

Többnézetű videó megjelenítés és kódolás

DR. LOIS LÁSZLÓ, DARÓCZY BÁLINT, LUSTYIK TAMÁS

Budapesti Műszaki és Gazdaságtudományi Egyetem, Híradástechnikai Tanszék
lois@hit.bme.hu

AGÓCS TIBOR, BALOGH TIBOR

Holografika Kft.
t.agocs@holografika.com

Kulcsszavak: *többnézetű videó, MPEG-4, videó kódolás, 3D animáció és megjelenítés, háromdimenziós televízió*

Ebben a cikkben először ismertetjük a többnézetű megjelenítést és kódolást, ami nemcsak a szintetikus képtartalmat hordozó háromdimenziós (3D) animációs és tervezési eszközök, hanem majdan a természetes (nem szintetikus) tartalmú 3D televíziózás alapja is lehet. Röviden bemutatjuk a jelenleg legjobban elterjedt többnézetű megjelenítő eszközöket, ezután pedig a többkamerás felvételek kódolásának két alapvető megközelítési módját, a nézetek közötti predikcióval kiegészített hagyományos képek közötti predikciós technikát alkalmazó hibrid videó kódolást, valamint a mélységi kép alapú reprezentációt alapuló többnézetű videó kódolást ismertetjük. A két módszer áttekintése után egy olyan tesztelési eredményről számolunk be, ahol egy természetes felvételhez hasonló többnézetű videót kódoltunk le a hagyományos irányvonalat képviselő MPEG-4 AVC-vel, és ezt hasonlítottuk össze egy olyan megoldással, ahol csak néhány referencianézetet kódoltunk az MPEG-4 AVC-vel, és a maradék nézetet a referencianézetekhez tartozó kódolt mélységi információkkal állítottuk elő. Emellett egy új veszteségmentes tömörítési módszert is mutatunk a mélységi képek kódolására, amely jelentősen javítja a tömörítési hatékonyságot az ismert módszerekhez képest.

1. Bevezetés

A többnézetű videó lehet az egyik továbbfejlődési iránya a jelenlegi házimozi vagy műsorszóró rendszereknek. Ez a terület jelenleg jelentős átfedést mutat a számítógépes grafikával és animációval, hiszen a nagyon kisszámú beviteli és megjelenítő rendszerek miatt még a természetes mozgókép területén való elterjedése a legoptimistább elképzelések szerint is csak középtávon várható, így a rendelkezésre álló képsorozatok főleg mesterséges 3D-s animációkból származnak.

A többnézetű videó felvétele több, rögzített topológiában elhelyezett kamerával lehetséges, ahol a kamerák száma megfelel a megjelenítő eszköz által produkálni képes nézetek számával. Mivel jelenleg már létezik akár 60 horizontális nézetet is szimultán megjelenítő eszköz, így ezzel a megvalósítással 60 kamera jelét kellene párhuzamosan kódolni, vagyis 1 többnézetű csatorna akár 60 egynézetű csatorna sáv szélességét is elfoglalhatja.

A többkamerás videófolyamok kódolása új távlatokat nyithat meg a videokódolás előtt, hiszen az egykamerás felvételek kódolása során csak a képen belüli és az időben közeli képek közötti redundanciát aknázzuk ki, miközben többkamerás videófolyamok esetén már a nézetek közötti redundancia is kihasználható.

A nézetek közötti redundancia kihasználására a hagyományos videó forráskódolási algoritmusok is felkészíthetők. Már az ISO MPEG csoportja is definiálta a többnézetű profil kiterjesztést az MPEG-2 szabvány keretében, miközben az MPEG-4 szabvány már eleve úgy lett kialakítva, hogy több eszközt is tartalmaz egy jelenet tetszőleges nézetből való megjelenítésére interaktív módon. Az MPEG-4 multimédiás szabvány egyetlen

jelenethez több objektumok kódol egymástól szeparáltan és tetszőleges (nem téglalap alakú) formában olyan módon, hogy azok más nézőpontokból is megjeleníthetők legyenek. Az MPEG-2 fő alkalmazási területe inkább csak a kétkamerás felvételek esetében jöhet szóba, addig az MPEG-4-ben több eszközkészlet közül is válogathatunk és a nézetszám tekintetében sincs jelentős kötöttség. Az MPEG-4-en belül a kötöttség a természetes videóra sem vonatkozik, mert a virtuális 3D-s világban való navigáció során természetes videóanyagok is beilleszthetők.

Eközben azonban egy másik irányzatot képviselő tudományterület, a számítógépes grafikai háttérrel induló mélységi kép alapú reprezentáció (Depth Image-based Representation, a továbbiakban DIBR) is nyújt már megoldást a többkamerás nézetek kódolására. Ez az eszközkészlet részben szintén szerepel az MPEG-4 eszközök között, de érdemes hangsúlyozni a különbséget is, mert ez a DIBR sajátosan egyszerű eszköz a feladat megoldására, miközben az MPEG-4 vizuális kódolási eszközei jelentős számú, a többnézetű videó kódolás szempontjából felesleges többletfunkciókat is tartalmaznak.

2. Többnézetű videó megjelenítő eszközök

Számos cég és egyetem foglalkozik 3D technológiák fejlesztésével világszerte, de természetesen az általuk létrehozott rendszerek alapelveik szerint elkülöníthetők egymástól. Ebben a fejezetben a 3D megjelenítő rendszereknek egy lehetséges csoportosítása kerül bemutatásra.

2.1. Volumetrikus megjelenítők

A „volumetrikus” megjelenítők esetében egy féligáteresztő vagy diffúz közegre (felületre) vetítik ki a fénysugarakat, amely felület vagy mozgásban van vagy áll, ennek megfelelően különböztetünk meg mozgó képernyős, illetve dupla vagy sokrétegű rendszereket. A féligáteresztő vagy diffúz felületeken aztán szóródnak a fénysugarak (transzmissziós rendszer), illetve visszaverődnek (reflexiós) róla.

A *mozgó képernyős rendszerek* a volumetrikus megjelenítőknek olyan típusa, amelyben az elnevezésnek megfelelően egy mozgó ernyőt tartalmaz, melyre LED-ek, illetve LCD-k segítségével a megjelenítendő 3D objektum megfelelő perspektivikus képét vetítik ki. Természetesen a folyamat nagy sebességgel zajlik annak érdekében, hogy a szem által összemossott képek az agyban 3D objektumokká álljanak össze. Megfelelő szinkronizálás mellett tehát 3D tárgyak váltak megjeleníthetővé – igaz csak az ernyő által behatárolt térfogatban.

A *többrétegű rendszerek* esetén kettő vagy több folyadékkristály rétegre történik a projekció, de ezek a felületek nem a megjelenítendő 3D objektum egyes nézeteit jelenítik meg, hanem olyan kétdimenziós vetületeit, melyek egymással párhuzamosak. A mélység érzetét tehát a ténylegesen mélységben eltoló leképezések hozzák létre, a mélységi felbontás pedig a folyadékkristály rétegek számával egyenlő. A jelenlegi legfejlettebb architektúrával egy olyan volumetrikus megjelenítő rendelkezik, mely 20 XGA (1024x768) felbontású folyadékkristály-rétegeket tartalmaz. Ezek be-, illetve kikapcsolásra átlátszóvá, illetve átlátszatlaná válnak és ezáltal a 20 adott mélységben elhelyezve a projekciós ernyő szerepét is ellátják. A 20 réteg be- és kikapcsolása szinkronizálva van a különböző mélységű vetületeket tartalmazó képek kivetítésével és a fókusztávolságot végző adaptív optikával. A hátránya az elrendezésnek, hogy a megjelenített objektumok szellemszerű hatást keltenek, mivel a tér egyes pontjaiba érkező fénysugarak egyik rétegben sem nyelődnek el, valamint ebben az esetben is korlátozott a 3D-s megjelenítés rendelkezésére álló térfogat. Emiatt csak azok az alkalmazások jöhetnek szóba, melyeknél egy kisebb térfogatban van szükség a kívánt objektumok megjelenítésére, illetve könnyen megjeleníthető képszeletek segítségével.

