



Megvilágításmodellezés a Radiance-szal

Vázlattól a leképezett képig – néhány óra alatt.

Amikor egy rönkházat szerettem volna megtervezni a számítógépen, hogy lássam, milyen látványt nyújt majd bizonyos fényviszonyok között, először az AutoCAD-del, a 3D Studio Maxszal és számos egyéb otthontervező kereskedelmi termékkel próbálkoztam. Egyik sem nyújtott valóságosnak ható kimeneti képet, vagy az általam elképzelt rönkfalak kezeléséhez valamilyen könnyen használható módszert. Ezután eljártam a Lawrence Berkeley National Laboratory (LBL) Radiance nevű világításszimulációs programjával, és arra a megállapításra jutottam, hogy ha segédprogramokkal egészítem ki, ezzel sokkal gyorsabban érek célba.

De mi is az a Radiance?

A Radiance egy eredetileg *Greg Ward Larson* által írt megvilágításszimulációs program. Már a kilencvenes évek elejétől hozzáférhető, és a nem üzleti felhasználók számára adott ingyenességet mostanában váltotta fel a nyílt forrású felhasználási szerződés. A programcsomag az előállított igen látványos képeket egy különleges formátumban tárolja, ami egyaránt rögzíti az elrendezés felületi mintáit és a megvilágítás körülményeit, az Autodesk LightScape és VIZ 4 profi felhasználásra készült programjaihoz hasonlóan. A film- és játékkészítésben használt programcsomagok a gyorsétermekben kapható ételekhez hasonlíthatók: a végeredmény lehet vonzó és népszerű, de a valósághoz nem sok köze van. A film- és játékkészítők számára a világítás fizikai részletei egyszerűen nem olyan fontosak, mint a sebesség, hiszen nagyon sok képpont gyors kezelését kell megvalósítani: egy kétórás film 172 800 képkockából áll, a játékoknak pedig valós időben kell futniuk. Az eredmény az, hogy a grafikus rendszerek által előállított kép inkább egy művészi algoritmus eredménye, ami kevéssé támaszkodik valóságos alapokra.

A Radiance kimenete a fizikai fény olyan laboratóriumi hűségű utánpótlása (már amennyire a bemenet valóságos), ami a szakmai közönség szigorú próbáin megállta a helyét.

A Radiance telepítése

A Radiance forráskódja a <http://radsite.lbl.gov> címről szerezhető be. A forráskód letöltését javaslom az előfordított RPM-csomagok helyett, ugyanis az RPM-ek a bővítményeket magukban foglaló állományok egyikét sem tartalmazzák. Ha sikerült letöltenünk a tarcsomagot, teendők a következők:

```
$ tar xzf rad3R4.tar.gz
$ cd ray
$ ./makeall install
```

Ezután válaszoljunk arra a kérdésre, hogy hova szeretnénk elhelyezni a programot. Én a `$HOME/radiance/bin` könyvtárat használtam a bináris állományok, és a `$HOME/radiance/lib` könyvtárat a bővítmények tárolására.

A `makeall` parancsfájl nem telepíti a leírást és a példaképeket, ezek számára is nekünk kell valami alkalmas helyet találnunk, például:

```
$ mv doc/man $HOME/radiance
$ mv obj $HOME/radiance
```

Gondoskodjunk róla, hogy ezek az elérési útvonalak szerepeljenek a saját profilunk `MANPATH` és `PATH` változóiban. Egy figyelmeztetés: létezik a csomagban egy fontos segédprogram, amit `rview`-nak hívnak. Sajnos a Vimben szintén van egy ugyanilyen nevű segédprogram, úgyhogy vagy meg kell változtatnunk a `PATH` változót, vagy a Vim `rview` programját kell átneveznünk. A Radiance segédprogramjának ne adjunk új nevet, mert a programcsomag más segédprogramjai közvetve hivatkoznak rá.

