

Számítógépes grafika

V. rész: dinamikai effektusok

A 3d trükksoftverek – a nagy versenynek és a számítógépek fejlődésének köszönhetően – évről évre új, beépített effektusokkal bővülnek. A PC-ken futó programokban úgy 3 évvel ezelőtt jelentek meg az első dinamikai szimulációs modulok. Sorozatunk eme számában áttekintjük, hogy hol tart ez a technológia, és mire használhatóak ezek a mára már széles körben elterjedt eszközök.

A dinamikai szimuláció lényege, hogy képesek vagyunk – a számítógép segítségével – a fizika törvényeinek megfelelő mozgásokat, ütközéseket és egyéb kölcsönhatásokat modellezni. Hogy a valóság pontosan milyen hatásait és törvényeit vagyunk képesek szimulálni, azt mindig a konkrét program eszköztára határozza meg. Általában elhanyagoljuk a légellenállást, feltételezzük, hogy nem közelítjük meg a fénysebességet és az atomi méreteket stb. Létezik azonban sok célszoftver – ilyen pl. a *RealFlow* – mely még az anyagok viszkozitása és hőmérséklete közötti összefüggést is figyelembe veszi. A következőkben bemutató jelleggel áttekintjük, hogy milyen szimulációs eszközök állnak a rendelkezésünkre (az Alias | Wavefront Maya programja alapján) és ezeket mire használhatjuk a gyakorlatban.

Részecskék

A dinamikai effektusok talán leglátványosabb és legegyszerűbb fajtája az úgynevezett *részecskeszerendszer* – *particle system* – szimulációja. Ebben az esetben a fizika törvényeit apró, kiterjedés nélküli pontokra alkalmazzuk. Ez a gyakorlatban akár több ezer részecske „reptetését” is jelentheti. Mivel ezeknek a pontoknak nincsen kiterjedése – általában – nem is ütközhetnek egymással. (Az ütközések vizsgálata sem megoldhatatlan probléma, de nagyon erőforrás igényes számítás!) A részecskéket több módon is létrehozhatjuk:

A részecskéket több módon is létrehozhatjuk:

Sketch particles: A részecskéket egy ecsettel, azaz általában egérrel hozzuk létre. Beállíthatjuk, hogy mennyire sűrű legyen a részecske-felhő, amely a vonásaink hatására jön létre.

Emitter: Ha azt szeretnénk, hogy a részecskék automatikusan jöjjenek létre, emittert kell létrehoznunk. Az emitter a paraméterek függvényében „ontja” magából az újabb és újabb pontokat. Meglehetősen sokféle emitter létezik, a legfontosabbak a következők:

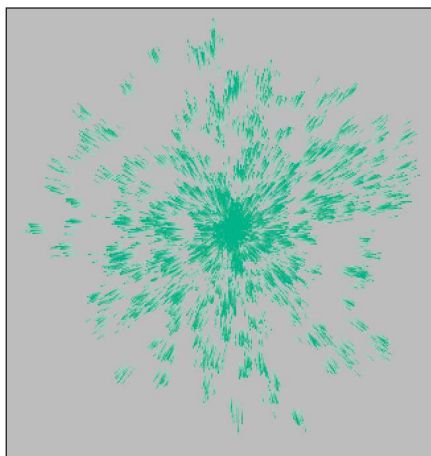
omni: a tér egy pontjából minden irányba indulnak szülő részecskék.

directional: szintén egy pontból, de fúvókaszerűen egy megadott irányba haladnak az új részecskék.

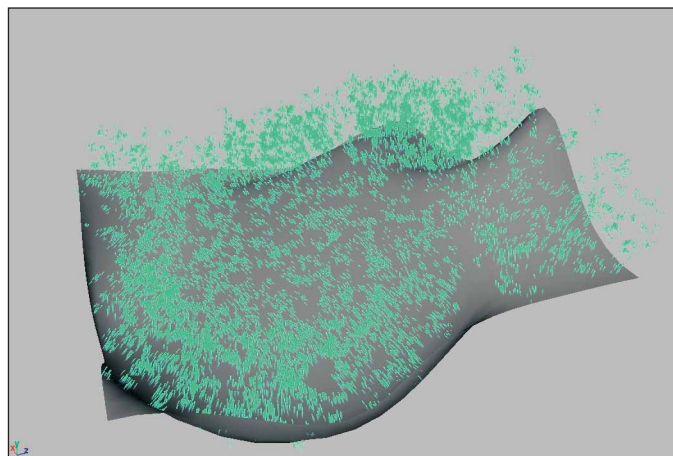
surface: lehetőségünk van egy felületen létrehozni az új részecskéket. Ez hasznos lehet, ha egy meglévő tárgyból akarunk részecskéket eregetni (pl. izzadást szimulálva), de használhatjuk kiterjedt részecskefelhő (pl. eső, hó) generálására is. Ez nehézkes lenne pontoszerű emittert használva.