2.2. Auto-sztereó megjelenítők

Az auto-sztereó megjelenítők speciális szemüveg (polarizációs, színszűrő, elsőtétülő stb.) nélkül képesek a 3D érzet kialakítására. Annak érdekében, hogy a jobb illetve bal szemek más és más képet lássanak, különböző optikai rétegeket helyeznek közvetlenül a nagyfelbontású képernyő fölé. Két alap technológiára bontható az autosztereó megjelenítők csoportja: sztereó és többnézetű megjelenítőkre.

Régóta léteznek két nézetet létrehozó autosztereó megjelenítők melyeknek alapját a „lenticular screen”

(hengerlencse-sor), „parallax barrier” (speciális rács) vagy „micro-polarizator” (speciális polarizációs eszköz) rendszerek adják

A képmegjelenítő eszköz tipikusan egy transzmissziós folyadékkristály réteg, melynek páratlan sorai vagy oszlopai a jobb szem számára hozzák létre a képet, míg a páros sorok vagy oszlopok a bal szem számára.

2.2.1 Sztereó megjelenítők

A *sztereó megjelenítők* két nézeti képet állítanak elő, ahol a két kép ezután a megjelenítő előtti tér egyes részeibe úgy van kivetítve, hogy az egyik képet a jobb szem lássa, a másikat pedig a bal. Ezen rendszerek leggyeszerűbb megvalósítása az úgynevezett passzív sztereó megjelenítő rendszerek.

A *passzív sztereó megjelenítő rendszerek* esetében a nézőnek egy adott ideális távolságban és megfelelő helyen kell állnia ahhoz, hogy sztereó képet lásson. Az elrendezés hátránya, hogy 50%-os valószínűséggel a néző a rossz pozícióban áll és ennek következtében hibás képet – úgynevezett pszeudo képet – lát. Ezen korlátozó tényezők miatt vált szükségessé a „tracking” vagy követő rendszerek bevezetése illetve olyan autosztereó megoldások kidolgozása, mint a többnézetű autosztereó megjelenítők.

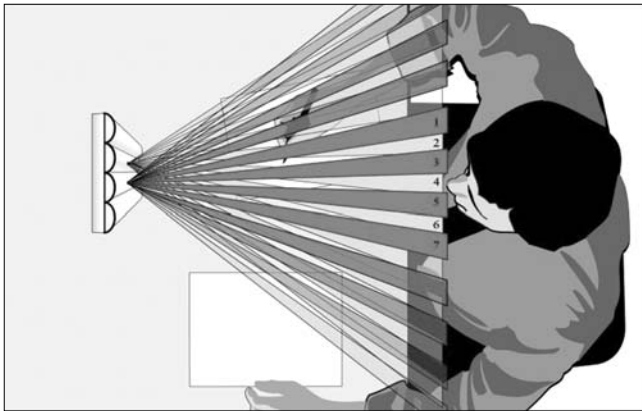
A *követő-rendszerek* képesek lekövetni a szem illetve a fej mozgását. Habár ez az elrendezés sem támogatja a több felhasználós – kollaboratív munkát lehetővé tevő – megoldást és lehetnek olyan effektusok, melyek zavarólag hatnak, mégis lehetővé teszi vízszintes parallaxisban a térbeliség megjelenítését és ennél fogva egyes alkalmazások számára kielégítő alternatíva lehet.

2.2.2 Többnézetű sztereó rendszerek

A *többnézetű sztereó rendszerek* alapelve, hogy a megjelenítő a tér különböző részeibe különböző képeket vetít ki. A képernyő előtti látótér fel van osztva véges számú, vízszintesen egymás mellett elhelyezkedő cellákra vagy ablakokra. Minden egyes ilyen cellában a háromdimenziós objektum egy perspektivikus (adott szögből látható) vetülete látható. A néző két szeme különböző képet lát, ugyanis más cellába esik a jobb, illetve bal szem, és ahogy a néző mozog a megjelenítő előtt, a szemei mindig más cellába esnek, ezzel létrehozva a 3D hatást. A rendszer hátránya, hogy miközben a felhasználó cellából cellába mozog a képek nem folyamatosan változnak, hanem az alacsony szögfelbontás miatt ugranak, ami zavaró lehet. Ezek a rendszerek is képesek a vízszintes parallaxis létrehozására, azaz a 3D objektumnak vízszintesen más-más szögből látható vetületeit képesek megjeleníteni. A többnézetű sztereó rendszerek három legjellegzetesebb fajtája a *lenticular screen*, a *parallax barrier* és az optikai szűrőkön alapuló megjelenítők.

A „*lenticular screen*” rendszer egy hengerlencse sor, mely egy nagyfelbontású LCD-re van helyezve oly módon, hogy az LCD képsíkja pontosan a lencsék fókuszsíkjában van. Az elrendezés eredménye, hogy a hen-

gerlencse alatt egymás mellett elhelyezkedő LCD pixelek más-más térszögből lesznek láthatóak, létrehozva ezzel a többnézetű rendszerekre jellemző cellás felosztást. A legújabb fejlesztésű hengerlencse sorokat tartalmazó megjelenítők 8-10 nézet létrehozására képesek.



1. ábra Hengerlencse alapú többnézetű megjelenítő

A „parallax barrier” megjelenítőkben egymáshoz nagyon közel (mikrométerekre) elhelyezett nagyon vékony apertúrák találhatók egymással párhuzamosan, melyek egy jól meghatározott távolságra találhatók a nagyfelbontású LCD-től. A többnézetű rendszerekre jellemző cella effektust az apertúra-rács hozza létre úgy, hogy mindig csak az adott nézetnek megfelelő pixelek láthatók, a többi pixel kitakarásban van. Ebben az esetben is – hasonlóan az előző kategóriához – a perspektivikus nézetbeli információk pixeloszlopok formájában vannak kódolva, így hozva létre a vízszintes parallaxis effektusát.

Az optikai szűrőkön alapuló megjelenítők optikai rétegeket tartalmaznak a nézetek térbeli felosztásához. A rendkívül kis méretű – az LCD pixeleivel nagyjából azonos nagyságú – RGB (piros, zöld és kék) szűrőket tartalmazó réteg a technológia kulcseleme. Egy létező megjelenítő megoldás nyolc perspektivikus nézet létrehozására alkalmas, amelyben az RGB szűrők egy ismétlődő struktúra szerint vannak elhelyezve. A speciális színszűrőkből álló réteg az LCD-n megjelenő „kombinált képről” érkező fényt úgy ereszti át, hogy az egyes irányokból más és más perspektivikus metszete látható a 3D-s objektumnak, mivel egy adott pozícióból nézve a fény bizonyos komponensei elnyelődnek, míg mások átteresződnek. Ez a típus már nem egyszerűen pixel-oszlopokban (amelyekben az RGB is benne van) kódolja a nézeteket, hanem a pixel-oszlopokban az egyes nézetek RGB komponensei vannak összekeverve.

