Dr. Mileff Péter

GRAFIKA PROGRAMOZÁSA

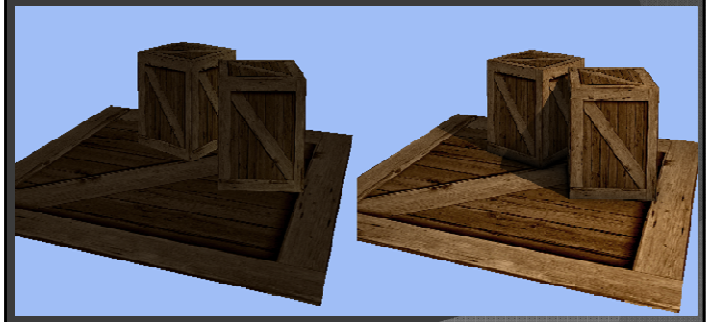
FÉNYEK A SZÁMÍTÓGÉPES GRAFIKÁBAN

Miskolci Egyetem
Általános Informatikai Tanszék

Fények a számítógépes grafikában

- A fények és árnyékok területe különösen frekventált terület a számítógépes vizualizációban.
  - Az utóbbi években ez tovább fokozódott
  - Oka a hardver folyamatos fejlődése
- **A miért egyszerű:**
  - A valóságban is a tárgyak a megvilágításoktól és árnyékoktól nyerik el igazi jellegüket.
  - **Rendkívüli vizuális élmény:** egy megfelelően elkészített megvilágítási rendszerrel segítségével
- Ebben ma a számítógépes játékok járnak élen úttörőként.
  - Pl: Cryengine, Unreal Engine, Source engine, stb.

Fények a számítógépes grafikában



Fények a számítógépes grafikában

- A terület sok éves fejlődésen ment keresztül.
- Új, további egyszerűsített megvilágítási módszerek is kialakultak
  - Oka: a mobil eszközök megjelenése
  - pl. Real-Time Area Lighting
- Maga a tématerület így nagyon nagy

5

ELMÉLETI ALAPOK…

6

A megvilágítási rendszerek alapjai

- **A megvilágítási modellek:**
  - A valóságot modellező, különböző megközelítésű leképzések
  - **Céljuk:** a valós világéval megegyező hatású képi inger előállítása
  - **Ki kell számítani:**
    - modellezni kell a fénysugarak felületekről történő visszaverődését.
    - Meg kell határozni a képernyő pixelben látható felületi pontokat és azok szem irányú sugársűrűségét (radiancia).
- A sugársűrűség meghatározása:
  - az **árnyalási egyenlet** (rendering equation) megoldásával
  - a számítógépes grafika ezen egyenlet megoldásával foglalkozik
    - Közelítő megoldások

7

A megvilágítási rendszerek alapjai

- A fények pontos fizikai szimulációja kimondottan számításigényes
- **Oka:**
  - a fényforrásból érkező fotonok útját kellene végigkövetni
  - a felületekről való visszaverődésekkel együtt.
- Ezt a megvilágítási forma a **globális illumináció**
- **Jellemzője:** a felületeken az indirekt megvilágítás hatása is érvényesül.
  - közvetett, más felületekről visszaverődött

8

A megvilágítási rendszerek alapjai

- A valós idejű vizualizációban jelenleg nem tudjuk ezt szimulálni
  - a jelenlegi hardver eszközök teljesítménye nem teszi még lehetővé.
- Egy mai játék sok objektummal és számos fényforrással dolgozik.
- A megjelenítésben jelenleg alkalmazott technikák így csak a **lokális illuminációt** alkalmazzák.

9

A megvilágítási rendszerek alapjai

- **Lokális illumináció:**
  - a pont színe a fényforrásokon kívül csak a lokális anyagjellemzőktől és geometriától függ,
  - nem vesszük figyelembe a szomszédos tárgyakról érkező szórt fényt.
  - Ezáltal csökken a realisztikusság, de az algoritmus hatékonysága és ezáltal a sebessége nő.

