Real-Time High Quality Rendering

GAMES202, Lingqi Yan, UC Santa Barbara

Lecture 10:
Real-Time Physically-Based Materials (surface models)

Announcements

- Correction: 2/3 contents covered after today's lecture!
- GAMES202 homework late submission is open now!
- GAMES101 graders
  - Now we have a few applications for graders!
  - Will soon reach out to get started!
- GAMES101 now has 599K views on Bilibili!

Last Lecture

- Real-Time Global Illumination (screen space cont.)
  - Screen Space Directional Occlusion (SSDO)
  - Screen Space Reflection (SSR)
- Real-Time Physically-Based Materials

Today

- Real-Time Physically-Based Materials
  - Microfacet BRDF
  - Disney principled BRDF
- Shading with microfacet BRDFs under polygonal lighting
  - Linearly Transformed Cosines (LTC)

PBR and PBR Materials

基于物理的渲染

- Physically-Based Rendering (PBR)
  - Everything in rendering should be physically based

材质 照明 相机 光传输

  - Materials, lighting, camera, light transport, etc.
  - Not just materials, but usually referred to as materials :)

- PBR materials in RTR

RTR 材质明显落后于离线渲染. { 丰富程度 <种类> / 质量 准确度 <RTR在保证速度前提下的妥协质量> }

  - The RTR community is much behind the offline community
  - "PB" in RTR is usually not actually physically based :)

PBR Materials in RTR

- PBR materials in RTR
  - 对于表面
    For surfaces, mostly just microfacet models (used wrong so not PBR) and Disney principled BRDFs (artist friendly but still not PBR)

微表面模型

{ 基于 Disney 原则设计的 BRDFs

<都不是PBR>

PBR Materials in RTR

- PBR materials in RTR
  - **For surfaces**, mostly just microfacet models and Disney principled BRDFs
    对于体积
  - **For volumes**, mostly focused on fast and approximate single scattering and multiple scattering (for cloud, hair, skin, etc.)
    主要集中于快速和近似的单次散射和多次散射

[Lara Croft from the Tomb Raidar series]

PBR Materials in RTR

- PBR materials in RTR
  - **For surfaces**, mostly just microfacet models (used wrong so not PBR) and Disney principled BRDFs (artist friendly but still not PBR)
  - **For volumes**, mostly focused on fast and approximate single scattering and multiple scattering (for cloud, hair, skin, etc.)

通常没有太多新理论，但有很多实现技巧.

  - Usually not much new theory, but a lot of implementation hacks*
  - Still, performance (speed) is the key factor to consider

Recap: Microfacet BRDF

Microfacet BRDF

- What kind of microfacets reflect wi to wo?
  (hint: microfacets are mirrors)

F: 从一个角度看过去，有多少能量会反射.

G:

D: 微表面法线分布.

Fresnel term | shadowing-masking term | **distribution of normals**

$$f(\mathbf{i},\mathbf{o}) = \frac{\boxed{\mathbf{F}(\mathbf{i},\mathbf{h})}\boxed{\mathbf{G}(\mathbf{i},\mathbf{o},\mathbf{h})}\boxed{\mathbf{D}(\mathbf{h})}}{4(\mathbf{n},\mathbf{i})(\mathbf{n},\mathbf{o})}$$

The Fresnel Term

反射率取决于入射角<以及光的偏振>

Reflectance depends on incident angle (and polarization of light)

eg. 入射方向和法线n乎垂直,反射的能量是最多的 <掠射角度>

反射率随掠射角度的增加而增加.

This example: reflectance increases with grazing angle

[Lafortune et al. 1997]

Fresnel Term (Dielectric, $\eta = 1.5$)

绝缘体 eg. 水

反射率

与法线的夹角

Fresnel Term (Conductor)

导体. <金属>

Fresnel Term — Formulae

精确：要考虑偏振.