2.3. Holografikus megjelenítők

Hasonlóan a hagyományos hologramokhoz, ezeknek a megjelenítőknek is megvan az a képességük, hogy tárolják a fény amplitúdó és fázis információját. Nagyrészt akusztó-optikai fénymodulátorokkal és speciális tulajdonságú folyadékkristá-

lyokkal kísérleteztek mozgó hologramok létrehozása érdekében, de mindeddig nem sikerült megfelelő méretű, használható rendszert építeni. A tisztán holografikus alapú technológia valódi 3D képből származó információt használ a holografikus interferencia-minta kiszámításához, és a számítógép által vezérelt lézersugár egy tükrrendszer segítségével képes 3D képek megjelenítésére. A sztereoszkóp és többnézetű megoldásokkal összehasonlítva legfőbb előnyeként a képminőség említhető, hátránya a nagy számítási kapacitás, amely az interferencia-minták kiszámításához szükséges.

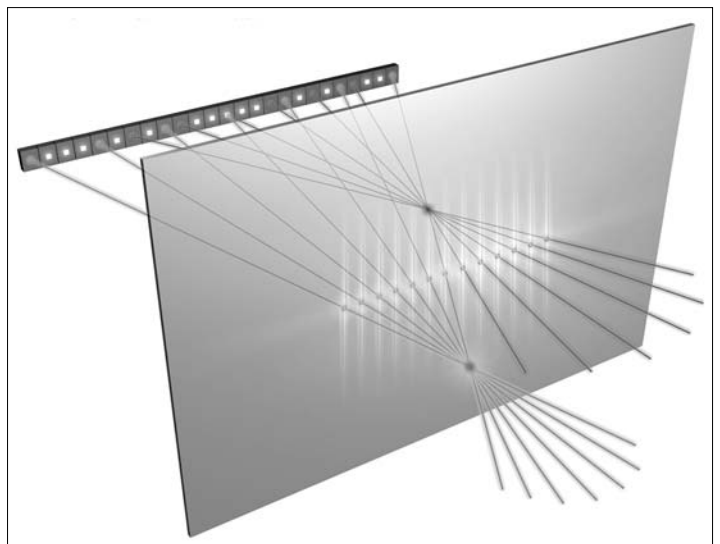
A Holografika kft. által kifejlesztett holografikus-technológia lényege, hogy a holografikus képernyőn minden egyes képpont képes különböző intenzitású és hullámhosszú fényt kibocsátani különböző irányokba. Ennek eredményeként megfelelő irányítás mellett bármilyen 3D-s nézet létrehozható. A fénysugarakat egy szabadalmaztatott speciálisan elhelyezett térbeli fénymodulációs rendszer hozza létre, és a holografikus képernyő végzi el a szükséges optikai transzformációt annak érdekében, hogy a fénysugarak tökéletes 3D-s nézetté álljanak össze.

Amikor egy pontot a képernyő előtt kell megjeleníteni, akkor a fénysugarak – melyek elhagyják a képernyőt – pontosan abban az adott pontban keresztezik egymást, és onnan terjednek tovább a néző felé. Mikor a képernyő mögött jelenik meg egy megjelenített pont, a képernyőről érkező fénysugarak úgy terjednek, mintha a képernyő mögötti pontból indultak volna ki, tehát a néző egy pontot lát a képernyő mögött (2. ábra).

A megjelenítő mindazon fénysugarakat előállítja, melyek jelen vannak egy valódi, természetes háromdimenziós objektum esetében, ez az oka annak, hogy ugyanaz látható, mint ami a valóságban. A legfontosabb előnye az elrendezésnek, hogy hasonlóan a tisztán holografikus megjelenítőkhöz, alkalmas a valódi mélységbeli információk megjelenítésére, valamint a korábban bemutatott típusokhoz képest nagy látószöggel rendelkezik.

2. ábra

A Holografika Kft. holografikus kijelzőjének működése



3. Háromdimenziós objektumok mélységi kép alapú reprezentációja

A mélységi kép alapú reprezentáció a számítógépes grafikai megjelenítő eszközökkel párhuzamosan fejlődött ki. A DIBR irányába való első lépést a 80-as években már elterjedt sprite-ok általánosítása jelentette. A sprite egy olyan síkbeli vetülete egy objektumnak vagy a 3D-s jelenet egy részének, amely affin transzformációval mozgatható a képsorozatok szintézise során a kamera mozgásának megfelelően. Nagyobb mértékű kamera mozgásnál vagy orientáció váltásnál azonban a sprite-ok ilyen formában történő felrajzolása már észrevehető torzítást eredményez. A sprite-ok képpontjai mellé letárolva azok mélységi értékeit is (sprite with depth [1]) már figyelembe lehetett venni a forma és a textúra finom mértékű nem affin torzulását is a kamera paraméterek változása függvényében. A mélység ebben a megfigyelőtől való távolságot jelenti.

A mélységgel rendelkező sprite-ok mintájára bevezették a mélységi információval kiegészített képet is (domborzati kép – relief image) [2], amelyben így egy képpont a színösszetevőkön túl tartalmazza már a mélységi értéket is.

Ma már léteznek olyan kamerák, amelyek a színösszetevőkön túl a képpont irányában a távolságot is érzékelni tudják, így ilyen jellegű domborzati kép felvehető már kamerával is. A domborzati kép felvételénél a kalibráción túl az egyetlen lényeges problémát csak a részben átlátszó felületek jelenthetik. Emellett azonban léteznek olyan algoritmusok is, amelyek többkamerás felvételeknél képesek meghatározni a különböző felvételek összetartozó képpontjait, és így háromszögessel meghatározható ezekre a képpontokra a távolság is.

A 3D-s jelenet bejárásánál a mélységi értékek segítségével megadható, hogy az új nézetben hová kerül a megfelelő képpont. Könnyen belátható, hogy a kamera elmozdulásával egyes képpont a képernyőn kívülre kerülnek, illetve átlapolódhatnak meglévő pontokkal, így a szintetizált kép hiányos lesz. A hiány betöltésére

többkamerás felvételeknél kedvező esetben használható egy másik kameraállásból felvett kép, illetve kisebb méretű lyukak esetén interpolációs szűrés.

Mivel az ilyen lyukakat az eredeti képen nem látható mintáknak kellene betömni, így ezen hiányzó minták közül azokat, amelyek nem a képen kívülről származnak, az eredeti kamerából látható képpontok fedik el. Ezen képpontok leírására szolgál a rétegzett mélységi kép (Layered Depth Image, LDI) [1], ahol egy képponthoz több színmélység értékpár (a szín maga is több, jellemzően háromdimenziós vektor 3x8 biten ábrázolva, míg a mélységi információt tipikusan 16 biten mintavételezik) is tartozik úgy, hogy a nem látható felületi pontok is szerepelnek a képernyő azon képpontjában, ahol a takarás hiányában látszanának. E leírási móddal a hiányzó képpontok egy része megadható, de ilyen jellegű leírást kamerával jelenleg nem lehet előállítani.

4. Többnézetű videó kódolás

Többnézetű videó kódolás esetén több kamerát helyezünk el egy adott jelenet felvételéhez, ahol a kamerák szinkronban vannak egymáshoz képest. Így azon képpontok, amelyek több kamerából is látszanak, azonos időpillanatban lesznek mintavételezve. A kamerák eltérő helyzetéből adódóan az azonos ponthoz tartozó képpontok a különböző csatornákon általában más-más koordinátára kerülnek, sőt az intenzitás és színtartalom sem feltétlenül egyezik meg egy valóságos felület esetén valóságos megvilágítási viszonyok mellett.