10

Lokális vs Globális illumináció

11

A megvilágítási rendszerek alapjai

- **A fejlődés iránya:**
  - A globális illumináció közelítése. Hamis globális illumináció.
    - Fake Global Illumination
  - Kiegészítik a lokális illuminációs modellt új algoritmusokkal
  - Ezek képesek:
    - a globális illuminációhoz hasonló eredményt elérni,
    - a lokális illuminációtól lényegesen jobban közelítő megvilágítás
  - Ilyen technikák
    - pl. az Ambient Occlusion, Lightmapping, Radiosity.
- A lokális illumináció mellett a vezető grafikus motorok kezdik bevezetni
  - Cél a látvány minőségi növelése

12

FÉNYFORRÁS TÍPUSOK, MEGVILÁGÍTÁSI MODELLEK…

13

Fényforrás típusok

- **Absztrakt fényforrás csoportok:** a valóság egy speciális foton sugárzási esetét modellező típusok
- **Pontszerű fényforrás (point light):**
  - a háromdimenziós világ egy pontjában található
  - kiterjedése nincs
  - A háromdimenziós tér egy tetszőleges p pontjában a sugárzási iránya p pontot és a fényforrás helyét összekötő vektor
  - Az intenzitás a távolság négyzetének arányában csökken
  - Pl. Az elektromos izzó - jó közelítéssel

14

Fényforrás típusok

- **Irányfényforrás (directional light):**
  - végtelen távollevő sík sugárzónak felel meg.
  - Az irány-fényforrás iránya és intenzitás a tér minden pontjában azonos.
  - Pl. a Nap a Földről nézve jó közelítéssel irány-fényforrásnak tekinthető.
- **Ambiens fényforrás (Ambient light):**
  - minden pontban és minden irányban azonos intenzitású.
- **Spotlámpa (spotlight):**
  - a virtuális világ egy pontjában található
  - iránnyal és hatóterülettel rendelkezik.
  - Pl. a zseblámpa spotlámpának tekinthető.

15

Megvilágítási modellek

- **Feladatuk:** a tárgyak árnyalásának kiszámítása
  - A felületekre eső fények intenzitásának és jellemzőinek meghatározása
- **Szórt háttérvilágítás (ambient light):**
  - az objektumok egyenletesen, minden irányból kapnak fényt
  - Hatása a nappali fényviszonyoknak felel meg erősen felhős égbolt esetén.
  - A számítógépes grafikában miért szükséges?
    - a felhasználó az ábrázolt jelenet összes objektumának a megvilágítását szabályozhassa.
  - A modellben nincs fényforrás: az objektumok „saját” fényüket bocsájtják ki
  - Ez megfelel annak, hogy a jelenetet egy irányfüggetlen, szórt fény világítja meg.

16

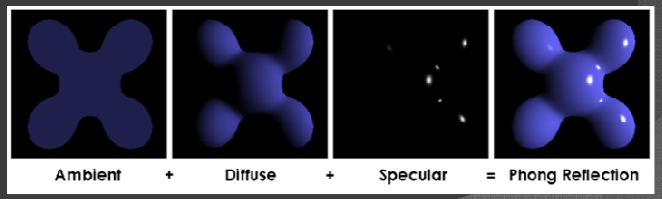
Megvilágítási modellek

- **Diffúz fényvisszaverődés (diffuse light):**
  - a matt felületek jellemzője
  - a megvilágított felület minden irányban ugyanannyi fényt ver vissza
- **Fényvisszaverődés fényes és csillogó felületekről (specular light):**
  - a sima felületekre általában az a jellemző, hogy rajtuk fényes foltokat is látunk
    - melyek helye nézőpontunkkal együtt változik
  - Ezek a felületek bizonyos irányokban visszatükrözik a fényforrásokat.
  - **Átmeneti eset:** a matt felületekre jellemző diffúz és a tökéletesen (ideálisan) tükröző felületekre jellemző visszaverődés között

17

Megvilágítási modellek

- **Shininess komponens:**
  - speciális anyagtulajdonság
  - a megvilágított anyagokon megjelenő spekuláris fényfoltok méretét és fényességét befolyásolja.