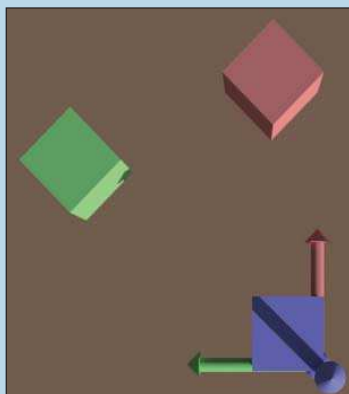
A Radiance bemenete

A Radiance új felhasználóinak először a jelenetek megalkotására szolgáló belső CAD-rendszer hiánya szúr szemet. A programot a 90-es évek elején kutatási céllal írták Unix alá, és ha a fájlformátumot megvizsgáljuk, az is azonnal nyilvánvalóvá válik, hogy a rendszer a hozzám hasonló, a csövezetek és a szöveges folyamatok hatékonyságától elragadtatott parancssormegszállottaknak készült (elvégre a monogramom is AWK). Mindamellelt rendelkezésre állnak azok a segédprogramok, amelyekkel a DXF, Wavefront, MGF vagy az ezekhez hasonló formátumú geometriai leírásokat lefordíthatjuk, így bármilyen programot használhatunk, ami képes az ilyen formátumba történő mentésre. A <http://linux.org> címen felsorolt modellezőprogramok jelentős része képes ilyen kimenet előállítására. A Desktop Radiance nevű Windows alapú AutoCAD/Radiance modul a Radiance honlapjáról is letölthető, ha éppen az AutoCAD megfelelő változatával rendelkezünk.

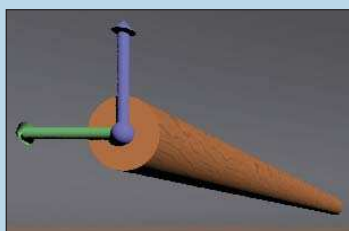
A Radiance bemeneti fájljai olvasható szövegek, amelyek remek lehetőséget nyújtanak parancsfájlok létrehozására. De legyünk óvatosak: a leírás helyenként olyan kifejezéseket tartalmaz, amiktől a fizikusok vérnyomása a magasba szökken, mint például a „watt per négyzetméter per térszög”. Nem árt a honlapon lévő összes leírást áttanulmányozni. Ha nem csak játszadoxni szeretnénk a programmal, talán érdemes beszerezni Greg Ward Larson „Rendering With Radiance” (Fényleképezés a Radiance használatával) című könyvét, vagy valami hasonlót. A könyv pillanatnyilag nem kapható, az antikváriumokban érdemes utánanézni (amerikai adat – a szerk.).

Saját próbálkozások: egy mintajelenet megalkotása

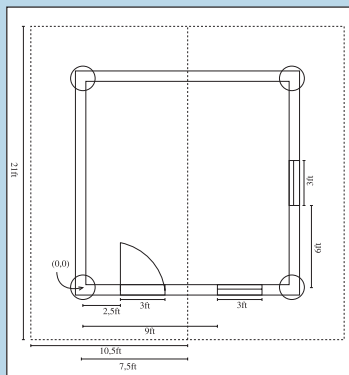
Az 1. lista egy eget és földet ábrázoló kép, amin az anyag beállítására rész, és ebből a részből egy gömb helyezkedik el a képen. Az ég és a föld nincs külön beállítva. Az egyetlen, amit a saját képhez meg kell változtatnunk, azok a `gensky` parancs beállításai. A listában lévő értékek november 25 déli időpontjának felelnek meg az északi szélesség 25. és nyugati hosszúság 80 fokán. A déli és keleti elhelyezkedést negatív értékekkel adhatjuk meg. Az első sor a tételre vonatkozó, már létező anyagot határozza meg (illetve üres részt, ha nincs megadva), az anyag vagy geometriai alakzat fajtájának a nevét (sphere – gömb, polygon – sokszög, plastic – műanyag, metal – fém),



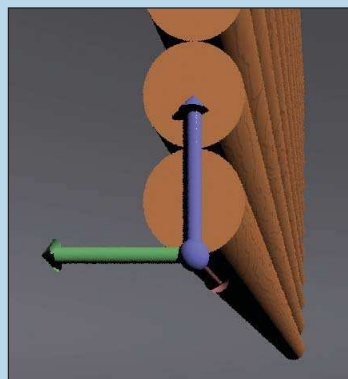
1. kép



2. kép



3. kép



4. kép

1. kép

Az xform viselkedése. A koordinátatengelyek színkóddal rendelkeznek: +x = vörös, +y = zöld, +z = kék. A kék kockát változtatlanul hagytam. A piros kockát elforgattam, majd eltoltam. A zöld kockát először eltoltam, ezután forgattam