volume: gyakran szükség van egy meghatározott formájú részecske-felhőre. Ha a volume emittert használjuk, akkor a részecske egy előre megadott testen belül fogak létrejönni, így akár ki is tölthetjük azt. Sajnos ez a funkció egyenlőre csak primitív formákra működik (téglatest, hasáb, tórusz stb.)

Ha létrejöttek a részecskék, mozgásukat a különböző erőhatások determinálják (l. később). A részecskéket a programok nem egyesével, hanem „csoportosan” kezelik. Ez azt jelenti, hogy a tulajdonságokat nem minden egyes pontra, hanem egy egész részecskeszereszerre adhatjuk meg. Egy ilyen rendszer rengeteg programfüggő tulajdonsággal, beállítással rendelkezik: energiamegmaradás, „mértéke”, cache használata, szimuláció pontosága, részecskék súlya, színe stb. Két fontos paraméter van, mely-

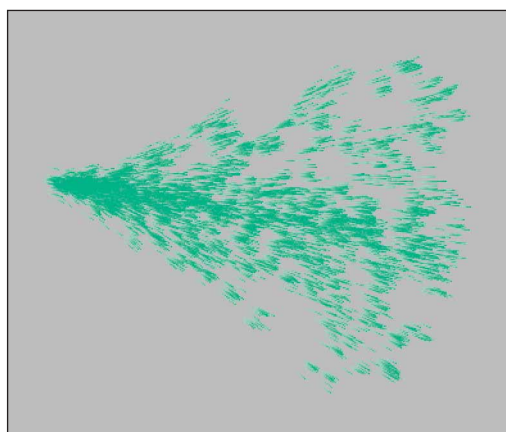


Részecskék omni emitterrel

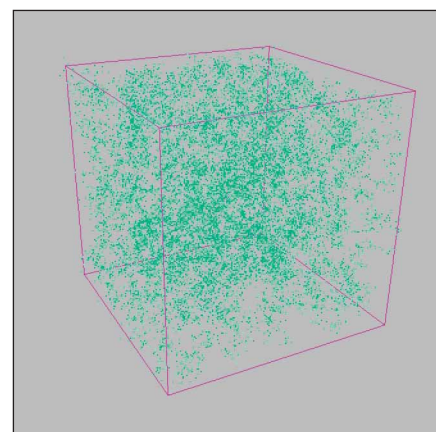


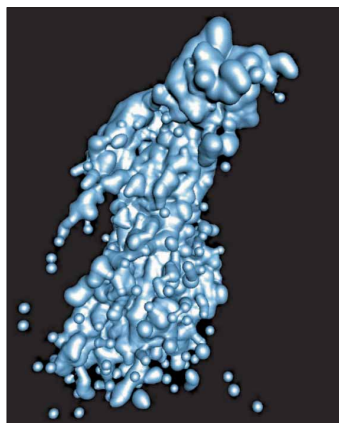
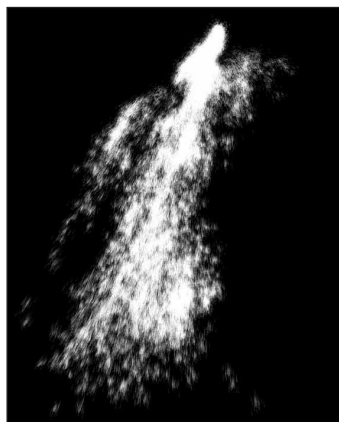
Részecskék surface emitterrel

Részecskék directional emitterrel



Részecskék volume emitterrel





Ugyanazok a részecskék különböző megjelenítési módokban: point, multi-streak, sphere, blobby

nek ismerete mindenképpen szükséges a legalapvetőbb trükkök készítéséhez is.

