

Sugárkövetéses programok időigényének becslése mintavételezés alapján

CSIKVÁRI ANDRÁS, JUHÁSZ SÁNDOR

Budapesti Műszaki és Gazdaságtudományi Egyetem, Automatizálási és Alkalmazott Informatikai Tanszék
juhasz.sandor@aut.bme.hu

Kulcsszavak: sugárkövetéses képgenerálás, teljesítménybecslés, mintavételezés, antialiasing

A számítógépek teljesítményének folyamatos növekedése következtében egyre gyakrabban alkalmazzák a nagy erőforrásigényű sugárkövetéses képgeneráló programokat jó minőségű képek, animációk készítésekor. A sugárkövetéses módszer szinte kompromisszummentes részletességet biztosít, hiszen a speciális effektusok (fénytörések, tükröződések és árnyékok) létrehozásához nincs szükség külön algoritmusokra. A sugárkövetés könnyen párhuzamosítható, így időigénye a futtató környezet kialakításával kompenzálható. A párhuzamos futtatókörnyezet méretezése a képszintézis idejének előzetes becslésén alapul. Az eddigi becselő módszerek kihasználják az objektumok tárolásának és a rajzolást végző programok sajátosságait, így használhatóságukat korlátozza a képet generáló program megfelelő ismeretének a hiánya. Javasolt módszerünk lényegesen kevesebb feltételezéssel él a programmal kapcsolatban, és a képet leíró forrásfájl felépítésének ismeretére nincs szükség.

1. Bevezetés

A számítási- és memóriaigény szempontjából a programok alapvetően két csoportra oszthatóak. Az általánosan használt, gyakori felhasználói beavatkozást igénylő programok, mint például a szövegszerkesztők, táblázatkezelők viszonylag kevés erőforrást igényelnek, ezzel szemben jól elkülöníthető egy olyan programcsoport, amely kevés külső beavatkozást kíván, viszont az előző csoporthoz képest nagyságrendekkel nagyobb erőforrásigénnyel rendelkezik. Ezeket a programokat általában nagy teljesítményű gépeken futtatják, szokszor fűrtözött rendszereket alkalmazva. Ide tartoznak többek között az adatbáziskezelő szoftverek, a webkişolgálók, és a fotó minőségű képeket előállító grafikus programok.

Az ilyen jellegű alkalmazások használata esetén a minőségi, idő- és költségkorlátok egyensúlyának beállítására szükségessé teszi az erőforrásigény megfelelő előzetes felmérését. A jelenleg a sugárkövetésre épülő képalkotás időigényének előzetes becslésére többféle módszert alkalmaznak. Ezek közös jellemzője, hogy nem használnak időigényes eljárásokat, ennek viszont az alacsonyabb becslési pontosság az ára, ráadásul a képet készítő program belső felépítésére is különféle megkötéseket tesz.

Célunk egy olyan új, mintavételezésre épülő, lineáris modell kialakítása, mely lehetővé teszi a különféle minőségben (felbontással) előállított képek erőforrás igényének előzetes meghatározását.

A cikk felépítése a következő: A 2. fejezet a kapcsolódó munkák áttekintésével foglalkozik, a 3. fejezetben bemutatjuk a javasolt lineáris modellt, áttekintve az alkalmazhatóság feltételeit és körülményeit. A 4. fejezet a becsült és mért adatokat hasonlítja össze, külön kiemelve a nem pontosan modellezett az élsimító (antialias) algoritmusok hatására.

2. Kapcsolódó munkák

A fotórealisztikus képalkotás időigényének felmérésére többféle módszert alkalmaznak. Ezek közül a leggyorsabb egy egyszerű, zárt alakban felírt formula alkalmazása lenne a képen található objektumok számának, elhelyezkedésének és/vagy méretének stb. figyelembevételével.

Cleary és Wyvill [1] által levezetett képlet szerint egy kép idő komplexitása elsősorban az objektumok méretétől, és nem a számától függ. Kiszámolták, hogy egy sugár mekkora valószínűséggel metszi azokat a térbeli cellákat, amelyek tartalmaznak egy adott objektumot. Ezt a modellt finomította MacDonald és Booth [2], valamint Whang [3].

A sugár útjának követésével, az átlagosan érintett cellák számának vizsgálatával foglalkozott Subramanian és Fussel [4], míg Reinhard, Kok és Jansen [5] egy általános, sugarankénti költségfüggvényt felállítására törekedtek.