Accurate: need to consider polarization

$$R_s = \left|\frac{n_1\cos\theta_i - n_2\cos\theta_t}{n_1\cos\theta_i + n_2\cos\theta_t}\right|^2 = \left|\frac{n_1\cos\theta_i - n_2\sqrt{1-\left(\frac{n_1}{n_2}\sin\theta_i\right)^2}}{n_1\cos\theta_i + n_2\sqrt{1-\left(\frac{n_1}{n_2}\sin\theta_i\right)^2}}\right|^2,$$

$$R_p = \left|\frac{n_1\cos\theta_t - n_2\cos\theta_i}{n_1\cos\theta_t + n_2\cos\theta_i}\right|^2 = \left|\frac{n_1\sqrt{1-\left(\frac{n_1}{n_2}\sin\theta_i\right)^2} - n_2\cos\theta_i}{n_1\sqrt{1-\left(\frac{n_1}{n_2}\sin\theta_i\right)^2} + n_2\cos\theta_i}\right|^2.$$

$$R_{\text{eff}} = \frac{1}{2}\left(R_s + R_p\right).$$

近似：Schlick's 近似.

Approximate: Schlick's approximation

基础反射率.

$$R(\theta) = R_0 + (1-R_0)(1-\cos\theta)^5$$

值域 (R_0 , 1)

$$R_0 = \left(\frac{n_1 - n_2}{n_1 + n_2}\right)^2$$

Normal Distribution Function (NDF)

微表面法线分布

- Key: the distribution of microfacets' normals

法线集中 <=> 有光泽 glossy

  - Concentrated <==> glossy

法线分布扩散 <=> diffuse

  - Spread <==> diffuse

如何变化?

可以在高度做 scale 操作
上下拉伸

Normal Distribution Function (NDF)

法线分布函数，<二维函数>

- The Normal Distribution Function (NDF)

  与正态分布无关，只是简称一致>

  - Note: has nothing to do with the normal distribution in stats
  - Various models to describe it
    - Beckmann, GGX, etc.
    - Detailed models [Yan 2014, 2016, 2018, …]

Normal Distribution Function (NDF)

- Beckmann NDF <表示各向同性>
  - Similar to a Gaussian
  - But defined on the **slope space**

定义在坡度空间

$$D(h) = \frac{e^{-\frac{\tan^2\theta_h}{\alpha^2}}}{\pi\alpha^2\cos^4\theta_h}$$

$\tan\theta_h: (0, +\infty)$

$\theta_h: (0, 90°)$

↳ 定义在 $\tan\theta_h$ 那条线上可以保证法线角度不会超下.

表面粗糙度.

α: roughness of the surface (the smaller, the more like mirror/specular)

θ_h: angle between half vector h and normal n

半向量h与法线n的夹角

Normal Distribution Function (NDF)

- GGX (or Trowbridge-Reitz) [Walter et al. 2007]
  - Typical characteristic: long tail! 可以有光晕效果

Normal Distribution Function (NDF)

- Comparison: Beckmann vs. GGX

https://planetside.co.uk/news/terragen-4-5-release/

Normal Distribution Function (NDF)

GGX 拓展

- Extending GGX [by Brent Burley from WDAS]
  - GTR (Generalized Trowbridge-Reitz)
  - Even longer tails 可以有更长的尾巴

Shadowing-Masking Term

解决微表面产生的自遮挡问题

- Or, the geometry term G

$$f(i,o) = \frac{F(i,h)G(i,o,h)D(h)}{4(n,i)(n,o)}$$

  - Account for self-occlusion of microfacets
  - Shadowing — light, masking — eye
  - Provide darkening esp. around grazing angles

提供变暗的操作

垂直看 几乎不变暗 1

掠射角 变暗很多 ~0

掠射角度

光线发射

眼睛接收

Shadowing-Masking Term

- Why is it important?
  - Suppose no G term, what will happen when the incident / outgoing is from grazing angle?

Substance Painter Viewport

$$f(i,o) = \frac{F(i,h)G(i,o,h)D(h)}{4(n,i)(n,o)}$$

Can be arbitrarily bright around grazing angles!

如果不考虑G项
则在 grazing angle $\begin{cases} F(i,h) \text{ 接近 } 1 \\ (n,i) \text{ 接近 } 0 \end{cases}$
$f(i,o)$ 会非常大. 导致边缘非常亮.