Az első paraméter a *részecskék életciklusát* (lifespan) hivatott állítani. Ha egy emitter folyamatosan hozza létre a részecskéket, akkor egy idő után elkerülhetetlenül megtelik a memória. Ezt hivatott megakadályozni a lifespan attributum, mely meghatározza, hogy a részecskék meddig maradjanak „életben”. Ha például egy havazó tájat készítünk, nem érdemes foglalkozni azokkal a hószemekkel, melyek már elhagyták a kamera látómezejét. Fontos azt is megjegyezni, hogy pl. a Maya tudja minden egyes részecskére, hogy az életciklusának melyik szakaszában van. Így ezzel a paraméterrel vezérelhetünk más tulajdonságokat is (szín, súly stb.). Ezzel a technikával egyszerűen megoldható, hogy a részecskék az „elmúlás” közeledtével elhalványuljanak vagy éppen kivilágosodjanak.

A részecske rendszerek másik fontos paramétere a *megjelenítésre* vonatkozik. Mivel a részecskék pontszerűek, ezért nem alkalmazhatóak a hagyományos (azaz felületeket megjelenítő) algoritmusok. Ha Mayát használunk, 7-féle speciális megjelenítési mód közül választhatunk. Ezen megjelenítési algoritmusok két csoportra oszthatóak:

Hardware render particles: ha ilyen megjelenítést választunk (pl. point, strike, multistrike, sprite...), akkor a részecskék megjelenítését a számítógépünk videokártyája végzi el. Ez azt jelenti, hogy nagyon gyors lesz a render, de utólag kell az így létrejött képeket kombinálni a hagyományosan renderelt szekvenciával. Ezt a megjelenítést egy dedikált ablakban (és nem a megszokott „render view”-ban) lehet elvégezni.

Software render: ezek a megjelenítési módok a részecskék alapján egy felületet (illetve testet) hoznak létre. Ez úgy lehetséges, hogy a pontokat gömböknek tekinti a program, és a közeli részecskék egybeolvadnak. Ez a megjelenítés nem igényli semmilyen speciális ablak használatát, és utólagos kompozitálásra sincs szükség. Ám sebesség szempontjából messze elmarad a hardware render megjelenítés gyorsaságától.

OLYMPUS

Én tudok választani!

Ajándék!
E-10 256 MB Compact Flash
4 Megapixel
4x optikai zoom

Ajándék!
E-20P 256MB Compact Flash
5 Megapixel
4x optikai zoom

C-1
1,3 Megapixel
2x digitális zoom
F 2,8

C-100
1,3 Megapixel
2x digitális zoom

C-1ZOOM
1,3 Megapixel
2x digitális zoom

C-3020ZOOM
3,1 Megapixel
3x optikai zoom

C-700UltraZoom
2,1 Megapixel
10x optikai zoom
kompakt

C-40Zoom
4,1 Megapixel
2,8 x optikai zoom+
2,5 digitális zoom

C-4040ZOOM
4 Megapixel
3x optikai zoom



OLYMPUS
Digitális mintabolt

axico
www.axico.hu
1074 Budapest, Dohány u.67. digitofoto@axico.hu
Tel:342-3255, Fax:351-2576



PINNACLE Pro-ONE

Professzionális
videoszerkesztő/készítő
rendszer



-Video, felirat, átmenet, grafika,
3D objektum, állókép

-Hollywood Fx átmenetek

-Képkorrekció, forgatás, nagyítás,
kép a képen

Mindez valós időben !

Átértékeltük a valós-idejű videoszerkesztés fogalmát!

Analog és DV feldolgozás, MPEG2 export, DVD-készítés, Adobe Premiere 6,
Külső csatlakozódoboz, valós-idejű szerkesztés, többcsatornás zeneszerkesztő

Hivatalos disztribútor

axico
www.axico.hu
1074 Budapest, Dohány u.67. sales@axico.hu
Tel:342-3255, Fax:351-2576



7 Emmy-díj



PINNACLE
SYSTEMS

Szilárd testek

Gyakori igény, hogy a virtuális tárgyak *ütközését, mozgását* szimuláljuk valósághűen. Erre alkalmas az úgynevezett *szilárd test* (rigid body) dinamika. Ebben az esetben a már kész modelleket (legyenek akár polygon akár NURBS felületek) tudjuk átalakítani „fizikai tárgyakká”.