2. kép

A genlog módosítatlan kimenete. A tengelyek VZK/xyz sorrendűek (V = +x, Z = +y, K = +z), az x tengely a rönk belsejében van

3. kép

Egy háziko alaprajza. A nyílások méreteit annak a falnak a kezdőpontjaitól kell megadni, amelyen elhelyezkednek

4. kép

A genlogwall kimeneteként születt meg, nem átalakított fal. A tengelyek színsorrendje: VZK/xyz

5. kép

A háziko nappali perspektivikus képe, néhány fával az érdekesség kedvéért. A kiadáshoz szükséges 3300x2200 minőségű leképezés egy 1,7 GHz-es noteszgépen körülbelül öt órát vett igénybe

6. kép

A háziko belsejének nagy látószögű képe külső napfény-megvilágítással. A kiadáshoz szükséges 3300x2200 minőségű leképezés egy 1,7 GHz-es noteszgépen körülbelül öt órát vett igénybe



5. kép



6. kép

© Kiskapu Kft. Minden jog fenntartva

és egy, a felhasználó által megadott nevet. A következő három csoport a tétel karakterlánc egész és valós (lebegőpontos) típusú értékei. Minden sor az értékek számával kezdődik, ezt követik maguk az értékek, amelyek annyi sort foglalhatnak el, amennyi szükséges.

A legtöbb bejegyzés csak valós értékkel bír. Ez a magyarázata a bejegyzések közepén gyakran előforduló két nullának: nincs karakterlánc vagy egész típusú érték. A réz (brass) utolsó sorában lévő 5-ös szám azt jelzi, hogy öt valós érték következik, a gömb (sphere) utolsó sorában lévő 4-es pedig négy valós értéket jelez. Az értékek meghatározott sorrendben követik egymást, például a gömb megadásához egy középpontra (x; y; z) és a sugárra van szükség.

Gyakran az anyagok jelentik az elrendezés legnehezebb feladatát. A legegyszerűbb, ha már létező anyagokat másolunk, és ezeket módosítjuk az igényeinknek megfelelően. Erről további részleteket a honlapon elérhető [refman.pdf](#) fájlban találhatóunk. Az 1. lista (49. CD Magazin/Radiance könyvtár) tetején lévő gensky egy beágyazott parancssoros segédprogram. A sor elején lévő felkiáltójel arra utasítja a rendszert, hogy a sort egy

olyan héjparancsnak tekintse, aminek a kimenete a kép részét alkotja. A Radiance számos ilyen segédprogramot tartalmaz, és tapasztalatom szerint a sajátjaink megírásával jóval egyszerűbbé és gyorsabbá válik a jelenetek megalkotása.

Az elemek mozgatása

A legtöbb jelenetszerkesztő program az elemeket a kezdőpontban helyezi el, ami valószínűleg nem egyezik meg a kívánt helyzettel. Ezen a gondon az xform segédprogrammal tudunk segíteni, aminek a parancsformátuma a következő:

```
xform -t transx transy transz
      -rx angle
      -ry angle -rz angle
      -s scalefactor tetszőleges_jelenetfájl
```

Az xform bármit fájlba tud alakítani, és lehetőségünk nyílik arra is, hogy ráírnyítsuk egy jelenetszerkesztő program kimenetét. A program alkalmas az objektumok átméretezésére (-s szorzótényező), tengely körüli forgatásra (-ry forgatási_szög az y

2. lista A nyílásleíró fájl (holes/southwall.holes) a déli fal számára

```
# Bejárati ajtó (padlótól 7 láb, kezdet:
# 2,5 láb a fal kezdőpontjától,
# 3 láb szélesség)
0 7 2.5 3:d
# ablak (3 lábtól 6 lábíg a padlótól,
# kezdőpont: 9 láb a faltól, 3 láb széles)
3 6 9 3:w
```

3. lista

A nyílásleíró fájl (holes/eastwall.holes) a keleti fal számára

```
# keleti fali ablak (3-6 láb a padlótól,
# kezdőpont: 6 láb a fal kezdőpontjától,
# 3 láb széles)
3 6 6 3:w
```