Ezek a modellek mind különféle feltételezésekkel élnek a teret felosztó, optimalizációs célokat szolgáló cellákkal kapcsolatban, így az ilyen becslést alkalmazó programnak a forrásfájlból kell kinyernie szükséges adatokat. Az ehhez szükséges ráfordítás nem csupán a futtató időt növelheti jelentősen, hanem a becselő alkalmazás fejlesztése is jelenetős feladat. Használhatóságukat tovább rontja, hogy sorra jelennek meg a különböző objektumtípusokra alkalmazandó speciális technikák, amelyek alkalmazása jelentősen nehezíti a fenti becslések gyakorlati alkalmazását.

Jiménez, Segura és Feito [6] a bonyolultabb, konkrét alakzatokra is hatékony vizsgáló algoritmust vezet be, míg Lischinski és Gonczarowski [7] a paraméteres felületek hatékonyabb kezelésére törekszik. A különböző, sugárkövetésben alkalmazandó hibrid struktúrákkal Chang [8] munkája foglalkozik.

Javasolt megoldásunk mintavételezésre épül, így a fent említett módszerekkel szemben előnye, hogy a program vagy a forrásfájl felépítésétől függetlenül, megfelelően választott minták mellett pontos becslést ad a képek elkészítésének időigényére. A módszer használatához fontos meghatározni, hogy mi az a minimális felbontás, ami még mintának tekinthető, és milyen becslőfüggvényt illesszünk a kapott minták időigényére. Természetesen törekedni kell a mintavételezés költségeinek minimalizálására is.

3. A becslés

3.1. A sugárkövetés időigényének becslése

A sugárkövetéssel algoritmus alapgondolata, hogy szimulálja a fénysugarak útját, így határozva meg az egyes pontokban található színértéket. A felesleges sugarak szimulálásának elkerülése érdekében a sugarakat a szemből indítják, majd ellenőrzik, hogy találkozik-e az egyes objektumokkal. Az objektumoknál a fénysugár több újabb sugárra ágazhat szét (árnyék, törés, tükrözés), melyekre az eredeti sugárhoz hasonló vizsgálatokat kell elvégezni. A pixelenkénti rekurziók számát azonban általában korlátozzák, mivel bizonyos mélység fölött a létrehozott kép minősége nem javul lényegesen, ellenben jelentősen növekszik az algoritmus futási ideje. A kép minőségét gyakran úgy javítják, hogy a látható pixelekhez, a területük felosztásával, több sugarat is rendelnek, majd a kapott színértékeket átlagolják (supersampling).

A fenti leírás alapján a sugárkövetéssel programok elvi futási idő komplexitása

$$O(2^k n s o), \quad (1)$$

ahol n a látható pixelek száma, s a pixelenként indított sugarak, o a sugaranként vizsgált objektummetrészek átlagos száma, k a rekurzió átlagos mélysége.

A korai sugárkövető motorokban a pixelenként kiinduló sugarak számát (s) konstansként lehetett beállítani, erre ma is van lehetőség, de az időigény csökkentésének céljából napjainkban ezt a paramétert már általában adaptív módszerekkel határozzák meg, azaz az alkalmazás egy pixelnél csak akkor lő ki újabb sugarat, ha egy meghatározott különbségnél nagyobb az első eredmény távolsága valamelyik szomszédos pixel színétől. A $2^k o$ tag az egy sugárra jutó átlagos számítási komplexitást takarja, cikkünkben ezt a tényezőt kívánjuk mérésrel közelíteni.

3.2. Feltevések

Ahhoz, hogy a mintavételezett képek elkészítésének időigényét hatékonyan felhasználhassuk a becsléshez, el kell dönteni, hogy az (1) képletben található s , o és k értékek hogyan függnek a mintákban állítható n pixelszámtól. Az összefüggés nyilvánvalóan érvényes, hiszen kis felbontás (kis n) mellett előfordul, hogy a sugár nem kerül olyan területre, amely esetleg jelentősen befolyásolná s , o vagy k értékét.

A becslés során élünk azzal a feltételezéssel, hogy n növelésével o és k értéke nem változik lényegesen. Ennek magyarázata az, hogy a mintavételezést olyan, már kellően nagy n értékekre végezzük, ahol már a kép elég részletes ahhoz, hogy a sugarak ne hagyjanak ki fontos részleteket.

Sajnos ez a feltételezés, ahogy erre a cikk végén részletesebben is kitérünk, az s értékére nem érvényes, így a becslő függvényeket $s=1$ beállítás mellett alkalmaztuk (pixelenként egyetlen sugár indítása), ami képszintézist végző program supersampling (anitaliasing) funkciójának letiltásával jár.

3.3. Becslő függvény

A minták számával kapcsolatban erős kompromisszumot kell kötni, hiszen hiába határozzuk meg nagyon pontosan a kép elkészítéséhez szükséges időt, ha ez közel annyi időbe kerül, mint magának a képnek a kiszámítása. Nem érdemes túl kifelbontású mintákat sem használni, mivel ez ellentmondana a 3.2. pontban leírtaknak.

