

- bitseb. változtatás (MPEG)

eventális mx. sorozás külön számokkal

- 52. dia: bal oldal: DCT értékek

- JPEG kódolás / dekódolás

- 128: szimmetrikus legyen a jel

az : ± 1 ig utasog

az adat viszont 0...255 tartományban van (csak pozitív)

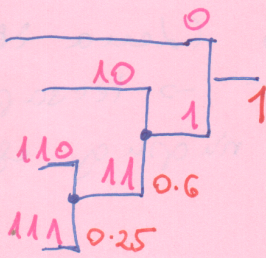
- DC-re prediktív kódolás

az: akkor jó, ha a kódolás tart. dekódolót

- csak a eventális veszteséges, a többi nem

- Huffman kód:

szimbólum:	P_i
a_1	0.4
a_2	0.35
a_3	0.2
a_4	0.05

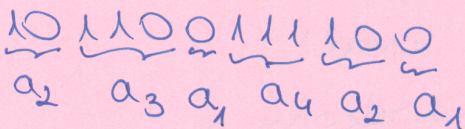


Huffman-tábla

szimbólum:	kód:
a_1	0
a_2	10
a_3	110
a_4	111

• először összerakjuk a
zet legrövideket → viszonyképeret összerakjuk ⇒
2 legrövidebb ö.f. ...

• miért jó? - mert egyszerű



- JPEG VLI tábla

- DC értéket osztályozásba sorolja

Huffman tábla a_1, a_2, \dots : osztályozás → hány biten van abszolút
VLI → amplitúdó

	syndol ₁ (Huffman) S ₁ S ₂ S ₃	syndol ₂ ampl. (VLI)
DC		
AC	S ₁ S ₂ S ₃ ↓ ↓ 4bit 4bit	ampl.

↓ ↓
 s-ε kategória

- JPEG szabvány

- eddig base line zóddat néshiz
- van progresszív is
- ha hierarchikus: alap beállításnál egyre nagyobb felbontással, de az első lép megjelenik

- JPEG hiba

- Évaatolás vissi bele
 - ↳ blokkosság
 - ↳ mosquito zaj (Lena példánál)
 - éles katar → lecsapjuk a nagyfrekv. komponensre → ringing effekt az él mellett

Mozgóép tömönítés

- plusz szabadsági f. → 100
- eszmás utáni éper hasznitana (de a haszn. rész nem u.o. van) ↓ precíz
- mozgás megkeresése

↳ optical flow

pixelekenti elmozdulást vizsgál
 ○ tömönítéshez nem túl hatékony,
 számításigényes (gradienset számol)

↳ blokk alapú mozgásbecslés

- Éper NxM-es blokkokra bontjuk
 - $D_x - D_y$ zónított a ~~re~~ rész → ebben keresem, hogy hol a legisebb az eltérés
 - valyem zötségh. min. v. max. → vizsgálunk
- keresési ablak

- költségfü - ez

• SAD (Summed Absolute Difference)

$$SAD(u, v) = \sum_{m=1}^M \sum_{n=1}^N |B_{act}(u, n) - B_{ref}(m-u, n-v)|$$

$-R_x \leq u \leq R_x$
 $-R_y \leq v \leq R_y$

• SSE (Summed Square Error)

$$SSE(u, v) = \sum_{m=1}^M \sum_{n=1}^N (B_{act}(u, n) - B_{ref}(m-u, n-v))^2$$

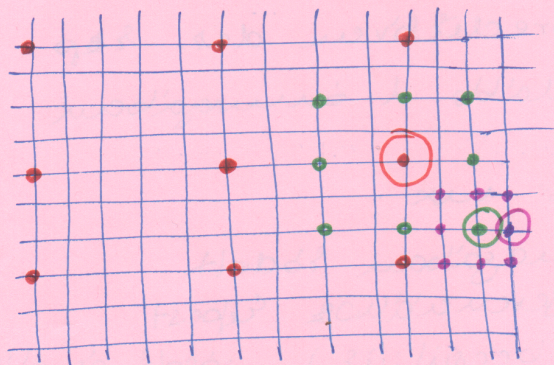
- a költségfü. nem túl nagy számításigényű, ami befolyásol az, hogy léd zerevize

a) a teljes keresési ablakban mindenkül végsőigézzük
Teljes / kimerítő keresés

b) Pixel-rekurzív keresés } zordabb eredményt használja
 c) Hierarchikus keresés } → nem használja

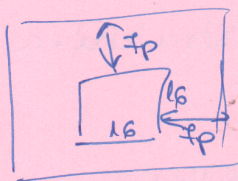
d) Logaritmus keresés (ez elterjedt)

TSS (three step search)



- 1) ± 4 ezzel eltdjuz "zinevize pontzat"
- 2) ± 2 ponttal eltdva nézzük
- 3) ± 1 ponttal → u.o.

$N = 9 + 8 + 8 = 25$ lépésben megoldást talál



a teljes keresés: ~~25~~ lépés lenne az ablakban

- MPEG 11-beu: 1) ± 1 beu nézzük ~~beu az oszt a legzisebb~~
 2) ± 2 beu nézzük ~~ban v. a zörsön lehet~~
 min: 17 lépés zell (3) → hierarchikus

- Érdemes blokk mérete

→ ha kicsi:

▷ pont. predikció

▷ nehezebb hosszabb képek

▷ mozgásvektorok & rendelkeznek

- mozgásbecslés: - vizsgálójelre makroblokkal (MB)
pikelpontosság felérés → 2x-es számításigény

MPEG szabványok

- bitsyntaxist adja meg, csak előzőt ad

- hosszú állatok elő, mihamarabb. használt, azt
te mondod meg

- pl: MPEG 1 part 3: MP3

MPEG-1

- egyszerű prediktor és transformációs érdős

- referenciák - 6 réteg

1. Szekvencia - maga a ^{videó} kép leírása

2. Képszoport (GOP)

- csak benne prediktoros der. kép

- van benne legalább 1 önmagában
érdős kép

- leírás: eventálsí u-x.

3. kép - intrazsdot: önmagában érdős

prediktor: új referenciákhoz képest

b-kép (2 irányból becsljű) - jdsolt és átlagolt

4. Szélet: Huffman tábla
skalársz def.

5. Makroblokk (16x16), 4:2:0 u-v-i struktúra
v, u: 8x8

6. Blokkok: DCT, eventálsí alapességei