Az, hogy egy tárgyat átalakítottunk szilárd testté, nem változtat semmit a formáján, csupán tudatjuk a számítógéppel, hogy mely testeket vegye figyelembe a dinamikai számításnál. Alapvetően kétféle szilárd test létezik:

Aktív (active rigid body): ebben az esetben a számítógép szimulálja a tárgy mozgását a fizikai összefüggések alapján. Ilyenkor nem tudjuk animálni a test helyzetét.

Passzív (passive rigid body): ilyenkor a test passzív, azaz hat másokra, de őrá nem hat semmi. Így elkerülhető, hogy a földön pattogó labda a földet is eltuszkolja a helyéről. Persze nem csak az egyhelyben álló testek esetén hasznos ez a funkció, hanem az animált tárgyak esetében is. Ha pl. egy dáköval akarjuk a biliárdgolyókat meglokní, akkor a dákö objektumot passzív szilárd testként kell kezelni, hiszen mi akarjuk a mozgását animálni, de nem árt, ha ütközik vele a többi szilárd test. (Mondanom sem kell, hogy magának a biliárdasztalnak is passzívnak kell lennie!)

A szilárd testekre is ugyanazokkal az erőhatásokkal hathatunk, mint a részecskékre, ezekről a hatásokról később szólnunk. Sőt az is nagyon hasznos, hogy a testekkel ütközhetnek a részecskék, így komplex trükköket tudunk megvalósítani. Nem okoz gondot például egy vízimalom elkészítése: a vizet sok részecskével szimuláljuk, ezek pedig ütköznek, azaz meghajtanak egy lapátkereket. Ez a kerék akár további fogaskereket is hajthat.

Puha testek, ruha

Volt szó részecskerendszerekről – ezekkel folyadékokat, gőzöket, porokat tudunk szimulálni, a szilárd testekkel a kemény tárgyak mozgását modellezzük, de nem esett szó a *puha anyagokról, felületekről*. Ezek szimulációja sajnos kicsit nehezekebb, mint a fenti esetek, de messze nem megoldhatatlan. A legtipikusabb „puha” felület a ruha. Sajnos ennek modellezését nem igazán lehet megkerülni, hiszen a nap szinte minden pillanatában találkozunk vele.

A puha testeket (soft body) a következő trükkel modellezzik a programok: minden felületet pontok határoznak meg (lehetnek vertexek, kontroll pontok stb.). Ha ezeket részecskéknak tekintjük, és valamiféle „rugórendszert” rakunk közéjük, akkor nem csinálunk butaságot, hiszen a valóságban is így működnek a puha testek. A részecskék (atomok, molekulák, rostok stb.) között nem túl merev kötés van, így bizonyos mértékig torzulhat a test. Pontosan ezt csinálja a Maya is: a felületet meghatározó pontok-

Egy puha és egy merev test ütközése. A deformációt és a visszapattanást a program automatikusan kiszámolja

ból részecskéket csinál, és ezek közé egy rugó-hálózatot (spring) feszít. Mivel a valóságban a részecskék megpróbálnak a helyükön maradni (a gumilabda is visszanyeri pattanás után az eredeti formáját), ezt érdemes a virtuális részecskékre megadni. A Mayában az úgynevezett goal megadása szolgál erre a célra: ha a puha testnek az eredeti (torzítatlan) formáját megadjuk mint goal-t, akkor a részecskék törekednek a torzítatlan helyükön maradni. Ezzel a technikával kiváló eredményeket érhetünk el például hajas ember, zászoló vagy pudding modellezésekor.