Phong árnyalás ambiens, diffúz és specular összetevőkből

18

Megvilágítási modellek

- Ezek alapján az árnyalási egyenletet általános alakban a következőképpen írhatjuk fel:

$$I = I_a + I_d + I_s$$

- I_a az ambiens,
- I_d a diffúz,
- I_s pedig a specular intenzitások.

- Ezek összege megadja a végleges színintenzitást.

19

Árnyalási modellek

- **Árnyalási algoritmusok**, amelyek az árnyalási egyenletet közelítik valamilyen szempontból.
- A sebesség és a minőség mérlegelendő alternatívák
  - így többféle algoritmus alakult ki
  - különböző optimalizálási megoldásokkal
- Az ideális megközelítés a pixelenkénti árnyalás
  - ahol minden pixelre külön ki kell számolni:
    - **a fény intenzitását**
    - **a normálvektorokat**
    - **egyéb jellemzőket**

20

ISMERTEBB ÁRNYALÁSI MÓDOK…

21

Árnyalási modellek

- **Árnyalási optimalizálásának alapelvei:**
  - **érdemes az árnyalási feladatot pixeleknél nagyobb egységekben megoldani**
  - kihasználni, hogy ha a szomszédos pixelekben ugyanazon felület látszik.
  - Ezen pixelekben látható felületi pontok optikai paraméterei igen hasonlók
    - normálvektora, megvilágítása, akár a látható színe is.
  - **Optimalizáció:**
    - vagy változtatás nélkül használjuk a szomszédos pixelekben végzett számítások eredményeit,
    - vagy pedig az inkrementális elv alkalmazásával egyszerű formulákkal tesszük azokat aktuálissá az új pixelben.

22

Árnyalási modellek

- **Konstans árnyalás:**
  - Síklapok árnyalására a leggyorsabb módszer.
  - a sokszögekre csak egyszer számítja ki az absztrakt fényforrások hatását.
  - **Színintenzitás:** a legegyszerűbb esetben a beeső fénysugár és a felületi normális szögének függvényében határozzuk meg
  - Ha valamelyik pixelben a sokszög látszik, akkor mindig ezzel a konstans színnel jelenítjük meg.
  - **Erőssége:** rendkívül gyors
  - **Hátránya:** Az eredmény általában nem kielégítő,
  - A mai napig számos valósidejű alkalmazás használja gyorsasága és egyszerűsége miatt.

23

Árnyalási modellek

- **Gouraud árnyalás:**
  - a háromszögek csúcspontjaiban értékeli ki a fényforrásokból odajutó fény visszaverődését
    - a csúcspontot érintő háromszögek normálvektorának átlagát felhasználva
  - Ezen modell segítségével kiszámoljuk a csúcspontoknál a szín intenzitást:
    - Gradiens átmeneteket számítunk a vertex pontoknál.
    - Ezután a háromszög belső pontjainak színét a csúcspontok színéből lineárisan interpolálja
  - **Erőssége:** az interpoláció, emiatt **gyors** raszterizáció
  - **Hátránya:** a gyengébb képminőség
    - Erősen lokalizált fény esetén az alacsony poligonszámú modellek esetén a fényvisszaverődés nem „szabályos”,
    - látszanak a vertex-ek menti interpolált értékek.

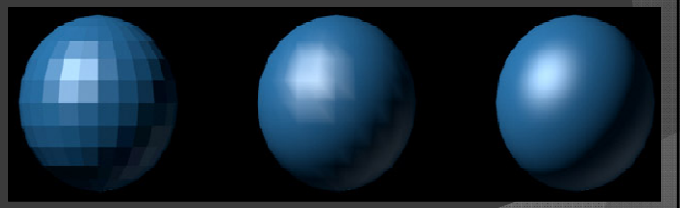
24

Árnyalási modellek

- **Phong árnyalás:**
  - A Goruaud árnyalástól abban különbözik:
    - magát a normálvektorokat határozza meg lineáris interpolációval a csúcspont normál vektorok segítségével.
  - A színek kiszámítása a színmodellnek megfelelően pixelenként történik.
  - Éppen ezért a legszebb, de leglassabb árnyalási mód.
  - A minősége nem függ a poligonok számától,
  - a „csillanás” akár a poligon közepén is megjelenik.