4.1. MPEG-2 többnézetű profil

Míg az MPEG-1 semmiféle ajánlást nem kínál sztereó képek, illetve más, egymás között redundáns tartalmú képsorozatok közös, hatékonyabb kódolására, addig az MPEG-2-ben már megjelent a sztereó-folyamok tömörítése. Az MPEG-2 multiview profilt 1996-ban definiálták MPEG-2 szabvány kiegészítéseként, fő tartalma ennek a profilnak az időbeli skálázás koncepciójának átértelmezése egy új profil keretében, valamint a kamera paraméterek tárolása az MPEG-2 szintaxisú bitfolyamban [3].

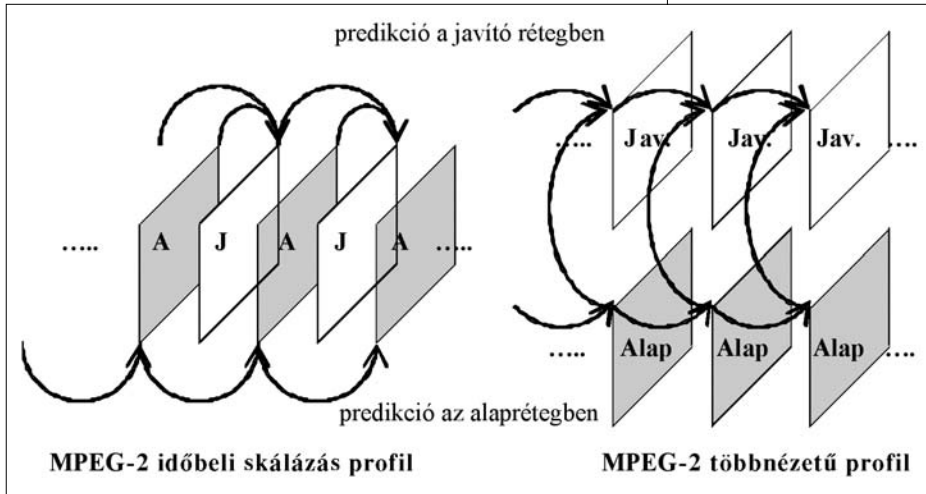
Ez a profil az MPEG-2 időbeli skálázást lehetővé tevő profilján alapul, ez utóbbi profil eredetileg az alaprétegen továbbított alacsonyabb képváltási frekvenciájú dekódolt képsorozatba illeszti be két szomszédos kép közé a javító rétegben küldött képet, így duplázva meg a képváltási frekvenciát. A javító rétegben minden makroblokkra megadható, hogy a mozgáskompensáció referenciáját az alap- vagy pedig a javító rétegből vegye-e.

Ugyanezt az elvet alkalmazták a többnézetű profilban is, itt azonban az a középső kameraállásnak megfelelő alaprétegből több javító réteg is becsülhető, ahol a javító rétegek természetesen egy-egy külön nézetet jelentenek, amelyekhez a megfelelő kamera paramétereket szintén a bitfo-



lyamban továbbítják. A rétegek közötti predikció így már nézetek közötti predikcióvá válik (3. ábra).

Az MPEG-2 többnézetű profilban csak abban az esetben beszélhetünk nyereségről a független kódolással kapott két teljes értékű szekvenciához képest, amennyiben a kiegészítő réteg(ek)et kisebb bitsebességgel lehet kódolni. A tapasztalatok alapján az oldal-só nézeteket javító rétegben minőségromlás nélkül kódolva nem kapunk szignifikáns tömörítési arány javulást, így a nézetek számának növelésével arányosan nő a szükséges sáv szélesség is [3].



3. ábra
Predikció az MPEG-2 időbeli skálázás és többnézetű profilokban

4.2. Más MPEG videókódoló kiegészítése nézetek közötti predikcióval

A fenti, MPEG-2-ben kimondott alapelv alkalmazható más kódolással is, amely képek közötti predikciót használ. Ilyenformában kézenfekvő a jelenleg ismert leghatékonyabb MPEG videótömörítés, az MPEG-4 AVC alkalmazása, hiszen itt már magának az AVC-nek a bevezetésével is jelentős kódolási hatékonyság növekedést lehet elérni.

Az MPEG-4 AVC [4] téglalap alapú objektumok kódolására alkalmas, így például természetes kamera képek kódolására is. A többkamerás rendszerek esetében az egyik legnagyobb problémát a kamerák helyes kalibrációja jelenti, hiszen több kamera esetében egyre komplexebb feladattá válik az azonos fényérzékenység beállítása.

Emiatt [5]-ben a képek közötti blokk alapú mozgás-kompenzációt kiegészítik egy megvilágítási kompenzációval is, ahol a megvilágítás hatását az intenzitás jelen történő eltolással és skálázással modelleznek. Így egy blokkhoz nemcsak a mozgásszegmentációs információt és a mozgásvektort, hanem az előbb említett két megvilágítási paramétert is kódolják. A módszer a tesztelések során 0.5-1.0 dB-es nyereséget mutatott, sőt a szubjektív értékelés során azt tapasztalták, hogy a kritikus helyeken előforduló hibák jelentősen csökkentek.

4.3. MPEG-4 Part 16: Animation Framework eXtension (AFX)

Az MPEG-4 AFX [6] jelentős mértékben tartalmazza a mélységi kép alapú reprezentáció eszközeit is. Az AFX statikus és animált 3D-s objektumait többnyire a VRML (Virtual Reality Modeling Language) alapján dolgozták ki, és természetesen az eszközök kompatibilitásuk a BIFS csomóponttal, a szintetikus bitfolyammal és az audióvizuális bitfolyammal is [7-9]. Az AFX fő adatstruktúrái között található két olyan is, amely a DIBR szempontjából döntő jelentőségű, ezek a *SimpleTexture* és a *PointTexture* adatstruktúrák.

A *SimpleTexture* adatstruktúra tartalmaz egy 2D képet, egy mélységi képet és a kamera paramétereit (pozíció, orientáció, projekció). A mélységi kép képpontjai továbbíthatók képpontonként az intenzitás és a színjelek mellett mint egy negyedik koordináta, illetve továbbítható a mélységi kép egy külön szűrkeskálás bitfolyamként is. A mélységi képek kódolása jellemzően veszteségmentes, hiszen a mélységi

információ torzulása a kamera paraméterek változása mellett képpont pozíció hibázást eredményez.

Egyetlen *SimpleTexture* adatszerkezettel azonban nem írható le egy objektum, *SimpleTexture* inkább csak egy irányból való megtekintésre alkalmas, a középső kamera állásból való kismértékű elmozdulást és orientációváltást engedélyezve.

A bonyolultabb esetek kedvéért a *PointTexture* adatstruktúrát kell használni az objektumok leírására. A *PointTexture* adatstruktúrában az objektumnak különböző egyenesekkel való metszeteit tároljuk úgy, hogy minden metszéspontra letároljuk a metszéspont mélységi értékét és a képpont színét.

4.4. Mélységi kép alapú többnézetű állókép és videó kódolás

Az MPEG-4 eszközein túl több irodalmi eredményben is beszámolnak mélységi kép alapú kódolásról. Az egyik legátfogóbb eredményt [10]-ban jelent meg, ahol az MPEG-4 AFX *OctreeImage* reprezentációjának hatékony kódolására kidolgozott saját algoritmuson túl kimerítő és érdekes áttekintés található a DIBR témakörrel is.