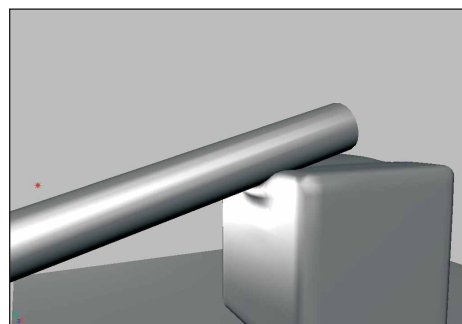
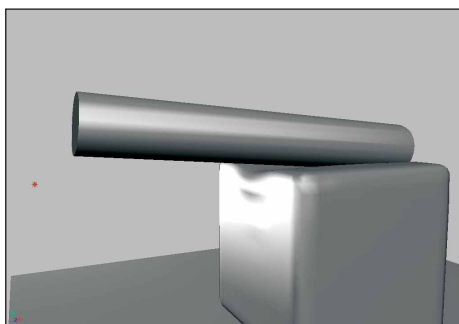
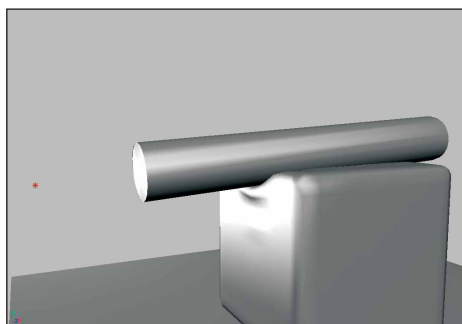
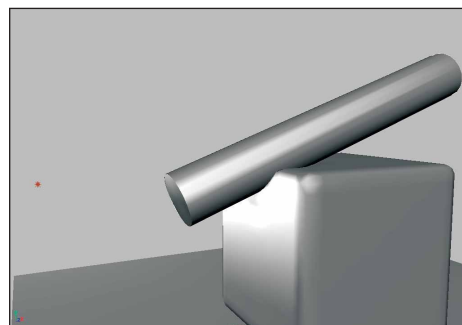
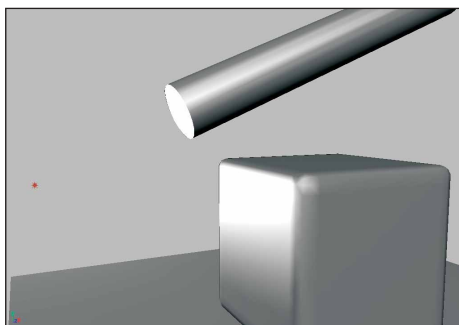
Mivel ez a technika a korábban leírt részecskedynamikát használja, így a puha testekre is hathatunk az erőhatásokkal, illetve ütközhetnek a többi szilárd testtel. Egy lelökött kávéskanál például ráeshet az asztalon lévő gyümölcskoscsnyára, ami meglepően realisztikusan fog remegni. Meglehetősen sok próbálgatást igényel egy puha test valósághű szimulációja, hiszen a rugóknak, az erőhatásoknak sőt még a szimuláció pontosságának is rengeteg (és nem elhanyagolható) paramétere van.

A ruha szimulációja is megoldható a „soft body” alkalmazásával, de a Maya rendelkezik egy specializált modullal (cloth) erre a célra. Az alapelv ugyanaz (részecskék és közöttük rugók), de a ruha elkészítése és paraméterezése illeszkedik a valósághoz: szövetdarabokat kell kiszabnunk és összevarrunk. A kész ruha paramétere is (amelyek igazából a rugókra vonatkoznak) beszédesebb: vastagság, rugalmasság, keménység, gyűrődés stb. Az így létrehozott ruha nehezebben illeszthető a dinamikai rendszerbe (az erőhatások, ütközések egy kicsit körülményesebbek), de cserébe nagyon gyors, szép és megbízható lesz a szimuláció. Persze így is előfordul, hogy néhány helyen a ruha „beleesik” az alatta lévő testbe, de az ilyen hibák könnyen kijavíthatóak a paraméterek okos beállításával.

Erőhatások

Sokat emlegettük a részecskékre és a szilárd testekre ható erőhatásokat, más néven *mezőket* (fields). Ezekkel a mezőkkel tudjuk „terelgetni” a részecskéket és testeket, valamint modellezhetjük a valóságban fellépő (erő)hatásokat. Az alábbi alapvető mezők szintén az összes programban a rendelkezésünkre állnak:

Gravity field: a fizikai gravitációhoz nem sok köze van, sokkal inkább a „mindennapok gravitációját” reprodukálja. Gyakorlatilag egy lefelé mutató konstans erőhatást fejt ki, amelyet tetszőlegesen paraméterezhetünk.



Turbulence field: nagyon hasznos eszköz, a térben minden irányba változó, kaotikus erőt hoz létre. Ha egy sűrű részecske-rendszerre hatunk vele, a viharos szél hatását kelthetjük.