25

Konstans vs Gouraud vs Phong árnyalás



26

Konstans vs Gouraud vs Phong árnyalás

27

FELÜLETEK NORMÁLISA...

28

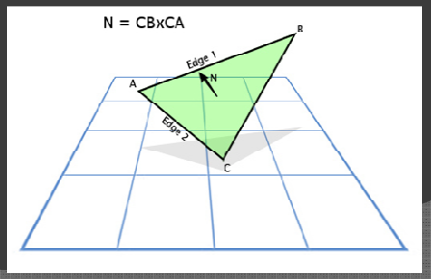
Felületek normálisa

- A megvilágítási modellek szerves részét képezi a felületek normálisa.
  - Célszerű tisztázni a felületi normális fogalmát
- **Felületi normális: egy felület normálisa egy a felületre merőleges, egységnyi hosszúságú vektor.**
- **Gyakorlatban:**
  - A normálisokat vertex-ekhez és felületekhez szokás társítani.
  - Ha egy modellen megvilágítást szeretnénk alkalmazni rendelkeznie kell kiszámított normálisokkal
  - Mind vertex, mind pedig háromszög (face) szinten

29

Felületek normálisa

- A normálisok kiszámítása egyszerű:
  - A háromszög vertex-eiből képezni kell két darab oldalt és azok vektorait.
  - A háromszög normálisa ilyenkor a két vektor vektoriális szorzata.



30

Normálist számító kód

```
CVector3 Cross(CVector3 vVector1, CVector3 vVector2){
   CVector3 vNormal;   // The vector to hold the cross product

    // The X value for the vector is:  (V1.y * V2.z) - (V1.z * V2.y)
   vNormal.x = ((vVector1.y * vVector2.z) - (vVector1.z * vVector2.y));

   // The Y value for the vector is:  (V1.z * V2.x) - (V1.x * V2.z)
   vNormal.y = ((vVector1.z * vVector2.x) - (vVector1.x * vVector2.z));

   // The Z value for the vector is:  (V1.x * V2.y) - (V1.y * V2.x)
   vNormal.z = ((vVector1.x * vVector2.y) - (vVector1.y * vVector2.x));

   return vNormal;
}
```

31

Felületek normálisa

Mikor számítsunk normálist?

- 1. A normálisok kiszámítását a modellek betöltésekor végezzük el
- 2. Már a modellező programban letárolva azt a vertex információk mellett a fájlban.

- Nagy modellek esetén célszerű fájlban tárolni
  - egy-egy komplexebb modelleket alkalmazó számítógépes játék esetében a betöltési idő nagyon megnőhet.

32

Mai trendek

- **A fejlődés iránya jól látszik:**
  - napjainkban a fix funkciós csővezeték leváltása folyik
  - a modern API-k már nem támogatják a fix funkciós megoldásokat.
    - A korábban alkalmazott *glLight()*, *glMaterial()*, stb függvények már csak „legacy” módban használhatók.
- Az OpenGL 3.0 verziótól mindent árnyalók segítségével kell megoldani
- **Előnye:**
  - az árnyalók bevezetése újabb lehetőség kínálkozik a grafikai minőség javításában,
  - a fények és árnyékok árnyalókkal való megvalósításával.
  - A programozók testre szabhatják a vizuális megoldásaikat.

33

Mai trendek

- **Hátránya:**
  - a programozóra több, főleg matematikai feladat megvalósítása hárul.
- Két fő csoport az árnyalókkal készített algoritmusoknak:
  - **vertex és pixel alapú árnyalások**
  - Attól függően, hogy egy egyszerűbb, vagy egy pontosabb számítási modellt szeretnénk alkalmazni.
- Az árnyalási egyenletet az árnyalókban implementált algoritmusokkal közelítjük.