LDI kódolására több eszközkészlet is alkalmas, nemcsak az MPEG-4 AFX. Erre egy több alternatívát tartalmazó megoldás található [11]-ben:

- Minden képpontban kódolják a rétegek számát, azaz a színmélység értékpárok számát. Ez a kép szűrkeárnyalatos képként kódolható, általában kevés gradációs szinttel. A JPEG-LS [12] algoritmus

jelentősen hatékonyabbnak bizonyult a Deflate (ZIP) algoritmusnál mind a tömörítési arány, mind pedig a futási idő szempontjából.

- A különböző színösszetevő és mélység rétegeket külön kódolják. Mivel az egyre nagyobb rétegeken egyre kevesebb a képpont, így az alábbi alternatívákat vizsgálták meg:
 - MPEG-4 forma-adaptív (Shape Adaptive) DCT,
 - téglalap alakúra kiegészített kép kódolása JPEG 2000-rel,
 - MPEG-4 arbitrary shape codec,
 - VOW (Video Object Wavelet) codec, amely az előzőhöz hasonlóan szintén tetszőleges alapú kép kódolására alkalmas eszköz.

A tapasztalatok szerint a VOW teljesített a legjobban a színrétegek és a távolságrétegek kódolásánál is.

A JPEG 2000 kódolást alkalmazták egyetlen kép többnézetű kódolására is [13]-ben. Itt a szerzők azt tapasztalták a mérések során, hogy a 16 bites mélységi képek kódolására JPEG-2000 állókép kódoló 0.3 bpp-es tömörítés mellett adott ki olyan rekonstruált képminőséget, amely a mélységi képből való reprezentáció számára még elfogadható. A mélységi kép és a kép elemzése alapján állítják be a JPEG-2000-ben a fontos régiókat a ROI (Region Of Interest) kódoláshoz, valamint a mélységi információt kompendálják a kódolás előtt.

5. Kísérleti eredmények, tapasztalatok

Ebben a szakaszban egy általunk generált virtuális szekvencián vizsgáltuk meg a mélységi kép alapú többnézetű videó kódolás hatékonyságát. A szekvenca kialakításánál arra törekedtünk, hogy egy természetes képi tartalomhoz hasonló jelenetet állítsunk össze. A mérések során pedig egy olyan beállítást kerestünk, ahol 60 kamera többnézetű képsorozata átvihető egy 24...30 Mbit/s-os csatornán, például egy teljes DVB-T multiplexet lefoglalva.

A kamerák itt természetesen egy virtuális térben voltak, ahol 12 különböző személy- és teherautó mozgott két szembejövő kanyarodó sávban, a háttérben pedig tömbház és több különböző fa volt. A kép felbontását a Holografika előző generációs kijelzőjéhez választottuk, amely 512x320, ez 40%-a az SDTV felbontásnak.

Az autók modelljeit eredetileg a Need For Speed 3 Hot Pursuit számítógépes játékhoz készítették a játék rajongói, ezek könnyen beolvasható formátumú és jól animálható objektumok.

Az autók esetében törekedtünk a minél több takarás megvalósítására, ezért átlátszó vagy rácsos ablakú, vezetőt és esetleg utasokat tartalmazó modelleket alkalmaztunk, a kamera pedig egy olyan cél-autót követ, amelyben sok utas ül és három kerékpárt is szállít a tetőcsomagtartón.

Ilyen módon – alkalmazkodva a holografikus kijelző paramétereire – 61 kameraállást vettünk fel a cél-autóra fókuszálva a vízszintes síkon, a cél-autótól egyenlő távolságra, egymástól 1°-onként orientálva.

A kameraállások közül csak néhányat tömörítettünk, a többi pedig a meglévő csatornák kameraképei és mélységi képei alapján szintetizáltuk. A vizsgált nézetszámokat és a hozzájuk tartozó kamera szögeket az 1. táblázat tartalmazza. A szintetizálás során első lépésben a mélységi kép bitmélységét vizsgáltuk, és azt találtuk, hogy a 9 bites egyenletes skalár kvantálás már elégséges az elfogadható, 38 dB körüli PSNR értékű képminőség eléréséhez akkor, ha a szintézishez az eredeti kameraképeket használjuk.

5.1. Képszintézis MPEG-4 AVC-vel kódolt referenciaképekkel

A képszintézishez szükség volt a kamerapozíció leírására is. A többnézetű videót OpenGL-ben készítettük el, így a kamera paramétereit is az OpenGL nézeti transzformációjának megfelelően nyertük ki.

Az OpenGL esetében egy virtuális 3D-s térben lévő (x, y, z) pontot az alábbi lépésekben képezi le a rendszer a képernyő egy (u, v, d) koordinátájára:

1. Nézeti transzformáció:

$$[x', y', z', h'] = [x, y, z, 1] \cdot T_{view} \quad (5.1)$$

ahol T_{view} a nézeti transzformációt jelöli, és a homogén koordinátás ábrázolásnak megfelelően négydimenziós vektorral dolgozunk. Ez a 4x4-es mátrix a kamera beállítások után kérendő le az OpenGL-től.

2. Perspektív transzformáció:

$$[x'', y'', z'', h''] = [x', y', z', h'] \cdot T_{pers} \quad (5.2)$$

ahol T_{pers} a perspektív transzformációt jelöli. Ez a 4x4-es mátrix szintén lekérdezhető az OpenGL-től.

3. Képernyő transzformáció:

$$u = (x'' + 1) \cdot \frac{V_{sx}}{2} + V_x \quad v = (y'' + 1) \cdot \frac{V_{sy}}{2} + V_y$$

$$z_d = \frac{Z_{max} - Z_{min}}{2} \cdot z'' + \frac{Z_{max} + Z_{min}}{2} \quad (5.3)$$

ahol u és v jelölik a szintetizált képen a képpont pozícióját, z_d jelöli a mélységi információt, a minimális és maximális mélységet z_{min} és z_{max} jelöli, a képernyő szélessége és magassága pedig v_{sx} és v_{sy} , a bal-alsó

1. táblázat

A mérésben referenciaként használt nézetek kamerái

Referenciának használt nézetek száma	Kamera szögek
13	0°, ±5°, ±10°, ±15°, ±20°, ±25°, ±30°
11	0°, ±6°, ±12°, ±18°, ±24°, ±30°
7	0°, ±10°, ±20°, ±30°
5	0°, ±15°, ±30°

koordinátát pedig (v_{sx}, v_{sy}) jelöli. Ez a lépés szintén összefogható egy 4x4-es mátrixba, amelyet most egyszerűen T_3 -mal jelölünk, így

$$[u, v, z_d, \tilde{h}] = [x'', y'', z'', h''] \cdot T_3 \quad (5.4)$$

Az előbbieken megadott egyenlőségeket felhasználva a DIBR alapú képszintézis a következő lépésekből áll:

- Jelölje a k -ik nézetben $R_k(\cdot)$ a referenciaképet, $Z_{buff,k}(\cdot)$ pedig a hozzá tartozó mélységi képet. Célunk a k -ik képből szintetizálni az n -ik képet a z -puffer algoritmusával. Tegyük fel, hogy a z -puffer már tartalmaz bizonyos pontokat, a hiányzó pontok mélységi értéke pedig ∞ .

