácsot használtam, és az alábbiak szerint állapítottam meg, hogy mennyi jelöltre kelljen elvégezni a szintézist:
    - Minden léptéken kvantálunk a  $D_{10}$  rácsra a norma korlát figyelembe vételével
    - Minden léptéken elvégezzük a kvantálást az  $(\frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2}) + D_{10}$  rácsra a norma korlát figyelembe vételével
    - A  $N$  darab lépték mindegyikén elvégezzük a fenti kvantálást, amely eredménye  $2 \cdot N$  darab vektor lesz, mindegyikre ismert a négyzetes hibája  $\underline{x}$ -hez képest. Vesszük az  $A \leq 2 \cdot N$  darab legjobb vektort (egy léptékről akár mind a kettő továbbjuthat), elvégezzük rajtuk a szintézist, és a kapott  $A$  darab LSF vektor közül a választott bonyolultabb távolságkritérium szerinti legjobbra döntünk. Ezek közül a következőket használtam:
      - WSE, minden lépték mindkét jelöltjére, vagyis  $A=2 \cdot N$
      - LSD, ahol  $A$  szabadon választható a teljes tartományán, azaz  $A=2, \dots, 2 \cdot N$
- Az általam kitalált új algoritmusban az az újdonság, hogy
- minden léptékről 2 jelölt jut el a végső döntésig, nem pedig egy

- a rácskvantálásra sikerült egy olyan sémát találni, amelyben néhány jelöltet ki lehet emelni a teljes kódkönyvtárból, és ezekre akár olyan komplex torzításkritérium is alkalmazható, mint a logaritmikus spektrális torzítás.
- Koordinátánkénti zsugorító tényezők alkalmazása:

A rácskvantáló a koordinátákra nézve jelentő szimmetriával rendelkezik, más szóval nem tesz különbséget a különböző koordináták fontossága között. Mivel mind az LSF vektorok predikciós hibavektora, mind a transzformált LSF vektorok predikciós hibavektora különböző koordinátánkénti eloszlású, ezért ezt a különbséget a kódlóban a koordinátánkénti zsugorítással ellensúlyoztam. Az 1-nél nagyobb „zsugorító tényező” (szorzótényező) finomabb kvantálást, az 1-nél kisebb pedig durvább kvantálást jelent a zsugorítás nélküli esethez képest.

- Léptékenkénti különböző korlát és pakolt indexelés:

Az MSLVQ piramis vagy gömb alapú korlátozással működik. Az újdonság az általam elvégzett módosításban az, hogy itt minden léptékre megadható különböző korlát. Szimulációs úton megmutattam, hogy a legkedvezőbb az a stratégia, amikor a nagyobb korlátot a nagyobb léptékekhez rendeljük hozzá. A nagyobb korlát egyúttal több kódszót is jelent, ezért kidolgoztam az ún. pakolt indexelést. A pakolt indexelés lényege az, hogy többféle lépték különböző korlátjának megfelelően elkészíti azt az optimális index kódszót, amelyből meghatározható a lépték és a léptéken belüli rácsvektor is:

- Adott  $N$  darab lépték, az  $i$ -dik léptéken  $K_i$  jelöli a kódpontok számát, ahol  $0 \leq i < N$
- Jelölje  $S_i = \sum_{k=0}^i K_k$ , ha  $0 \leq i < N$  és legyen  $S_{-1}=0$
- Legyen a rácsvektor léptéke  $n$ , és a kódszó index  $i_{LVQ}$ , ahol  $0 \leq n < N$  és  $0 \leq i_{LVQ} < K_n$
- Ekkor a dekódernek küldendő pakolt index a következő módon kapható meg:

$$pakolt\_index(n,i) = S_{n-1} + i_{LVQ}$$

- A dekóder a kapott *pakolt\_index* alapján a következő módon kapja meg a rácsvektort:
  - Keressük meg azt a  $0 \leq n < N$  értéket, amelyre