34

EGYSZERŰ VERTEX SHADER ALAPÚ ÁRNYALÁS…

35

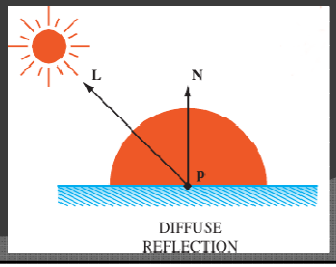
Vertex shader alapú árnyalás

- **Alapelve:**
  - Vertex árnyaló alkalmazása a számításokhoz
    - (Minden vertex-re egyszer hajtódik végre)
  - A fény intenzitásértékeket a vertex-ekben számítjuk ki
  - a kiszámított intenzitásértékeket interpoláljuk
  - a vertex-ek között elhelyezkedő pontokban
- **Lineáris interpoláció:**
  - gyors
  - minőségileg nem a legjobb

36

Vertex shader alapú árnyalás

- **Példa (egyszerű):**
  - egy irányított fényforrás alkalmazása
  - Csak diffúz árnyalás megvalósítása
  - Lambert-féle fény visszaverődési modell:
    - a visszaverődött fény intenzitása a megvilágítási szög koszinuszával arányos



37

Vertex shader alapú árnyalás

- A modellt leíró matematikai összefüggés (OpenGL Red Book):

$$I_o = L_d * M_d * \cos(\theta)$$

  - I_o jelenti a visszaverődött intenzitást,
  - L_d a fény diffúz színe,
  - M_d a material diffúz összetevője
  - cos() pedig a két vektor által bezárt szög.
- Jelen példában eltekintünk az anyagjellemzőktől
- A két vektor által bezárt szög meghatározása:
  - a szög éppen megegyezik a felületi normális (N) és a vizsgált felületi pontból a fényforrásba mutató vektor skaláris szorzatával.

38

Vertex shader alapú árnyalás

- **Vertex árnyaló:**

```
varying float Diffuse;

void main(void){
   // transform the normal into eye space and normalize it
   vec3 Normal = normalize(gl_NormalMatrix * gl_Normal);

   // OpenGL specification, the light is stored in eye space
   vec3 Light = normalize(gl_LightSource[0].position.xyz);

   Diffuse = max(dot(Normal, Light),0.0);
   gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex;
}
```

39

Vertex shader alapú árnyalás

- Az árnyalóban megtörténik a Lambert-féle visszaverődés kiszámítása minden vertex-re.
- **Árnyalás megvalósítása:**
  - Elsőként a modell normál vektorait a kamera (nézeti) térbe transzformáljuk.
  - Majd a fény irányvektorát normalizálni kell:
    - a skaláris szorzatban azonos mértékek kerüljenek alkalmazásra.
  - A skaláris szorzat eredménye maga a **diffúz összetevő**.

40

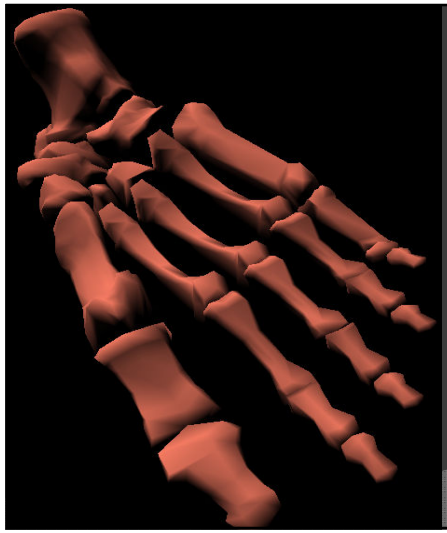
Vertex shader alapú árnyalás

- **Fragment árnyaló:**

```
varying float Diffuse;
void main(void){
   // Multiply the light Diffuse intensity by a white color
   gl_FragColor = Diffuse * vec4(1,1,1,1);
}
```

- A vertex árnyalóból a diffúz összetevő átadásra kerül a pixel árnyalónak.
- A pixel végső színét a *gl_FragColor*-ban kapjuk meg,
  - az egyszerűség kedvéért egy fehér színt alkalmazunk az árnyalás alap színeként.

41



Eredmény

42

GAME OVER