- A kép minden (u_k, v_k) koordinátájára:
 - Kiolvassuk a mélységi kép (u_k, v_k) pozíciójából a z_k mélységi információt:

$$z_k = Z_{buffk}(u_k, v_k) \quad (5.5)$$

- Meghatározzuk az eredeti 3D-s koordinátát, azaz (x, y, z) -t:

$$[x, y, z, h] = [u_k, v_k, z_k, 1] \cdot (T_{view,k} \cdot T_{pers,k} \cdot T_{3,k})^{-1} \quad (5.6)$$

- Meghatározzuk az n -ik képen a koordinátát és a mélységet:

$$[u_n, v_n, z_n, h_n] = [x, y, z, h] \cdot T_{view,n} \cdot T_{pers,n} \cdot T_{3,n} \quad (5.7)$$

A homogén osztás után kapott pozícióra pedig bemásoljuk a képpontot akkor, ha a z -puffer szerint közelebb van, mint az eddigi pont.

A fenti algoritmussal tehát az összes referenciaképet alkalmazva felépíthető a szintetizált kép. Nyilvánvalóan a mátrix szorzások összevonhatóak, így az $(u_k, v_k) \rightarrow (u_n, v_n)$ konverzióhoz egyetlen mátrix szorzást kell csak elvégezni. Hasonlóan számítható az (5.6) és (5.7) alkalmazásával a k -ik kép egy képpontjának az n -ik képen a mélységi értéke is, nemcsak a képernyőn lévő pozíciója, tehát ilyen módon a mélységi képek közötti redundanciát ki lehetne használni.

A 4. ábra különböző részei a kép szintézisének lépéseit mutatja be. Az első lépésben a nézetszámban legközelebbi referenciaképről építjük fel a Z -puffer algoritmussal a szintetizált képet (4.1. ábra), ezután nézetszámban kifejezett távolság szerinti növekvő sorrendben következnek a referenciaképek. A következő referenciakép hatását a 4.3 ábra mutatja. Ezután a többi referenciaképet is felhasználjuk a hiányzó pontok kitöltésére (4.4. ábra), végül pedig a még mindig hiányzó pontokat interpolációs szűréssel állítjuk elő.

Annak érdekében, hogy a távolabbi referenciaképekről jövő bizonytalanabb becslés ne okozzon problémát, így a z -pufferben való vizsgálat során a mélységi érték különbséget megnöveljük a referenciaképtől való távolság függvényében, és csak akkor módosítjuk a képpontot, ha ezzel a büntető értékkel együtt is negatív lett a különbség. Ezzel a megoldással az azonos pontok nem fogják felülírni egymást környezetét, mivel a mélységi értékek kvantálása miatt a fenti transzformáció távoli referenciaképre 1-2 képpontnyi tévedést okoz a pozícióban az (u, v) koordinátarendszeren belül.

A szintetizált képek és a dekódolt referenciaképek minőségének vizsgálata során azt tapasztaltuk, hogy a szubjektív képvizsgálatok szerint a képminőség csak a szintetizált képek bizonyos területein rosszabb a dekódolt referenciaképekhez képest (tipikusan az objektumok határpontjainál), azonban a PSNR-ben kifejezett képminőség drasztikus minőségromlást jelzett a szintetizált képek rovására.

Ezért a mérésekhez a hagyományos PSNR torzítás-kritérium mellett két másik kritériumot is megvizsgáltunk abból a célból, hogy a szintetizált képen levő durva hibák okait megtaláljuk. A továbbiakban $PSNR_{lefedett}$ jelölésű torzításkritérium a DIBR képszintézis során megkapott pontokban mért PSNR értéket adja meg, azaz ekkor eltekintünk az interpolációval megkapott pontoktól. A másik, a $PSNR_u$ jelű PSNR alapú hibakritérium a DIBR képszintézis raszterizációs lépésének hibájától próbál eltekinteni oly módon, hogy a kerekítés során az (5.7) lépésben kapott u vízszintes képernyő koordináta érték kerekítését úgy végzi el, hogy az a torzítás szempontjából kedvezőbb legyen. Ez utóbbi kritérium a dekóderben nem alkalmazható, mivel ehhez ismernie kellene a szintetizált kép eredetijét.

A 2. táblázat a három torzításkritérium által adott eredményt mutatja be 5, 7, 11 és 13 nézet esetén, 32-es AVC kvantálási paraméterrel. A PSNR és $PSNR_u$ kiértékelésekor egy aluláteresztő szűrést is végeztünk a szintetizált képen, ezt a lépést a $PSNR_{lefedett}$ esetén nem alkalmaztuk a lefedetlen pontokban lévő nem definiált értékek miatt.

A táblázatból megállapítható, hogy a legjobb eredményt akkor kapjuk, ha a raszterizációs lépést úgy hajtjuk végre, hogy a kerekítést mindig kedvező irányba végezzük el. Mivel nincs olyan információ a dekóderben, amely alapján ez a kerekítés megfelelően elvégezhető lenne, ezért ez egy elvi mérés csak marad és konkrét rendszerben nem alkalmazható. Megállapítható azonban, hogy a nem lefedett pontok kihagyása a mérésből gyakorlatilag nem okoz hibát, sőt azáltal, hogy ekkor nem végzünk utószűrést, még romlik is az objektíven mért képminőség.

Nézetek száma	Csak a szintetizált képekre			Minden képre		
	PSNR	PSNR _{lefedett}	PSNR _u	PSNR	PSNR _{lefedett}	PSNR _u
5	24.988	24.625	28.029	25.726	25.394	28.518
7	25.589	25.282	28.831	26.555	26.283	29.425
11	25.783	25.527	29.082	27.269	27.059	29.973
13	25.776	25.535	29.073	27.535	27.345	30.129

2. táblázat
Adott nézetszám mellett az átlagos PSNR értékek a dekódolt referencia képekből szintetizált képekre, valamint az összes dekódolt képre



4.1. ábra
Becslés a balra lévő szomszédos referenciaképről
(PSNR=24.10 dB).



4.2. ábra
Becslés a jobbra lévő szomszédos referenciaképről
(PSNR=27.63 dB).



4.3. ábra
A két szomszédos referenciaképről való
eredő becslés (PSNR=33.45 dB).
A jobb oldali nagy fa felett és mellett, valamint
a fehér autó előtt képpontok hiányoznak.



4.4. ábra
A többi referenciakép felhasználása
a 2.3. ábra becsléséhez (PSNR=34.64 dB).
A hiányzó képpontok többsége megllett.



4.5. ábra
A 2.4 ábrán látható eredmény szűrése
a hiányzó pontok kitöltésével (PSNR=36.25 dB).



4.6. ábra
Az eredeti kép

Ezekből a mérésekből azt a tapasztalatot vonhatjuk le, hogy a raszterizáció pontosítása és valamilyen adaptív utószűrés lehet az a két eszköz, melyeket érdemes fejleszteni a képsztízis javítása érdekében.

Azt is megállapíthatjuk a szintetizált nézetek átlagos képminőségi értékeit vizsgálva, hogy esetében a referencia nézetek számát 7-re célszerű megválasztani, hiszen 11 és 13 nézetre már a szintetizált nézetek átlagos PSNR értékei nem javulnak lényegesen, miközben a mélységi képekből is és a referencia képekből is többet kell kódolni.

5.2. Mélységi térképekhez rendelt saját veszteségmentes tömörítés

A mélységi képek olyan kritikus információt tartalmaznak, amelyet vissza kell tudni állítani hibamentesen a hiányzó nézetek rekonstrukciójához. Emiatt itt nem alkalmazható a veszteséges tömörítés.