A kép komplexitása elvileg egyenesen arányos n -nel, mivel elvileg minden pixelen ugyanazt a művelet-sorozatot kell elvégezni, ugyanakkor a becslőfüggvény meghatározásakor a becslés költségének alacsony szinten tartása érdekében 1-2 mintánál többet nem engedhetünk meg. A fentiek miatt a renderelési időt egy két szabad paramétert tartalmazó lineáris függvénnyel becsüljük:

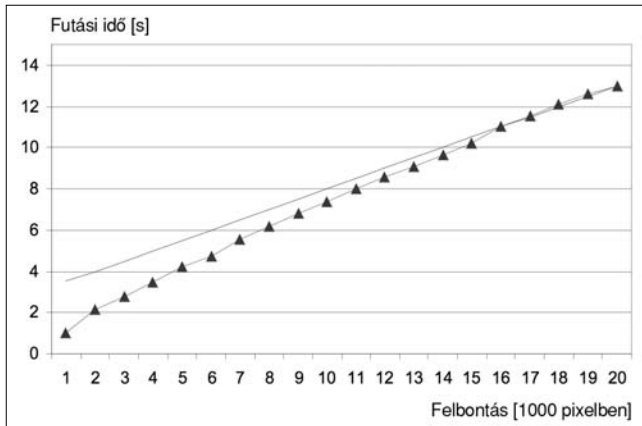
$$T(n) = an + b \quad (2)$$

Az alkalmazott $T(n)$ időfüggvény a és b paramétereinek a beállításához legalább két mintára van szükség. Az a értéke határozza meg az egy pixelre jutó időköltiséget, a b paraméter a forrásfájlanként változó konstans költségeket takarja. Ide tartozik többek között a fájl értelmezése, és az esetlegesen szükséges preprocessing műveletsor is (például a részecske rendszerekkel modellezett effektusok generálásához szükséges mátrixok előállítását).

Ahhoz, hogy a minták – 3.2. pontban leírt – felbontási feltételének eleget tegyünk, először meg kell vizsgálni, hogy mi az a minimális felbontás, ami alá nem érdemes menni, feltéve, hogy a becsülni kívánt kép 1024x768-as felbontással kerül elkészítésre. Ezt a legegyszerűbben úgy határozhatjuk meg, hogy a mintákat 1024x768 és 800x600 felbontásnál vesszük fel, így létrehozva az ideálisnak vélt egyenest majd vizsgáljuk a jósolt és az eredeti időigényt a kis felbontású képek tartományában. Nyilvánvaló, hogy a kis felbontású tartományban, ahol az eltérés nagy vagy erősen ingadozik, nem várható pontos becslés. Ennek oka, hogy az itt található, erősen minta függő viselkedésű pontokra illesztett egyenes jelentősen eltér a sok mérés alapján meghatározott ideális lineáris közelítéstől. Ezt szemlélteti az 1. ábra (a következő oldalon).

A mérések során elsőként a becslésekhez egységesen használt minták felbontását állapítjuk meg. A minták méretének kiválasztása ugyanúgy a becslési mű-

velet részét képezi, akárcsak a maga a becslőfüggvény. Második lépésként a minimális felbontás és a becslőfüggvény ismeretében végrehajtjuk a becsléseket, és megállapítjuk a hibák mértékét a mérési pontokban.



1. ábra
Egy kép lehetséges időigénye a felbontás függvényében. Túlságosan alacsony felbontáson az ideálistól való nagy eltérés miatt nem érdemes mintát venni.

4. Mérések

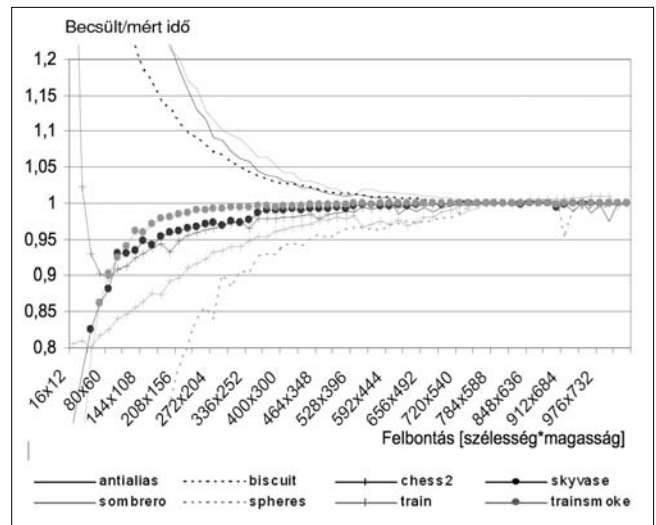
4.1. Élsimítás nélküli eredmények

Az általános következtetések levonása érdekében az időméréseket nyolc, lényegesen különböző képen (2. ábra) végeztük el, 16x12-es felbontástól kezdve 1024x768-as felbontásig. A tesztsorozatban szereplő képek viszonylag széles tartományt fednek le, a bennük található objektumok számának és anyagi minőségének tekintetében. A renderelő programként a Povray 3.5 parancssoros verzióját választottuk, a mérések során egy 2.26 GHz-es 256MB memóriával ellátott P4-es számítógépet használtunk, melyen Windows XP operációs rendszer futott.