$$S_{n-1} \leq pakolt\_index < S_n$$

- A kapott  $n$  léptékindex alapján meghatározható az  $i_{LVQ}$  kódvektor index:

$$i_{LVQ} = pakolt\_index - S_{n-1}$$

Szimulációs vizsgálatokkal megmutattam, hogy a 18..24 bit/keret bitsebesség tartományon az eddig ismert PMSLVQ eljárásához képest a javulás legalább 0.1 dB logaritmikus spektrális torzításban, ami jelentős javulást jelent. Az igen kis bitsebességen (18..19 bit/keret) a DCT-P-MSLVQ séma adta a legjobb eredményt gömb alapú korlátozással, e felett pedig a KLT-P-MSLVQ séma volt a legjobb, általában gömb alapú korlátozással, de bizonyos tartományokon a piramis alapú korlátozás adott jobb eredményt.

## Az új tudományos eredmények publikációi

### Folyóirat cikkek

- [F1] H. L. Vu, L. Lois (2000), „Efficient Distance Measure for Quantization of LSF and Its Karhunen-Loeve Transformed Parameters”, *IEEE Trans. on Speech and Audio Processing*, Vol. 8, No. 6, pp. 744-746, Nov. 2000.
- [F2] L. Lois, H.L. Vu (1999), „Spectral Coding of Speech LSF Parameters Using Karhunen-Loeve Transform”, *IEICE Trans. Fundamentals, Special Section on Information Theory And Its Applications*, Vol. E82-A, No. 10, pp. 2138-2146, October 1999.
- [F3] H. L. Vu, L. Lois (1999), „A New General Distance Measure for Quantization of LSF and Its Transformed Coefficients”, *IEICE Trans. Fundamentals, Special Section on Digital Signal Processing*, Vol. E82-A, No. 8, pp. 1493-1500, August 1999

### Konferenciakiadványban megjelent publikációk

- [K1] L. Lois (2002), „Interaktív műsorszórás alkalmazások megvalósítása a digitális videó műsorszórásban”, *Proceedings of the 10th Conference and Exhibition on Television and Audio Technologies*, Budapest, Hungary, pp. 95-104, May 2002
- [K2] L. Lois (2000), „Rácskvantálás alkalmazása a video transzkódolásban”, *Proceedings of the 9th Conference and Exhibition on Television and Audio Technologies*, Budapest, Hungary, pp. 171-181, May 2000
- [K3] H. L. Vu, L. Lois (2000), „Optimális távolságkritérium LSF paraméterek kódolására”, *Proceedings of the 9th Conference and Exhibition on Television and Audio Technologies*, Budapest, Hungary, pp. 183-191, May 2000
- [K4] L. Lois (1999), "Variable Length Coding of Transformed LSF Parameters", *Proceedings of ESCA EUROSPEECH'99 Conference*, Budapest, Hungary, Vol. 3. pp. 1455-1458, 1999
- [K5] L. Lois, S. Bozóki (1998), „Transcoding Of MPEG Video Using Lattice Vector Quantization”, *Proceedings of the IEEE International Conference on Image Processing*, Chicago, Illinois, USA, Vol. 2. pp. 341-345, October 1998
- [K6] L. Lois, H. L. Vu (1998), „Transcoding Alapú Kommunikáció a Videó- és Beszédkódolásban”, *Proceedings of the 8th Conference and Exhibition on Television and Audio Technologies*, Budapest, Hungary, pp. 32-39, May 1998.

- [K7] H. L. Vu, L. Lois (1998), „A New General Distance Measure for Quantization of LSF and Their Transformed Coefficients”, *Proceedings of IEEE Int. Conf. On Acoustics, Speech And Signal Processing*, pp. 45-48, 1998.
- [K8] H. L. Vu, L. Lois (1997), „Spectral Sensitivity of LSP Parameters and Their Transformed Coefficients”, *Proceedings of the European Conf. Speech Comm. Technology, EUROSPEECH*, Rhodes, Turkey, pp. 1251-1254, September 1997.
- [K9] H. L. Vu, L. Lois (1997), „Optimal Transformation of LSP Parameters Using Neural Network”, *Proceedings of IEEE Int. Conf. On Acoustics, Speech And Signal Processing*, pp. 1339-1342, 1997.
- [K10] L. Lois, H. L. Vu (1996), „Video and Speech Compression at Low Bit Rate”, *Proceedings of the IEEE International Conference on Multimedia Technology and Digital Telecommunication Services*, Budapest, Hungary, pp. 89-96, October 1996.