A kvázi-szabványos alkalmazások használata ezen a ponton ott válik nehézé, hogy nem 8, hanem 9 bites szimbólumaink vannak, így a módszereket adaptálni kell.

A teszteléshez az alábbi kvázi-szabványnak tekinthető veszteségmentes adat-, illetve képtömörítő algoritmusokat vizsgáltuk meg:

Deflate algoritmus (ZIP):

Ez a módszer az LZ-77 (Lempel-Ziv) módszer egy variációja, ahol 32 kilobájtos visszatekintő puffert alkalmaznak, és ebben keresik meg a most következő karakterek legfeljebb 258 hosszú múltbeli előfordulását. A kódszimbólum vagy az új karakternek megfelelő kódszimbólum (<új karakter>), vagy egy <hossz><múltbeli távolság> kódszimbólum-pár, ahol a <hossz> és az <új karakter> szimbólum-ABC közös. Így a dekóder először ebbe a kódszimbólum-ABC-be eső szimbólumot vár, és a vétel után tudja eldönteni, hogy kell-e még a <múltbeli távolság>-ot is beolvasni.

GIF87a (LZW):

Ez a formátum valójában az LZ-78 módszer Welch-féle kiterjesztésére épül. Eredetileg 1, 2, 4 vagy 8 bites szótárral indul, és a szótárfa 12 bitig bővül.

Burrows-Wheeler transzformáció

WFC és RLE-BIT algoritmussal:

Ez a módszer a Burrows-Wheeler transzformáción (BWT) és az erre épülő, Jürgen Abel által készített *abc* veszteségmentes tömörítőkön alapul [14]. A módszer ismert arról, hogy különösen szövegállományok esetében a Deflate algoritmust akár 50%-ban is képes túlszárnyalni a tömörítési arány tekintetében, azaz a kapott fájl méret a 2/3-a lesz a másik algoritmus által tömöríthetethez képest.

PNG (Portable Network Graphics):

Ez a módszer szomszédos soronként kódolja a képet úgy, hogy minden sorra kiválaszt egy prediktort az ötféle beépített prediktorból, és elküldi a prediktor kódját. Ezután csak a predikciós hibát kell tovább kódolni, amit az LZ-77 eljárással tömörítenek 32 kilobájtos visszatekintő puffer alkalmazásával. Ez a módszer képes 8-nál több bites forrásszimbólumokat is kódolni, azonban csak úgy, hogy mindig leválaszt a forrásszimbólumból 8 bitet, és a maradékot egy következő körben kódolja le. Így a 9 bites forrásszimbólum-sorozat kódolását egy 8 bites és egy 1 bites szimbólumsorozat kódolásként hajtja végre.

JPEG-LS kódolás:

Prediktív veszteségmentes kódolás, ahol a becslést a már meglévő három szomszédos képpontból végzik el, a becslési hibát pedig Huffman-kódolással tömörítik.

A 9 bites forrásszimbólumok kódolását a JPEG-LS kódolás teljes mértékben támogatja. A többi módszer közül egyedül az LZ-77 módszer tudja módosítás nélkül kezelni akkor, ha a 9 bites szimbólumok 2 bájttal vannak tárolva.

Ekkor a veszteség csupán annyit, hogy a <hossz> <múltbeli távolság> szimbólumpárban általában páros hosszok és távolságok fognak dominálni, vagyis 1 bit redundáns lesz, az <új karakter> pedig hol 8 értékes bitet, hol csak 1 értékes bitet fog tartalmazni.

A tesztelés során azonban kidolgoztuk a saját 9 bites forrásszimbólumokon működő LZ-77-es algoritmust is, így ezzel már ez az elvi veszteség sem lépett fel. A LZ-77 algoritmus 9 bites verziójánál érdekes még, hogy mekkorára válasszuk ki a legnagyobb múltbeli távolságot, ugyanis ezt az értéket kisebbre választva az algoritmus gyorsabb lesz, és ha nincs szükség a nagy értékekre, akkor ezzel nem is veszítünk. Hasonlóan lehet csökkenteni vagy növelni a legnagyobb hossz értékét is.

A tesztelések alatt azt tapasztaltuk, hogy az eredeti 258-as hossz mellett 8 kilobájt az optimális múltbeli puffer méret, mert e fölött már csökken a tömörítési arány, aminek az oka, hogy kevés esetben fordul elő nagy távolság. A kisebb pufferméret pedig szintén ront a tömörítési arányon, mert ekkor értékes múltbeli minták csúszhatnak ki a pufferből. Nagyobb, 2K méretű maximális távolság esetében azonban azt tapasztaltuk, hogy lehetséges a keresési puffer 2 kilobájtosra való csökkentése a tömörítési arány minimális romlásával (néhányszor tíz bájt másodpercenként). A teszt eredményeket tartalmazó táblázatban ezért ezt a két módszert tüntettük fel, mint az LZ-77 kódolás saját adaptációját.

A GIF-ben található LZW kódolás módosítása egyszerű, itt csak annyit kell tenni, hogy a 8 bites szótár helyett 9 bites szótárral indítjuk a kódolást, és ezért a 12 bit helyett 13 bitig növesztjük a szótárat. A tesztelés során használt algoritmusban az a frissítési stratégia vált be, hogy a szótár megtelésekor azonnal ürítjük annak tartalmát, vagyis visszatérünk a 9 bites szótárra.

Módszer	Összes fájl méret 610 képre [Bájt]	Átlagos bitsebesség egy nézetre [Mbit/s]
LZW: 9 bittől 13 bitig	19332712	6.34
WinZIP (LZ77, 8 bites)	12049986	3.95
Saját 9 bites LZ77 kódolás	10474861	3.43
Saját 9 bites PNG kódolás (PNG prediktor + LZ77)	11782353	3.86
Prediktor + RLC + Huffman	12261700	4.02
JPEG-LS	16188900	5.31
BWT + RLE-BIT + WFC	8321324	2.73

3. táblázat
A különböző
veszteségmentes
tömörítő eszközök
hatékonyságának
összehasonlítása
a 9 bites mélységű
képen

Hasonlóan egyszerű a Burrows-Wheeler transzformáció 9 bites módosítása is, ekkor a 8 bites szimbólumok helyett 9 bites szimbólumokat használunk. Ebben az esetben az Abel-féle módosításnál az aritmetikai kódolás bitfolyam szerkezeténél csak a nem-futamhossz szimbólumok kódolását kellett kibővíteni, ami a meglévő hét aritmetikai kódoló mellé egy nyolcadik felvételt jelentette a megfelelő forrás-ABC mérettel.

A PNG-nek megfelelő kódolást is teszteltük, itt csak a PNG prediktorait kellett megvalósítanunk, hiszen a 9 bites LZ-77 kódolást az előző tesztekhez már kidolgoztuk. Utolsó módszerként kipróbáltuk azt is, amikor a PNG prediktorának predikciós hibáját futamhossz kódolással (Run Length Coding, RLC) és Huffman entrópia kódolással tömörítettük.

A tesztek eredményét a 3. táblázat mutatja be. Ebből megállapítható, hogy a Burrows-Wheeler transzformáció jelentősen jobb eredményt ér el, mint a többi módszer. A módszer hátránya a viszonylag nagy futási idő, ebből a szempontból a Prediktor + RLC + Huffman hibrid kódoló a legjobb, amely ráadásul nem szerepelt rosszul az összehasonlításban a BWT-t leszámítva.