A fentiek alapján minden kép 1024x768 és 800x600-as felbontású változatának időigényére egy egyenest illesztettünk, majd ezt ideális lineáris közelítésnek te-

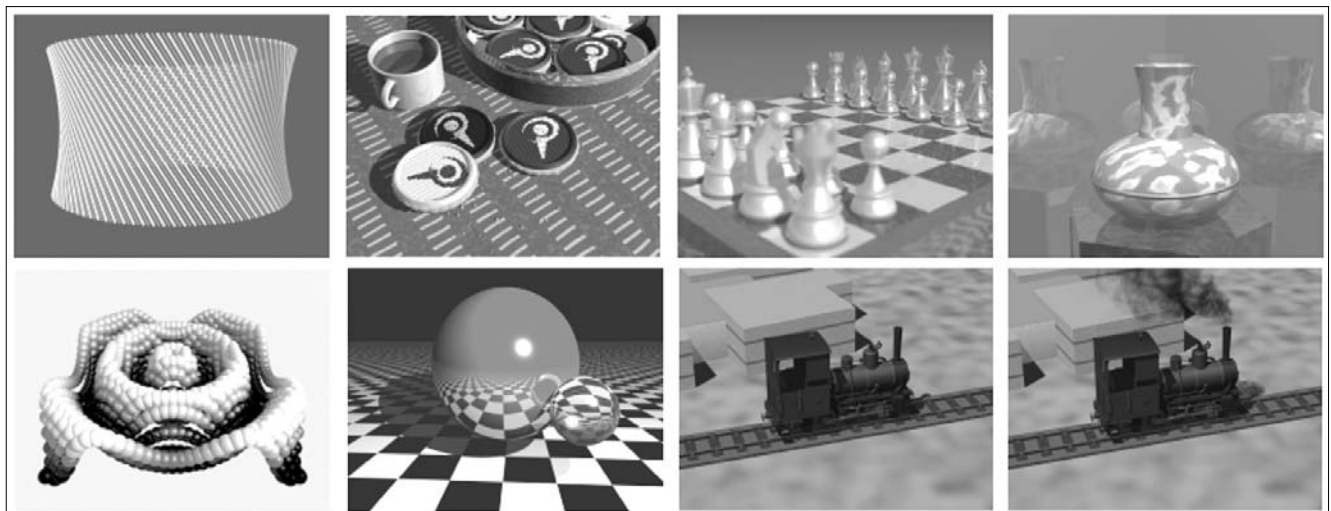
kintve minden mérési pontban kiszámítottuk a becsült (egyenes által adott) és a valódi időigény arányát. A kapott eredményt a 3. ábra szemlélteti. 224x168-as képméretnél már mind a nyolc kép esetén a kapott arány belül volt a 20%-os tartományon (0,8-1,2). Ez alatt a felbontás alatt azonban az eltérés már meredeken változik, azaz onnan biztosan nem érdemes mintát venni. Érdemes megfigyelni, hogy az ideális 1 arányt a görbék egy része felülről, más része alulról közelíti meg, azaz a mintavétel után a $T(n)$ függvény egyaránt becsülhet fölé és alá is a jelenet jellegétől függően.

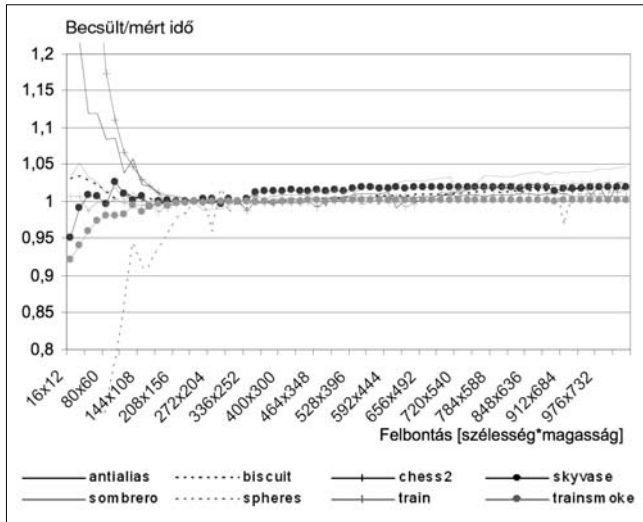
3. ábra A becsült és a valódi időigény aránya a felbontás függvényében a nyolc képre. Az ideális egyenes az 1024x768 és a 800x600-as felbontások időigényére illeszkedik