4. lista Adatok a tetőszerkesztő számára (roofdata)

```
0 0 p
21 0 p
21 10.5 b
0 10.5 b
0 0 p

# Szegélyek
c:3 0 3 7.5
c:18 0 18 7.5
```

tengely körüli forgatást jelent) és új helyzetbe való eltolásra ($-t\ x\ y\ z$ jelenti az $x\ y\ z$ távolságokkal való eltolást). A különböző lehetőségek többször is használhatóak az utasításon belül, tetszőleges sorrendben. A műveletek olyan sorrendben hajtódhatnak végre, ahogy a parancssorban megjelennek. Figyeljünk a jelenetszerkesztő program által használt alapértelmezett helyzetre. A legtöbbjük az origóban egy jelzősarkot helyez el. Az első ábrán a `genbox` által létrehozott kockákon látható az `xform` hatása. A képen a nézőpont a $+z$ tengelyen helyezkedik el, és az origó felé tekintünk. A piros tengely a $+x$, a zöld pedig a $+y$. A kék kocka a `genbox` hívásának eredeti eredményét mutatja. A piros kocka ugyanennek a `genbox` hívásnak az eredménye a következő `xform` műveletek után:

```
!genbox redplastic box1 .5 .5 .5 | xform -rz
45 -t 2 0 0
```

A zöld kocka pedig:

```
!genbox greenplastic box2 .5 .5 .5 | xform -t
2 0 0 -rz 45
```

Az anyagtípusok, mint a `redplastic` (piros műanyag), közvetlenül ezek előtt az utasítások előtt lettek meghatározva, de a listákon nem szerepelnek. Látható, hogy a műveletek egymásutánja hogyan befolyásolja a működést, és ezáltal a kimenetet.

Bonyolultabb jelenetek

Számos olyan jelenetszerkesztőt írtam Perlben, ami rönkkabinok és rönkházak összeállítására használható (ezek természetesen elektronikus formában is hozzáférhetőek). Ebben a cikkben ezek közül a `genlogwall`, `genlog` és `genroof` nevűt fogom alkalmazni. Kimeneteik mértékegysége a hüvelyk, még akkor is, ha a kényelmesebb használat érdekében a bemenet lábban van megadva. Az általam használt anyagok szintén megtalálhatóak az elektronikus terjesztésben.

A `genlog` segédprogram egy sapkás hengert fog eredményezni, aminek a középpontja a $+x$ tengelyen helyezkedik el (2. kép). Ehhez több értékre van szükség:

```
genlog anyagnév hossz_láb átmérő_hüvelyk
```

Az anyagmeghatározásnak már szerepelnie kell a jelenetben, a nevét magunk találjuk ki. Az általam használt előre meghatározott anyagok fájlja az egyes irányoknak megfelelően három faanyaggal rendelkezik: `xpine`, `ypine`, `zpine`. Olyan anyagot kell választanunk, amelyik illik a rönk végső illesztéséhez. Ha egy tíz láb magas, nyolc hüvelyk átmérőjű, a $+z$ irányba mutató póznát szeretnénk a (15ft; 0ft; 0ft) kezdőponton felállítani, akkor a következő parancsot kell kiadnunk:

```
!genlog zpine mypole 10 8 | xform -ry 90 -t
180 0 0
```

Ne felejtjük el, hogy az `xform` számára megfelelő mértékegységeket kell alkalmaznunk: 180 hüvelyk 15 lábnak felel meg. A `genlogwall` segédprogram a következő formában használható:

```
genlogwall anyagnév hossz_láb magasság_láb
rönkátmérő_hüvelyk [nyílás_adat_fájl]
```

A `nyílás_adat_fájl` nem kötelező érték, ebben adhatjuk meg, hogy milyen nyílásoknak kell a falban lenniük, és miket kell ezekbe a nyílásokba helyezni (például ajtót vagy ablakot). Most a könnyebbség kedvéért a 3. képen látható alaprajzról dolgozunk. Négy fal látható a képen, amelyek mindegyike 15 láb hosszú. A házikó délnyugati sarkát választottam (0;0;0) pontnak, az x értéket kelet felé, az y -t északi irányban növelve, a z növelésével pedig felfelé haladhatunk. Ez a tájolás dél felé fordítja a házat, s `gensky` szabályosan előállított égboltjának megfelelően. A `genlogwall` az előállított falat mindig a (0;0;0) pontból indulva az x tengelyre fekteti, ahogy az a 4. képen látható. A nyílások adatfájljának szerkezete is egyszerű, soronként egy nyílást írhatunk le benne:

```
nyílás_alja_láb nyílás_teteje_láb
nyíláskezdet_láb szélesség_láb[:w|d]
```