## Irodalomjegyzék

- [1] J.H. Conway, N.J.A. Sloane, "Sphere packing, Lattices and Groups", Springer Verlag, 1993
- [2] N .S. Jayant and P. Noll, "Digital Coding of Waveforms: Principles and Applications to Speech and Video." Prentice-Hall, 1984.
- [3] J.H. Conway, N.J.A. Sloane, "A fast encoding method for lattice codes and quantizers", *IEEE Trans. Inform. Theory*, pp. 820-824, Nov. 1983
- [4] J.H. Conway, N.J.A. Sloane, "Fast quantizing and decoding algorithms for lattice quantizers and codes", *IEEE Trans. on Information Theory*, pp. 227-232, 1982
- [5] T. R. Fischer, "A pyramid vector quantizer", *IEEE Trans. on Information Theory*, pp. 568-583, July 1986
- [6] P. Rault, C. Guillemot, "Lattice vector quantization with reduced or without look-up table", *SPIE Int. Conf. on Visual Communication and Image Processing*, San Jose, CA, Jan. 1998, vol. 3309, pp. 851-862
- [7] J.M. Moureaux, P. Loyer, M. Antonini, "Efficient indexing method for Zn and Dn lattice quantizers", *Proc. of IEEE Int. Conf. on Image Processing*, 1996, Vol. 3. pp. 447-450
- [8] J.M. Moureaux, P. Loyer, M. Antonini, "Low complexity indexing method for Zn and Dn lattice quantizers", *IEEE Transactions on Communications*, 1998, pp. 1602-1609
- [9] G. Keesman et al.: "Transcoding of MPEG-2 bitstreams", *Signal Processing: Image Communications*, Vol. 8, Sept. 1996, pp. 481-500.
- [10] P. A. A. Assuncao and M. Ghanbari: "Transcoding of MPEG-2 video in the frequency domain", *Proc. of IEEE Int. Conf. On Acoustics, Speech And Signal Processing*, 1997, Vol 4
- [11] F. Itakura, "Line Spectrum Representation of Linear Predictive Coefficients of Speech Signals", *Journal Acoust. BoG. America*, Vol. 57, pp. S35(A), 1975.
- [12] Vasilache, M. Vasilache, I. Tabus, "Predictive multiple-scale lattice VQ for LSF quantization", *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Phoenix, May 1999, pp. 657-660
- [13] K.K. Paliwal and B.S. Atal, "Efficient Vector Quantization of LPC Parameters at 24 bits/frame", *IEEE Trans. Speech Audio Processing*, Vol.1, No.1, pp. 3-14, January 1993.
- [14] R. Laroia, N. Phamdo, N. Farvardin, "Robust and Efficient Quantization of Speech LSP Parameters Using Structured Vector Quantizers", *Proc. of IEEE Int. Conf. On Acoustics, Speech And Signal Processing*, 1991, pp. 641-644
- [15] D. Chang, S. Ann, C. W. Lee, "A Classified Vector Quantization of LSF Parameters", *Signal Processing*, 1997, pp. 1347-1350
- [16] W.R. Gardner, B.D.Rao, "Theoretical analysis of the high-rate vector quantization of LPC parameters", *IEEE Transactions on Speech and Audio Processing*, vol. 3, pp. 367-381, 1995
- [17] M. Skoglund, J. Skoglund, "On non-linear utilization of intervector dependency in vector quantization", *Proc. IEEE Int. Conf. On Acoustics, Speech And Signal Processing*, Seattle, USA, 1998