Nézzük meg, milyen eredményt ad a becslés, ha a mintákat az elfogadható két legkisebb, de egymástól már elég távol (majdnem kétszeres pixelszám) lévő 240x180-nál és 320x240-nél vesszük fel. Ez esetben 320x240 + 240x180 (120 ezer) pixelt számítottunk ki, ami kb.15%-a az 1024x768-as felbontás 786432 pixel-számának, azaz a becslés ideje nagy.

2. ábra A nyolc különféle tulajdonságú jelenet egy-egy képe: antialias, biscuit, chess2, skyvase, sombrero, spheres, train, trainsmoke



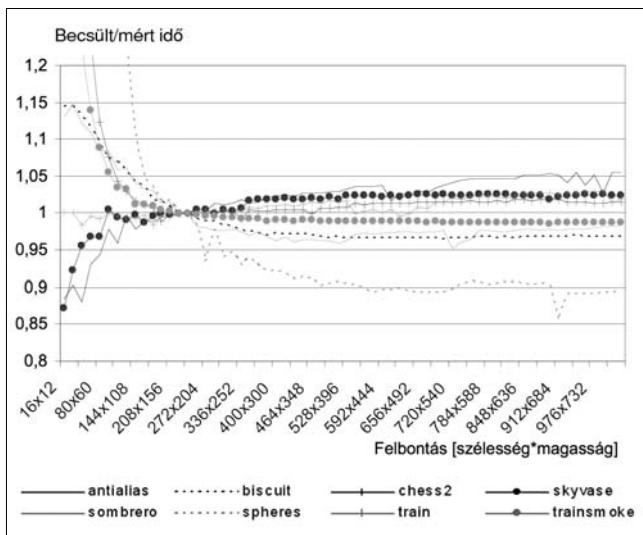


4. ábra

A becslés tévedésének a nagysága 5% alatt marad, ha a mintákat a 320x240 és 240x180-as felbontásnál vesszük fel.

A becslés relatív hibája a 4. ábrán követhető: a becslés pontosnak tekinthető, hiszen relatív hibája a mintáknál nagyobb felbontású esetekben is 5% alatt marad.

A mintákat nem érdemes egymáshoz túl közel felvenni, mert ekkor a helyi mérések zaja a becslő egyenes meredekségére (a) jóval erősebben hat. 240x180 és 224x168-as felbontású minták esetén (a két minta pixelszámának aránya csupán 1,15) összesen 80832 pixelt kell kiszámítani, ami csak kb.10%-a az 1024x768-as felbontás pixelszámának. Ennél kisebb méretű mintákat már nem érdemes használni, mivel a becslés pontossága erősen lecsökken (5. ábra).



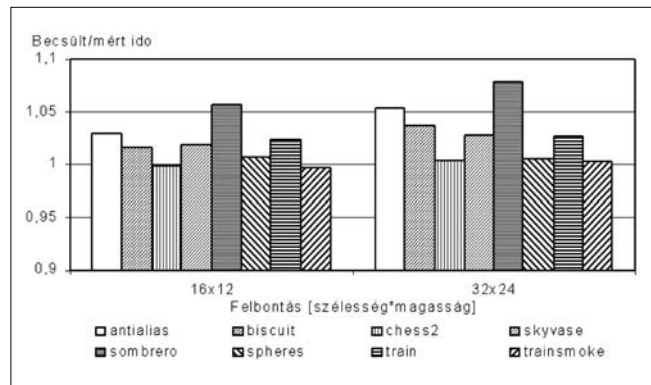
5. ábra

A becslés pontossága lényegesen rosszabb 240x180 és 224x168-as felbontású minták alkalmazása esetén.

A lineáris közelítés pontos eredményt szolgáltat nagyobb (például nyomtatott plakátok készítéséhez alkalmazható) felbontások esetén is, ráadásul a becslés az eredményhez viszonyítva relatíve kevesebb időt igényel.

nyel. A fenti állítás ellenőrzésére, megvizsgáltuk a becslő függvény viselkedését nagyobb 2048x1536 és 4096x3072-es felbontás használatakor.

