

Képfeldolgozás (transzformációs kódolás, intra- és inter-predikció)  
MATLAB-al

## Videotechnika gyakorlat II.

2015

### Tartalomjegyzék

0.1. JPEG-szerű DCT alapú képtömörítés vizsgálata . . . . .	2
0.2. Változó blokkméretű DCT alapú képtömörítés vizsgálata . . . . .	2
<b>1. Intra- és inter-prediktív tömörítés demonstrációs példaprogramok</b>	<b>3</b>
1.1. Mozgáskompenzált predikció . . . . .	3
1.1.1. Mozgáskeresés, mozgásbecslés, predikciós hiba . . . . .	3
1.1.2. Mozgáskompenzált prediktív kódolás . . . . .	4
<b>2. H.264 intra-predikció demo</b>	<b>4</b>

## 0.1. JPEG-szerű DCT alapú képtömörítés vizsgálata

Vonatkozó MATLAB script:

JPEG\_blockiness.m

A kód tetszőleges képre (a világosságjel komponensre) elvégzi a  $N \times N$ -es blokk alapú DCT transzformációt, és a kapott DCT együtthatókat a JPEG kvantáló mátrixának és egy Qscale skálafaktor szorzatának megfelelő kvantálási lépcsők szerint újrakvantálja.

**Vizsgáljuk meg néhány képre a blokkosodást, illetve a PSNR értéket a blokkméret, illetve kvantálási skálafaktor függvényében !**

**Mit valósít meg a**

```
Wq =      [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 81 194 113 92;
           49 64 78 87 103 121 120 101;
           72 92 95 98 121 100 103 99];

if not (N==size(Wq,1))
    Wq=round(imresize(Wq,[N N],'nearest'));
end
```

**kódrészlet ?**

## 0.2. Változó blokkméretű DCT alapú képtömörítés vizsgálata

Vonatkozó MATLAB script:

var\_N\_block\_DCT.m

A kód nagyban hasonlít a 0.1 fejezet JPEG-szerű kódolójához. Az alkalmazott DCT transzformáció ellenben nem fix  $N \times N$ -es blokkméretű, hanem a képtartalom függvényében  $2 \times 2$ -től  $32 \times 32$ -ig terjedő blokkméretű DCT-t alkalmaz. A képtartalomfüggő blokkfelbontást (quad-tree alapú szub-blokk képzést) a MATLAB Image Processing Toolbox qtdecomp függvénye segíti. A qtdecomp tetszőleges négyzetes képmátrix homogenitását vizsgálja. Ha a képmátrixon belül a homogenitás egy beállítható küszöbértéket teljesít, akkor a négyzetes blokkon nem bontja tovább. Ha a homogenitási kritérium nem teljesül, a négyzetes képmátrixot négy egyenlő részre bontja, majd ezek homogenitását vizsgálja. A függvény rekurzívan addig bontja szub-blokkokra az eredeti képblokkot, amíg minden szub-blokk teljesíti a homogenitási kritériumot, vagy a felbontás elérte a megadott legkisebb blokkméretet. A homogenitási határérték 0 és 1 között értelmezett.

A kód a  $8 \times 8$ -as szabványos JPEG kvantáló mátrixot decimálja/interpolálja a  $2 \times 2$ -es-től  $32 \times 32$ -es blokkméretig. Minden blokkméret esetén a kapott kvantáló mátrixot megszorozza egy beállított (minden blokkméretre azonos) Qscale kvantálási skálafaktoral.

**Vizsgáljuk meg a változó blokkméretű kódolás hatását különböző képekre, különböző kvantálási skálafaktor (Qscale), és homogenitási határérték (Thresh) érték mellett. Hasonlítsuk össze a**

kapott PSNR értékeket ugyanarra a képre a 0.1 fejezet fix blokkméretű kódolójával kapott PSNR értékekkel !

## 1. Intra- és inter-prediktív tömörítés demonstrációs példaprogramok

### 1.1. Mozgáskompenzált predikció

#### 1.1.1. Mozgáskeresés, mozgásbecslés, predikciós hiba

Vonatkozó MATLAB scriptek:

```
motion_comp_pred_full_search_seq.m % Teljes kereses  
motion_comp_pred_hierch_search_seq.m % Hierarchikus kereses
```

A scriptek célja a blokk alapú teljes keresés, illetve hierarchikus (3 felbontási szintet használó) mozgáskeresést használó mozgás-kompenzált predikció vizsgálata. A scriptek segítségével megjeleníthetjük a blokkokhoz rendelt mozgásvektorokat, az egymást követő képek közötti predikció hibáját mozgáskompenzáció nélkül, és mozgáskompenzáció használata esetén.