5.3. A referenciaképek és mélységi képek becsült együttes sávszélesség igénye

A 4. táblázat azt mutatja be, hogy a generált tesztszekvenciára mekkora bitsebesség igény lép fel az adott számú referencia nézet mellett mélységi rekonstrukcióval. A kapott eredmények azt mutatják, hogy a 61 kamerás felvételhez a körülbelül 27 dB-es PSNR értékű minőségi szinthez már 20.74 Mbit/s is elegendő. Körülbelül ugyanezen a bitsebességen 25.7 dB-es rekonstrukció lehetséges a hibrid módszerrel 5 nézet esetében, azonban ennél több nézetszámra a bitsebesség drasztikusan megemelkedik a mélységi képek kódolásához szükséges magas bitsebesség miatt.

Ezek a bitsebességek már alkalmasak arra, hogy a fejezet elején említett sávszélesség korlátot betartva továbbítani lehessen a 61 nézetű videót. A két módszer közül azonban az AVC egyértelműen előnyösebb a valamivel magasabb minőség, a szabványosság és a kidolgozottság miatt, a hibrid módszerben azonban még jelentős tartalékok vannak, például a raszterizáció pontosításától és a mélységi képek közötti redundancia kihasználásától további hatékonyság növekedést várunk.

Az eredmények értékelésekor ki kell emelni azt is, hogy míg a hibrid módszernél a referenciaképek jobb

minőségben kódoltak (QP=32), addig a szintetizált képek rossz minősége lerontja az átlagos PSNR értékét arra a szintre, amit az AVC kódolás QP=42 kvantálási paraméterrel képes elérni.

Ez a megfigyelés, valamint az előző szekcióban megfogalmazott tapasztalat azt mutatja, hogy célszerű minimálisra választani a referencia nézetek számát, főleg a mélységi képek veszteségmentes tömörítéséhez szükséges magas bitsebességek miatt. Ugyanitt egy lehetséges tömörítési arányt javító megoldás lehet a mélységi képek közötti redundancia kihasználása például az (5.6) és (5.7) alkalmazásával végrehajtott predikcióval és a predikciós hiba veszteségmentes kódolásával, sőt ugyanígy az azonos nézetben az időbeli redundancia is kihasználható, ezt azonban ezen méréseink során még nem vettük figyelembe.

Megjegyezzük továbbá, hogy a kijelző 2D felbontása csak 512x320, ami a standard televízió 720x576-os felbontásának csupán a 40%-a, valamint azt, hogy a 27 dB alatti PSNR érték nem elfogadható érték műsor-szórás célú felhasználásra.

Mindezen tapasztalatok alapján a Híradástechnikai Tanszék és a Holografika Kft. úgy döntött, hogy az első 60 nézetet tartalmazó videós átviteli méréseket 60 független videókódolóval indítja el, de a kutatást folytatjuk a hibrid megoldás további fejlesztésével és a közeli jövőben már ezt a megoldást is alkalmazni kívánjuk a többnézetű videó megjelenítő videó formátumaként.

6. Összefoglalás

Ebben a cikkben röviden bemutatuk a többkamerás felvételek kódolásának két alapvető megközelítési módját. A mérési eredményeinkben egy olyan tesztelési eredményről számoltunk be, ahol egy természetes felvételhez hasonló többnézetű videót kódoltunk le mind a hagyományos irányvonalat képviselő MPEG-4 AVC-vel, mind hibrid MPEG-4 AVC és DIBR módszerrel. Emellett egy új, veszteségmentes tömörítési módszert is mutatunk a mélységi képek kódolására, amely jelentősen javítja a tömörítési hatékonyságot az ismert módszerekhez képest.

További kutatási témát jelenthet a mélységi képek veszteségmentes tömörítésénél az, hogy sok képpont mélységértéke becsülhető lenne más nézetekből vagy más időpillanattól is, amit prediktív kódolással lehetne kihasználni.

Referencia nézetek száma	AVC QP	Referencia nézetek [Mbit/s]	Mélységi képek [Mbit/s]	Teljes bitsebesség [Mbit/s]	Átlagos PSNR [dB]
5	32	7.44	13.65	21.09	25.726
7	32	10.42	19.1	29.52	26.555
11	32	16.37	30.01	46.38	27.269
13	32	19.34	35.47	54.81	27.535
61	42	20.74	nincs	20.74	26.678

4. táblázat
Adott nézetszám és MPEG-4 AVC kvantálási paraméter (QP) mellett az átlagos bitsebesség értékek a dekódolt referencia képekre és a belőlük szintetizált képekre

Érdekes kérdés továbbá a mélységi képek veszteséges tömörítésének értékelése, itt szintén az MPEG-4 AVC tűnik a legjobb megoldásnak, mert magas minőségű kódolást is lehetővé tesz. Hasonlóan jelentős feladat lehet még a raszterizációs lépés pontosítása, például O-puffer (offset buffer) vagy több képpontos felületi elemek (foltok) használatával. Vizsgálható tovább még a nézetek közötti predikció alkalmazása és esetlegesen a szintetikus képek predikcióba való bevonása, de itt a mélységi információ jelentős tömörítése tűnik szükségesnek.

Irodalom

- [1] Jonathan Shade, Steven Gortler, Li-wei Hey, Richard Szeliski, „Layered Depth Images”, SIGGRAPH 98, Orlando Florida, July 19-24, 1998. Computer Graphics Proceedings, pp.231–242.
- [2] M. Oliveira, G. Bishop, D. McAllister, „Relief textures mapping,” in Proc. SIGGRAPH, July 2000, pp.359–368.
- [3] Jens-Rainer Ohm, „Stereo/Multiview Video Encoding Using the MPEG Family of Standards”
- [4] ISO/IEC 14496-10:2003, Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding
- [5] Joaquin Lopez, Jae Hoon Kim, Antonio Ortega, George Chen, „Block-based Illumination Compensation and Search Techniques for Multiview Video Coding”, Picture Coding Symposium, San Francisco, CA, December 2004.
- [6] Information Technology – Coding of Audio-Visual Objects – Part 16: Animation Framework eXtension (AFX), ISO/IEC Standard JTC1/SC29/WG11 14 496-16:2003.
- [7] Information Technology – Coding of Audio-Visual Objects – Part 1: Systems, ISO/IEC Standard JTC1/SC29/WG11 14 496-1.
- [8] Information Technology – Coding of Audio-Visual Objects – Part 2: Visual, ISO/IEC Standard JTC1/SC29/WG11 14 496-2.
- [9] Information Technology – Coding of Audio-Visual Objects – Part 3: Audio, ISO/IEC Standard JTC1/SC29/WG11 14 496-3.
- [10] Leonid Levkovich-Maslyuk, Alexey Ignatenko, Alexander Zhirkov, Anton Konushin, In Kyu Park, Mahnjin Han, Yuri Bayakovski, „Depth Image-Based Representation and Compression for Static and Animated 3-D Objects”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 14, No.7, July 2004, pp.1032–1045.
- [11] Jiangang Duan, Jin Li, „Compression of the Layered Depth Image”, IEEE Transactions on Image Processing, Vol. 12, No.3, March 2003, pp.365–372.
- [12] M. Weinberger, G. Seroussi, G. Sapiro, „The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS,” IEEE Trans. Image Processing, vol. 9, August 2000, pp.1309–1324.
- [13] Ravi Krishnamurthy, Bing-Bing Chai, Hai Tao, Sriram Sethuraman, „Compression and Transmission of Depth Maps for Image-Based Rendering”
- [14] www.data-compression.info/JuergenAbel/Preprints/Preprint_After_BWT_Stages.pdf