A becslés által adott, meglepően jó eredményeket a 6. ábra szemlélteti: 2048x1536-os felbontásban csupán a sombrero jelenet lép ki az 5%-os hibahatárból, 4096x3072-es felbontásnál, bár csupán néhány tized százalékkal, de az antialias példa becslése is rosszabb 5%-nál, viszont az eltérés minden esetben 10% alatt van. A becslési idő és a végleges kép kiszámításának időaránya itt már 4%, illetve 1% alatt van.



6. ábra A 240x180-as és 320x240-es minták által adott becslés jól alkalmazható nagyobb felbontású képek készítésénél is

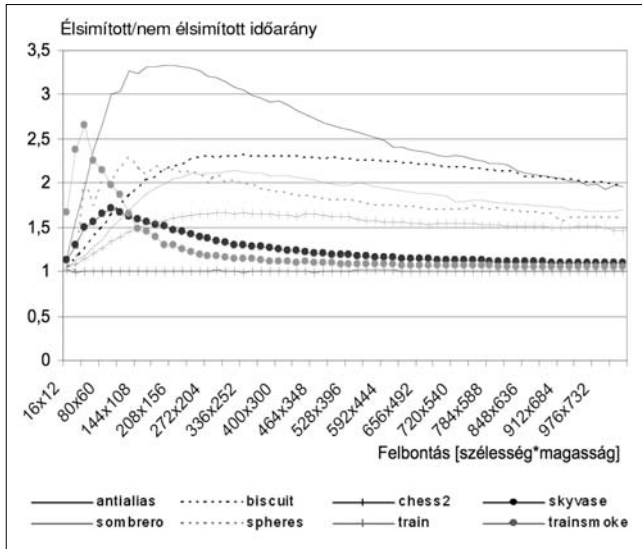
4.2. Az antialiasing hatása

Eddig az (1) képletben szereplő s paraméter értékét az adaptív antialiasing használatának letiltásával 1-re állítottuk. Ennek az indoka az volt, hogy nem az a feltetelezés, hogy s értéke n -től független lenne egy bizonyos méret felett, de a bizonyítás a 3.2. fejezetben még elmaradt.

A 3.3. fejezetben használt (2) lineáris becslőfüggvény hatékony alkalmazásával bebizonyosodott, hogy a sugárkövetéssel programok időigénye alap esetben tényleg arányos a renderelt pixelek számával. Ha tehát megmutatjuk, hogy egy képet antialiasing funkcióval és anélkül elkészítve a két időigény aránya a felbontással erősen változik, akkor ezzel belátjuk, hogy s értéke függ n értékétől még magasabb felbontásokon is.

Az adaptív élsimítás (antialiasing) lényege, hogy ha az éppen renderelt pixel színe jelentősen eltér valamely szomszéd pixel színétől, akkor a program az adott pixelhez kissé közelebb egy újabb sugarat lő ki. Ezután az új sugár által vizsgált színt is megvizsgálja és szükség esetén (rekurzívan) újabb sugarat számol ki. A pixel végső színe a sugarak által meghatározott színek súlyozott átlaga lesz. A programnak meg kell adni, hogy mekkora az a két szín közötti távolság, ahol már nem folytatja az iterációt. Az élsimított képek elkészítésénél a Povray motorban a színvektorok közötti távolságkülöbséget 0.3-ra állítottuk (+A0.3 +AM2 paraméterek).

A 7. ábrán (a következő oldalon) látható, hogy a görbék erős emelkedéssel kezdenek körülbelül addig, míg minden él láthatóvá nem válik, majd ahogy az éleket tartalmazó pixelek aránya egyre kisebb a teljes képhez



7. ábra
Az antialiasing funkcióval és anélkül készített képek időigényének aránya