Shadowing-Masking Term

- A commonly used shadowing-masking term
  - The Smith shadowing-masking term
  - Decoupling shadowing and masking 把 shadowing. Masking 解耦合.

$$G(\mathbf{i},\mathbf{o},\mathbf{m}) \approx G_1(\mathbf{i},\mathbf{m})G_1(\mathbf{o},\mathbf{m})$$

法线分布.

<BRDF会非常小>
在 grazing angle
就不会过亮

Multiple Bounces

- Missing energy! 能量损失.
  - Especially prominent when roughness is high (why?)

https://fpsunflower.github.io/ckulla/data/s2017_pbs_imageworks_slides_v2.pdf

Multiple Bounces

- Missing energy!

  roughness越高
  自遮挡情况越严重 → 损失能量越多 <如果只考虑1 bounce>
  光线在微表面多次弹射占比越大

- Adding back the missing energy?
  - Accurate methods exist [Heitz et al. 2016]
  - But can be too slow for RTR

- Basic idea
  - Being occluded == next bounce happening

    被遮挡 == 下次弹射发生

The Kulla-Conty Approximation

基于经验式的方式补全多次反射丢失的能量.

<传出的 2D BRDF 波瓣的总能量>

- What's the overall energy of an outgoing 2D BRDF lobe?

<1次弹射.
可以出射的能量>

E 和观察方向有关.

$$E(\mu_o) = \int_0^{2\pi} \int_0^1 \underbrace{f(\mu_o, \mu_i, \phi)}_{\text{BRDF}} \overbrace{\mu_i d\mu_i}^{\cos\theta} \overbrace{d\phi}^{\sin\theta\, d\theta\, d\phi}$$

球面立体角积分

从任何方向 $L_i = 1$.

$d\mu_i = \cos\theta\, d\theta$.

Note: $\mu = \sin\theta$

- Key idea

<1次 bounce 中
被遮挡的能量>

可设计1个附加波瓣.将其能量为 $1 - E(\mu_o)$

  - We can design an additional lobe that integrates to $1 - E(\mu_o)$

对于不同的入射方向、传出的 BRDF 波瓣也是不同的.

  - The outgoing BRDF lobe can be different for different incident dir.

考虑到 BRDF 可逆性.(相互)

  - Consider reciprocity, it should be* of the form

$$c(1 - E(\mu_i))(1 - E(\mu_o))$$

new BRDF 的形式

The Kulla-Conty Approximation

- Therefore, 设计这样一种BRDF 使得它的积分正是丢失的那一部分能量.

$$f_{\mathrm{ms}}(\mu_o, \mu_i) = \frac{(1 - E(\mu_o))(1 - E(\mu_i))}{\pi(1 - E_{\mathrm{avg}})}, E_{\mathrm{avg}} = 2\int_0^1 E(\mu)\,\mu\,d\mu$$

- FYI, validation:

$$\begin{aligned}
E_{\mathrm{ms}}(\mu_o) &= \int_0^{2\pi}\int_0^1 f_{\mathrm{ms}}(\mu_o, \mu_i, \phi)\mu_i d\mu_i d\phi \\
&= 2\pi\int_0^1 \frac{(1 - E(\mu_o))(1 - E(\mu_i))}{\pi(1 - E_{\mathrm{avg}})}\mu_i d\mu_i \\
&= 2\frac{1 - E(\mu_o)}{1 - E_{\mathrm{avg}}}\int_0^1 (1 - E(\mu_i))\,\mu_i\,d\mu_i \\
&= \frac{1 - E(\mu_o)}{1 - E_{\mathrm{avg}}}(1 - E_{\mathrm{avg}}) \\
&= 1 - E(\mu_o)
\end{aligned}$$

https://fpsunflower.github.io/ckulla/data/s2017_pbs_imageworks_slides_v2.pdf

The Kulla-Conty Approximation

- But neither $E(\mu)$ nor $E_{avg} = 2\int_0^1 E(\mu)\mu \, \mathrm{d}\mu$ are analytic

- But we already know what to do!
  - Hint: in split sum, how do we deal with a difficult integral?
  - Precompute / tabulate!