Vortex field: ezzel a mezővel egy „forgószelet” tudunk létrehozni, amely egy megadott tengely körül forog. Mivel – mint azt fizikai tanulmányainkból tudjuk – ez a forgató erő a testeket a forgás középpontjától kifelé repíti. Ha azt szeretnénk, hogy az alanyok körpályán mozogjanak (vagy lassabban szóródjanak szét), akkor biztosítani kell valamilyen „benntartó” (centripetális) erőt.

Radial field: ez a mező éppen alkalmas az imént említett feladatra, ugyanis egy középpont felé (vagy attól kifelé) vonzza a részecskéket/testeket.

Ezekon kívül még számtalan erőhatás lehet az általunk használt program palettáján: „igazi” gravitációs erő, lassító erő a légellenállás szimulálására stb. Vannak bizonyos erőhatások, melyek nem egy mező formájában jönnek létre. Ilyen például a súrlódás, illetve az ütközésekkor létrejövő rugalmas erő.

A dinamikai szimulációkban minden esetben figyelembe vesszük a programok a fenti tényezőket, így lehetőségünk van mindig megadni a következő fontos paramétereket: tapadási, ill. csúszási súrlódás erőssége, ütközéskor az energia elnyelésének mértéke stb.

Fontos megjegyezni, hogy lehetőségünk van a szilárd testek között sajátos fizikai kapcsolatokat (constraint-eket) létrehozni. Ezekkel tudunk komplex, testekből álló rendszereket összerakni. A legfontosabb összekötések közül néhány:

- Oda tudjuk szögezni a testet a tér egy fix pontjához vagy egy másik testhez.
- Ki tudunk jelölni egy fix tengelyt úgy, hogy a test csak a tengely körül lesz képes forogni (fogaskerekék készítésénél elengedhetetlen).
- Két testet egy csuklóval össze tudunk kötni (nyakláncot tudunk csinálni könnyen).

A gyakorlat

Az előbbieken áttekintettük, hogy milyen alapvető eszközök állnak rendelkezésünkre a programok dinamikai moduljában. De mire tudjuk ezeket használni?

Sok olyan feladat van, amiről nyilvánvaló, hogy dinamikai szimulációval kell megoldani: *eső létrehozása, ruha készítése, homokvihar, csobbanó víz* stb.

A valóságban azonban rengeteg olyan effekt elkészítésénél használják a részecske-rendszereket, melyről nem is feltételeznék a gyanútlan néző. Egy tipikus ilyen alkalmazás az *embertömeg, állatvagy egyéb tömeg létrehozása*. Könnyen meg lehet ugyanis oldani, hogy a részecskék helyére egyéb képek vagy figurák kerüljenek. Így egy kellően kaotikus részecske-rendszer minden pontját egy lovas katonára cserélve szép, nagy sereget kaphatunk. A magyar mozinézők a *Sacra Coronában* már láthattak így készült tömeget.

A részecskék helyettesítése más tárgyakkal (particle replacement) látványos trükköket eredményezhet. A tömegeken túl halra-



A Sacra Corona című filmben részecskék segítségével készült a lovas tömeg. Minden pont helyére az eredeti felvételtől kiemelt katona került

jokat, hulló rózsaszirmokat vagy éppen a felszín felé emelkedő buborékokat készíthetünk.

A dinamikai szimulációk használata sokat segíthet bármely számítógépes animáció élethűvé tételéhez. Sajnos a kezdő 3d grafikusok azt hiszik, hogy a szép arcok és a szörnyek modellezése jelenti a 3d grafika lényegét. A dinamikai rendszerek (és egyéb technikai jellegű feladatok) legalább annyira fontosak a jó animáció elkészítéséhez, mint a modellezés vagy textúrázás. Javasolom mindenkinek, hogy legalább alapszinten ismerkedjen meg a használt program ilyen jellegű szolgáltatásaival, hiszen gyakran kisebb energia befektetésével látványosabb dolgot érhetünk el, mintha valamilyen csudaszép 3d modellt akarnánk elkészíteni. Kezdetben érdemes próbálkozni egyszerűbb trükkökkel (eső, hó, homok, tekebábuk stb.) és csak később belevágni komolyabb feladatokba (remegő izmok az animált figurán, menekülő halraj stb.). Jó szórakozást!

Vass Gergely

Magyar Filmlaboratórium
gergely_vass@siggraph.org

A VideoPraktika magazin új reklámfilmjében részecskéket használtunk a TV képernyőjén látható fehér zaj elkészítéséhez