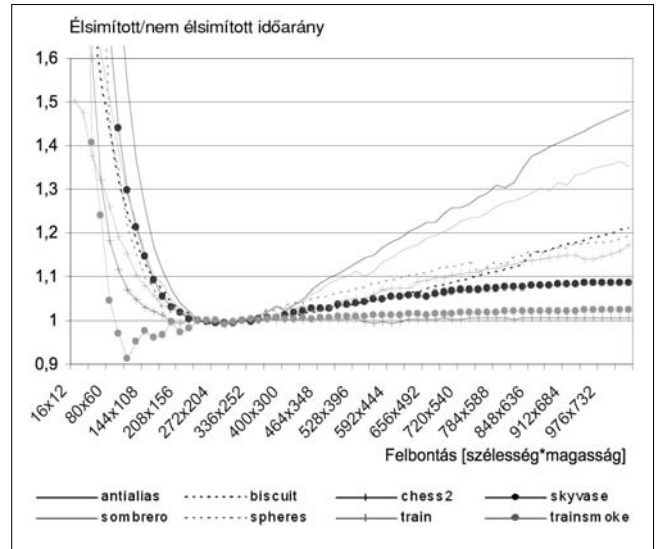
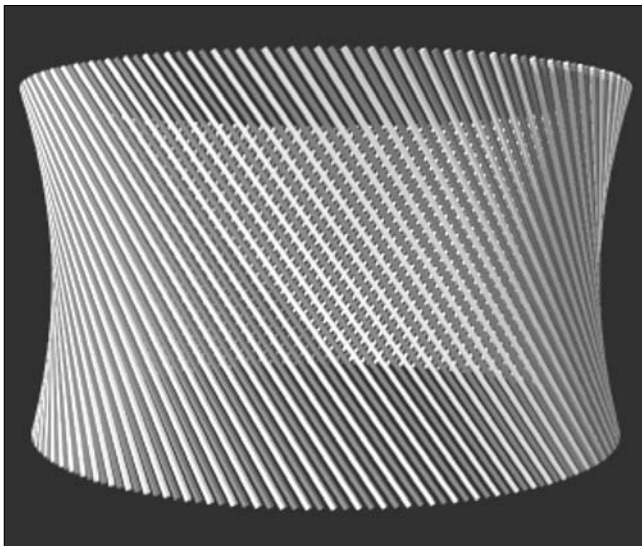
viszonyítva, úgy csökken az antialiasing funkcióval készített kép időigénye is a normál kép időigényéhez képest. Külön kiemelendő az antialias nevű kép (8. ábra), amelyet pontosan az élsimítás egy szélsőséges esetének bemutatására készült.

A másik véglelet a chess2 (9. ábra) kép jelenti, ahol egy elmosási hatás használata miatt, a két kép időigényének az aránya gyakorlatilag egy.

Önmagában tehát a (2) függvény csak korlátozottan alkalmazható az antialiasing funkció használatával készített képek időbecslésére.

A 10. ábra illusztrálja a hiba mértékét abban az esetben, ha az eddig használt 320x240 és 240x180-as felbontások segítségével végzünk becslést a renderelő program időigényére, miközben az adaptív élsimítási funkció be van kapcsolva.

8. ábra
Az antialias.pov fájlban 200 darab ferdén elhelyezett piros és zöld hasáb váltogatja egymást egy kör mentén



10. ábra
A hiba mértéke erősen nő nagy felbontású képeknél, ha az élsimítási funkció be volt kapcsolva

5. Összefoglalás

A sugárkövető motorok képeiken igen élethű fényhatásokat tudnak ugyan létrehozni, de nagyon erőforrásigényesek. Mivel az algoritmus könnyen párhuzamosítható, gyakran alkalmaznak klaszter rendszereket a feladat gyorsabb végrehajtására. A klaszter megfelelő méretének megállapításához szükség van egy hatékony, a képszintézist végző program belső működéséről lehetőleg minél kevesebbet feltételező futási időbecslő módszerre.

Cikkünkben egy új mintavételezésre épülő lineáris becslő függvényt mutattunk be, amely két, viszonylag alacsony felbontású minta alapján képes hatékonyan megbecsülni a különféle minőségű, a mintáknál akár nagyságrendekkel több pixelből álló képek várható időigényét. A módszer a gyakorlatban minőség és erőfor-

9. ábra
A chess2.pov fájlnál nincs lényeges eltérés a normál és az antialiasing változat futásideje között



rás igény hangolásakor, illetve a párhuzamos futtató környezet megfelelő méretezésénél alkalmazható.

Kiderült, hogy a modellillesztéshez felhasznált 320x240-es és 240x180-as felbontású minták a jelenettől függetlenül megfelelő pontosságot biztosítanak. Ez arra enged következtetni, hogy minimális felbontású mintának az a megfelelő, ahol már emberi mértékkel mérve is jól kivehető a kép.

Ennek alapján elvárható, hogy nagyobb felbontások, például plakátok esetében is alkalmazható legyen a fent említett két minta alapján történő közelítés, hiszen ilyenkor a készítő arra törekszik, hogy az távolról is jól kivehető legyen.

A mérések során a mintákban 4:3-as képarányt választottunk, de ezt sehol nem használtuk ki direkt módon, viszont fontos, hogy a minták képarányának azonosnak kell lennie a becsülni kívánt kép arányaival, különben az egyes pixelek időkölségei rossz súllyal fognak szerepelni a teljes időigényben, és így pontatlan értéket kapunk a $T(n)$ becslőfüggvény a paraméterére.

Láttunk azt is, hogy a lineáris becslés csak korlátozott pontossággal használható adaptív antialiasing alkalmazása esetén, mivel ilyenkor erősen túlbecsüli a valós időigényeket.