Az első két érték a padlótól számítandó, az utóbbi kettő a fal indítóélétől ($x=0$, x növekszik). A végén lévő nem kötelező címke azt jelzi, hogy ajtóval (`d`) vagy ablakkal (`w`) szeretnénk-e kitölteni nyílást. Tervem két ilyen nyílásleíró fájl hívását tartalmazza (lásd a 2. és 3. listákat). Többszörös falakhoz újra felhasználhatjuk ugyanazokat az adatokat, de csak akkor, ha a nyílásokat is ugyanúgy akarjuk megadni.

Házikónk utolsó művelete a tető elkészítése. Egy szokványos tető előállítás nem egyszerű dolog, ezért a `genrooftool` egy kicsivel jobban megoldoztatja a felhasználót, mint az eddigi műveletek.

A `genroof` a tető sík részeit készíti el; a teljes tetőt a művelet ismétlésével és az `xform` használatával állíthatjuk elő.

A `genroof` működéséhez egy adatfájlra van szükség a tetőrész sarokpontjainak x-y koordinátaival (lábban kifejezve), amiket az alaprajzról az élek mentén az óramutató jártásával ellentétes irányban olvasunk le. A sarokpontok mindegyikének az első (pozitív) síknegyedben kell lennie, a tetőéleknek pedig az x tengellyel párhuzamosan kell futniuk.

A pontok meghatározásához az alaprajzot úgy forgassuk, hogy a tető éle balról jobbra fusson. Most egy picit feledkezzünk el a tető alsó feléről, és a felső rész bal alsó sarkát tekintsük az új kezdőpontunknak. Ekkor a pontjaink: (0;0), (21;0), (21; 10,5), majd ismét (0;0). Eddig rendben.

A pontokat külön sorban kell felvinnünk, a koordinátákat szóközzel elválasztva, és a sor végén jelölve, hogy a pont a tető aljához (b), a közepéhez (mp) vagy a tetőélhez (p) tartozik-e. Ez azért szükséges, mert szabálytalan tetőrészek létrehozására is lehetőségünk nyílik.

Szükség van továbbá az oromszegélyek megadására, ha azt szeretnénk, hogy a `genroof` fával töltsse ki ezeket is. Ha a tető tájolása szerint nézünk az alaprajzunkra, látható, hogy a szegélyeknek a fal mentén a (3;0)→(3;7,5) és (18;0)→(18;7,5) koordináták szerint kell haladniuk. Adjuk ezt az információt is a tető adataihoz a `c`: előtaggal. A 4. lista mutatja a házikóhoz tartozó tető teljes adatfájlját.

A `genroof` indításához tartozó parancssor pedig a következő:

```
genroof -o túlnyúlás_láb típusnév név
tetőadatfájl magasság_láb vastagság_hüvelyk
```

A túlnyúlás értéke a segédprogram számára lehetővé teszi, hogy beállítsa a tetőrész helyzetét, így nyugodtan a fal magasságának megfelelő helyzetbe mozgathatjuk, nem kell a lejtés és a csatlakozás részleteivel foglalkoznunk. Az 5., már teljes házikóhoz tartozó listán látható, hogy a tetőrészek z irányú mozgásával a tető illeszkedik a fal magasságához, annak ellenére, hogy a túlnyúló rész a fal felső éle alá lóg. Mivel tetőnk szimmetrikus, ugyanazt a `genroof` utasítást egy másik `xform`-mal minden gond nélkül használhatjuk a másik tetőrész létrehozásához.

A jelenet megtekintése

A Radiance része egy `rad` nevű segédprogram, ami a Unixban használt `make` parancshoz hasonlóan működik. A `rad` bemeneti fájljában tárolt változók határozzák meg a jelenet leképezésének módját, és azt, hogy a fény szimulálásához szükséges sok egyéb program mi módon kerüljön meghívásra.

A 6. lista egy megjegyzésekkel ellátott példát mutat erre.