**Feladat:** a *football-seq*, *tennis-seq*, *stefan-seq* alkönyvtárakban található képszekvenciák néhány 10 képkockát tartalmazó részletein vizsgáljuk meg a két script működését.

Hasonlítsuk össze a teljes keresés sebességét, illetve pontosságát (utóbbit a PSNR értéke alapján) a hierarchikus keresés sebességével és pontosságával egy kiválasztott szekvencia adott intervallumára. Hogyan értelmezzük ebben az esetben a PSNR értéket ?

A teljes keresés esetén választhatunk full, half, quarter - pixel pontosságú mozgásvektor meghatározások között, figyeljük meg, hogyan nő drámaian a mozgáskeresés időtartama, illetve hogyan változik a mozgáskeresés pontossága a tört-pixel pontosságú keresés hatására.

A script beállításai:

```
seq_length=10;  
filename_mask='tennis_seq/tennis';
```

A `seq_length` paraméter értéke megadja, hogy a kiválasztott szekvencia első hány egymást követő képe között végezzünk mozgáskeresést. A `filename_mask` paraméterrel a vizsgált szekvenciák közül választhatunk. Lehetséges értékei:

```
football_seq/football  
stefan_seq/stefan  
tennis_seq/tennis
```

A keresés pontossága a **PredictionType** változóval állítható, lehetséges értékei 'full', 'half' és 'quarter'.

A keresés blokkmérete az  $N$  paraméterrel állítható, a keresési mező mérete az adott blokk körül a  $W$  paraméterrel. Vizsgáljuk meg, hogy hogyan változik a keresés sebessége ha csökkentjük, illetve növejtük a  $W$  értékét. Mi a racionális felső határa a  $W$  értékének ? Figyeljük meg, hogyan változik a predikció pontossága a blokkméret,  $N$  értékének csökkentésével és növelésével.

A hierarchikus keresés esetén (`motion_comp_pred_hierch_search_seq.m`) a script beállításai hasonlóak a teljes keresés esetéhez. A  $L$  szintű keresés esetén (alap esetben  $L=3$ ) esetén első lépésben a  $2^{L-1}$ -szeresen alul-mintavételezett képen keressük a legjobb illeszkedésű referenciablokkokat, majd a talált mozgásvektorok segítségével ezt finomítjuk egy  $2^{L-2}$ -szeres alul-mintavételezésű képen, stb ... míg eljutunk az eredeti felbontású képen az utolsó keresési lépéshez.

### 1.1.2. Mozgáskompenzált prediktív kódolás

Vonatkozó script:

```
mot_pred_coding.m
```

A script segítségével a mozgás-kompenzált predikciós blokkok veszteséges DCT transzformációval történő kódolásának hatását vizsgálhatjuk. Az előző scriptekhez hasonlóan elvégezzük a mozgás-keresést, majd a reziduális blokkokat DCT transzformáljuk, és a DCT együtthatókat újrakvantáljuk. A prediktív kódolást (az előző scripekhez) hasonlóan elvégezzük mozgáskompenzáció nélkül is.

Önálló feladat: a kód értelmezésével állapítsuk meg hogy a kapott ábrák mit ábrázolnak. Mit kell módosítani a kódban, ha a következő ábrákat szeretnénk látni: eredeti képek (kódolatlan képek), mozgás-kompenzált predikcióval és veszteséges DCT-vel kódolt és rekonstruált képek, mozgás-kompenzált predikció nélkül teljes kép alapú predikcióval és veszteséges DCT-vel kódolt és rekonstruált képek. Hogyan lehet változtatni a kvantálás mértékét? Vizsgáljuk meg a durvább kvantálás hatását.

## 2. H.264 intra-predikció demo

A vonatkozó matlab script:

```
intra_pred_h264.m
```

A kód tetszőleges kép 4x4-es blokkméret alapú intra-predikciós hibaképet számolja ki a H.264/AVC szabvány szerinti 9 lehetséges predikciós irány/lehetőség használatával. A képet 4x4-es blokkokra osztja, és minden 4x4-es blokkot a tőle balra és fölötté lévő blokkok szomszédos pixeleinek meghatározott lineáris kombinációjából prediktíven kódol. Minden blokk esetében a lehető legkisebb predikciós hibát eredményező predikciós irányt választja.

**Vizsgáljuk meg, hogy különböző képek esetén a predikciós irányoknak hogyan változik a statisztikája.**

**Hogyan értelmezhető a program végén kiírt predikciós nyereség?**

```
fprintf('Prediction Gain = %5.2f dB\n', 20*log10(std2(A)/std2(E)))
```

**A predikciós irányok szerinti predikciós hibaképet számító IntraBlkPredict függvényben szereplő SAD kritérium mit fejez ki, és miért lehet előnyösebb a használata, mint az MSE használata?**