Irodalom

- [1] Cleary, J. G., Wyvill, G.:
Analysis of an algorithm for fast ray tracing using uniform space subdivision,
The Visual Computer (4), 1988, pp.65–83.
- [2] MacDonald, J. D., Booth, K. S.:
Heuristics for ray tracing using space subdivision,
The Visual Computer (6), 1990, pp.153–166.
- [3] Whang, K.-Y., Song, J.-W., Chang, J.-W., Kim, J.-Y., Cho, W.-S., Park, C.-M., Song, I.-Y, Octree-r:
An adaptive octree for efficient ray tracing,
IEEE Transactions on Visualiz. and Comp. Graphics, 1(4), 1995, pp.343–349.
- [4] Subramanian, K. R., Fussell, D. S.:
Automatic termination criteria for ray tracing hierarchies,
Graphics Interface '91, 1991, pp.93–100.
- [5] Erik Reinhard, Arjan J. F. Kok, Frederik W. Jansen:
Cost Prediction in Ray Tracing,
Rendering Techniques '96, 1996, pp.41–50.
- [6] J. J. Jiménez, R. J. Segura, F. R. Feito:
An Optimized Ray-Tracing for Complex Solids,
WSCG'2001, Plzen, Czech Rep., 2001, pp.16–19.
- [7] Daniel Lischinski, Jakob Gonczarowski:
Improved techniques for ray tracing parametric surfaces.
The Visual Computer: Int. Journal of Comp. Graphics 6(3), 1990, pp.134–152.
- [8] Allen Y. Chang:
A Survey of Geometric Data Structures for Ray Tracing,
Technical Report~TR-CIS-2001-06,
CIS Department, Polytechnic University, 2001.

Hírek

A számos világrekordot felállított **Solaris 10** teljesítménye és ár-teljesítmény aránya áttörést jelent a pénzügyi szolgáltatásoktól a távközlésig terjedő piaci szegmensekben tevékenykedő ügyfelek számára. Hálózatkezelése vagy a többszálás SPARC processzorokra és az x86 architektúrára való speciális optimalizálás következtében a Solaris 10 rekordsebességgel futtatja a vállalati alkalmazásokat a Sun Fire rendszereken és Sun Java munkaállomásokon.

A kétprocesszoros, 64 bites hardveren futó Java Virtual Machine (JVM) kategóriájában elért rekord a SPECjbb2000 sebességpróbán született, amely a JVM szoftver megvalósításának, az operációs rendszer teljesítményének, a rendszer processzorai és memóriája méretezhetőségének mércéje. A tesztet Sun Fire V20z kiszolgálón hajtották végre, amely egyike a cég által gyártott azon rendszereknek, melyekben a legújabb, 252-es modellszámú, Direct Connect architektúrájú AMD Opteron processzorok működnek. Az eredményeket az új Sun Studio 10 fordítóval érték el, amely a C, a C++ és a FORTRAN programnyelveken írt alkalmazások fejlesztésével is kiemelkedő teljesítményt nyújt.

Az **Oracle** az **Intel** fejlesztési és műszaki csoportjával közösen folytatja a rádiófrekvenciás azonosításon és a távérzékelésen alapuló megoldásainak optimalizálását. A közös szolgáltatásorientált vállalati keretrendszeren (Service-Oriented Enterprise, SOE) nyugvó modell gyorsabbá teheti e korszerű technológiák elfogadását és bevezetését. A RFID-hálózatokból származó adatok napi mennyisége megbízható adatkezelési képességeket és jelentős számítógépes feldolgozási kapacitást igényel. Az RFID-adatok karbantartását és összesítését egy EPCIS szabványú megoldás biztosítja, amely a közös Oracle-Intel alapú szolgáltatásorientált vállalati keretrendszerre épül.

Az Oracle egyik szakértő partnere, az **eSpirit Kft.** olyan dobozos szoftvert fejlesztett ki **OWB Name&Adress** néven, amely az Oracle adatfeltöltő eszközhöz funkcióit kiegészítve széles körű adattisztítási megoldást nyújt a felhasználóknak. A szoftver duplikátumkezelési képességeit egészíti ki az eSpirit által fejlesztett név- és címadat tisztító eljárásával. Az OWB Match/Merge duplikátumkeresési és konszolidációs megoldás, funkcióinak hatékonyabbá tétele érdekében azonban fontos a konszolidációba bevont adatok – tipikusan a név és címadatok – minőségének feljavítása is.

Az OWB Name&Adress működésének alapelve, hogy bármilyen szabad szerkezetben beírt nevet és címet elemeire bont szét, és az egyes elemeket a programba épített magyar nyelvű referencia szótárak alapján javítja, standardizálja. A Name&Adress szoftver kötegelte és online módon egyaránt alkalmas személynevek és címadatok javítására.