    预计算 制表 ⇒ 你猜 { 观测方向 μ_o roughness <BRDF>

- Dimension / parameters of $E(\mu)$ and E_{avg}?
  - $E(\mu)$: roughness & μ [therefore, a 2D table]
  - E_{avg}: just roughness [therefore, a 1D table]

Precomputed table for $E(\mu)$

The Kulla-Conty Approximation

- Results

https://fpsunflower.github.io/ckulla/data/s2017_pbs_imageworks_slides_v2.pdf

The Kulla-Conty Approximation

- What if the BRDF has color? <吸收. 能量损失>
  - Color == absorption == energy loss (as it should)
  - So we'll just need to compute the overall energy loss

    只需要计算总的能量损失.
- Define the average Frensel (how much energy is reflected)

  假设反射很多次. 平均每次反射多少能量

$$F_{avg} = \frac{\int_0^1 F(\mu)\mu \, \mathrm{d}\mu}{\int_0^1 \mu \, \mathrm{d}\mu} = 2\int_0^1 F(\mu)\mu \, \mathrm{d}\mu$$

[Therefore, just a number]

  E_{avg} 是可以看到的能量 <不再参与进一步的 bounce>
- And recall that E_{avg} is how much energy that you can see (i.e., will **NOT** participate in further bounces)

  $(1-E_{avg})$ 参与后续 bounce 的能量.

The Kulla-Conty Approximation

- Therefore, the proportion of energy (color) that

  (初始能量为1)　多少能量被反射

  - You can directly see: $F_{avg}E_{avg}$　多少能量被反射可以直接看到.
  - After one bounce then be seen: $F_{avg}(1-E_{avg})\cdot F_{avg}E_{avg}$
  - …
  - After k bounces then be seen: $F_{avg}^{k}(1-E_{avg})^{k}\cdot F_{avg}E_{avg}$

  上述都加合　得到颜色项.

- Adding everything up, we have the color term
  - Which will be directly multiplied on the uncolored **additional BRDF**

    将直接与未着色的附加BRDF相乘.

$$\frac{F_{\text{avg}}E_{\text{avg}}}{1-F_{\text{avg}}\left(1-E_{\text{avg}}\right)}$$

The Kulla-Conty Approximation

- Results

https://fpsunflower.github.io/ckulla/data/s2017_pbs_imageworks_slides_v2.pdf

However, An Undesirable Hack

- Combining a Microfacet BRDF with a **diffuse** lobe
  - Pervasively used in computer vision for material recognition
  - COMPLETELY WRONG
  - COULDN'T BE WORSE
  - I NEVER TAUGHT YOU SO

- Issues
  - Physically incorrect
  - Not energy preserving 不保证能量守恒.
    (fixed in Kulla-Conty)
    (can also be fixed in other ways)

Questions?

Disney’s Principled BRDF

Why is it needed?

- Motivation
  - No physically-based materials are good at rep. all real materials
    - e.g. lacking diffuse term in most microfacet models
  - Physically-based materials are not artist friendly
    - e.g. “the complex index of refraction n-ik”
- High level design goal
  - Art directable, not necessarily physically correct
  - But again, referred to as PBR in real-time rendering…

What is “principled”?

- The BRDF is designed with a few important principles
  - Intuitive rather than physical parameters should be used.
  - There should be as few parameters as possible.
  - Parameters should be zero to one over their plausible range.
  - Parameters should be allowed to be pushed beyond their plausible range where it makes sense.
  - All combinations of parameters should be as robust and plausible as possible.

How does it work?

- A table showing the effects of **individual** parameters

Pros and Cons

- Easy-to-understand / control
- A wide range of materials in a single model
- Open source implementation is available
- Not physically based
  - But is it a big problem?
  - Academia vs. industry
- Huge parameter space

Questions?

Shading Microfacet Models using Linearly Transformed Cosines (LTC)

Linearly Transformed Cosines

- Solves the shading of microfacet models
  - Mainly on GGX, though others are also fine
  - No shadows
  - Under polygon shaped light

Linearly Transformed Cosines

- Key idea
  - Given the viewing direction, any outgoing 2D BRDF lobe can be transformed to a cosine
  - The shape of the light can also be transformed along
  - Integrating the transformed light on a cosine lobe can be **analytical**

Next Lecture

- More Real-Time Physically-Based Materials!

https://www.wired.com/story/cloud-gaming-infrastructure-arms-race/

Thank you!