A legtöbb olyan változó, amihez fájlnev tartozik, annyiszor adható meg, ahány fájl a hozzárendeléshez rendelkezésre áll. A `view` változó szintén többször megadható. Minden egyes `view`-meghatározás egy újabb kép létrehozását eredményezi. Meg kell határozunk a nézet nevét, a nézőpont helyét (`-vp` nézőpont), az irányt, amerre nézni kívánunk (`-vd` a nézet irányának vektora), és a felfelé mutató irányt (`-vu` felfelé mutató vektor). Kedvelem a `-vt` beállítást is, amivel nagy látószögű (halszemoptikás) nézetet hozhatunk létre.

A különféle minőségi beállításoknál a `H` érték használata nagyon időigényes képelőállítást eredményez (több mint tíz óra egy 2 GHz-es gépen). A legtöbb esetben erre nincs is szükség, az `M` beállítás is remek eredményt ad. Az `L` az interaktív leképezéshez használható. A programleírás és egy kis kísérletegetés segíthet abban, hogy megtaláljuk a jelenetünkhöz legjobban illő beállítást.

Ha azonnali nézetet kívánunk, az alábbi parancsot használjuk:

```
$ rad -o x11 cabin.rif
```

A kép túl világosnak és fakónak tűnhet ebben a nézetben. Az `E` billentyű, majd ezt követő `ENTER` megnyomásával egy fényes pontra kattintva a képen javíthatjuk az expozíciós időt. Ezzel nem kell megvárunk a leképezés befejezését sem.

Az expozícióval annyiszor próbálkozhatunk, ahányszor csak akarunk. A Radiance képadatainak dinamikartománya messze meghaladja a monitorunkét, ami azt jelenti, hogy anélkül kaphatunk teljesen sötét vagy tiszta fehér képet, hogy vesztítsünk a kép adataiból. A képkalkotás ezen a téren teljesen eltér a normál képfájloktól, ahol a képet túl fényesre állítva maradandó adatvesztés lehet az eredmény.

Ha az interaktív leképezést használjuk, a `rif` fájlból az `L` parancssal más nézetet is betölthetünk. Ha például a `rif` fájlban van egy `interior` nevű belső nézetünk, az `L interior` begépelésével betölthetjük a nézetet. Saját kezűleg is felvehetünk egy nézetet a `V` billentyű és `ENTER` megnyomásával, csak a kért adatokat kell megadnunk. Az interaktív leképezésből a `Q` gomb és az `ENTER` megnyomásával léphetünk ki.

Az összes nézet képének előállítását a következő parancssal történhet:

```
$ rad cabin.rif
```

Ezt követően a megtekintéshez az alábbi parancsot használhatjuk:

```
$ ximage *.pic
```

Az `ximage` programban is beállíthatjuk az expozíciót. A képre kattintva és az `A` billentyűt megnyomva az expozíció önműködően beáll, a `H` jelenti az emberi szem reakciójának megfelelő beállítást, az `=` (egyenlőségjellel) pedig arra a képpontra állítjuk, amelyikre rákattintunk az egérrel. Az 5. és 6. kép házikóknk két nappali képét mutatja.

Az `ximage` arra is lehetőséget nyújt, hogy egy képterület átlagos fényadatait kiolvassuk. A kívánt tartományra húzzunk egy téglalapot, majd nyomjuk meg az `L` gombot a fényességszám (luminancia) vagy az `ENTER`-t a fényességértékeknek a leolvasásához. A számok fizikai jelentéséről a <http://www.intl-light.com/handbook/rad.html> címen találhatunk rövid ismertetést. A képből való kilépésre a `Q` gombot használhatjuk.

A hely korlátozottsága miatt ebben a cikkben a Radiance lehetőségeinek csak nagyon kis részét mutathattam be. Ha a jeleneteinket a mindennapi élet egyéb tárgyival tovább szeretnénk bővíteni, mindenképpen látogassunk el a Radiance weboldaláról elérhető honlapokra, ahol érdekes bútorokra, növényekre lelhetünk.

A cikkhez tartozó listák megtalálhatóak a 49. CD Magazin/Radiance könyvtárában.

Linux Journal 2003. június, 110. szám



Anthony W. Kay

Programozóként dolgozik az Oregon állambeli Eugene-ben. Ha éppen nem fák és építőanyagok szimulációján dolgozik, túrázni indul, hogy élőben is tanulmányozhassa őket.