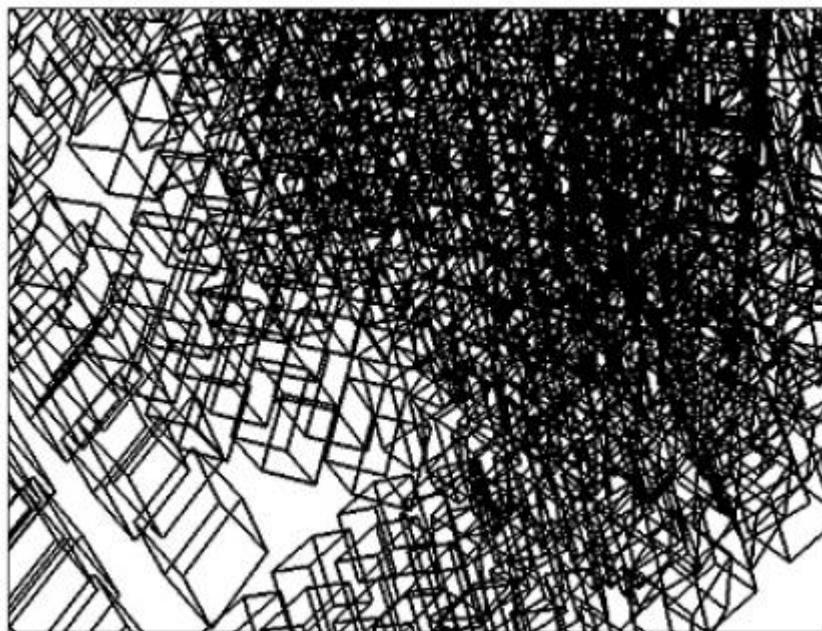


## بسمه تعالی سایه زنی

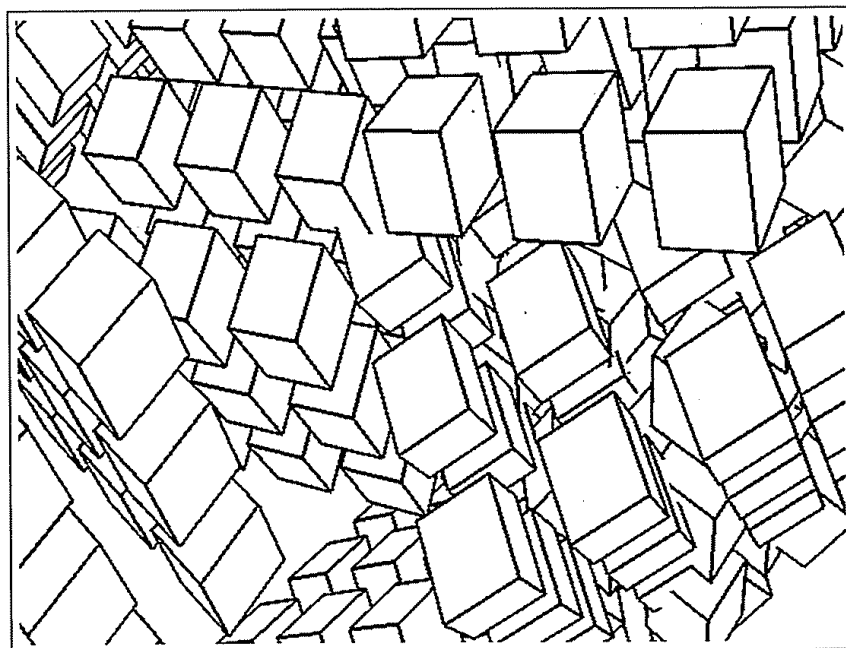
### مقدمه

نمایش واقعی اشیاء بوسیله تولید تصاویر پرسپکتیو آنها بصورت قاب سیمی در اولین مرحله قابل دستیابی است.



شکل ۱ نمایش قاب سیمی یک صحنه (Hill, et al)

با حذف سطوح مخفی به نمایش واقعی اشیاء نزدیکتر می شویم.

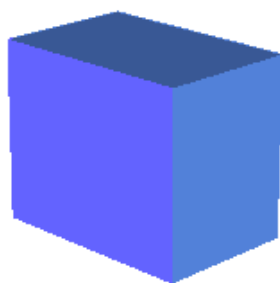


شکل ۲ نمایش قاب سیمی با حذف سطوح مخفی.

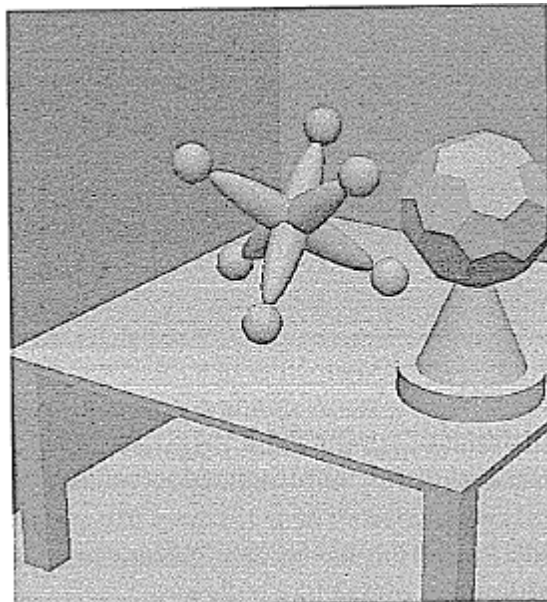
. مرحله بعد اعمال سایه یا رنگ به سطوح قابل دید می باشد. در ساده ترین حالت می توان تصور نمود که هر وجهی یک شدت یا رنگ ذاتی دارد و ارتباطی با منابع نوری اطراف خود ندارد.



اما در واقع قسمتهای مختلف شیء مقادیر متفاوتی از نور تابش شده را باز می تابانند.



در سایه زنی صاف<sup>۱</sup> محاسبه مقدار نوری که از یک وجه بازتابانده می شود در یک نقطه محاسبه می شود، ولی در سایه زنی نرم<sup>۲</sup> محاسبه نور بازتابانده شده در هر نقطه از وجه بصورت مجزا محاسبه می گردد.

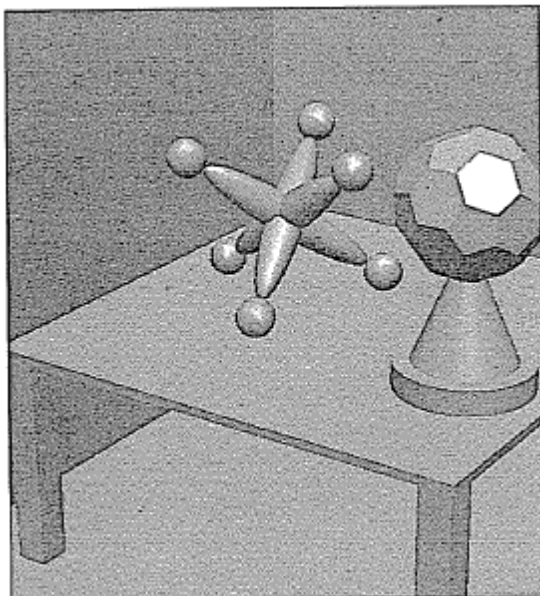


شکل ۳

<sup>۱</sup> Flat shading

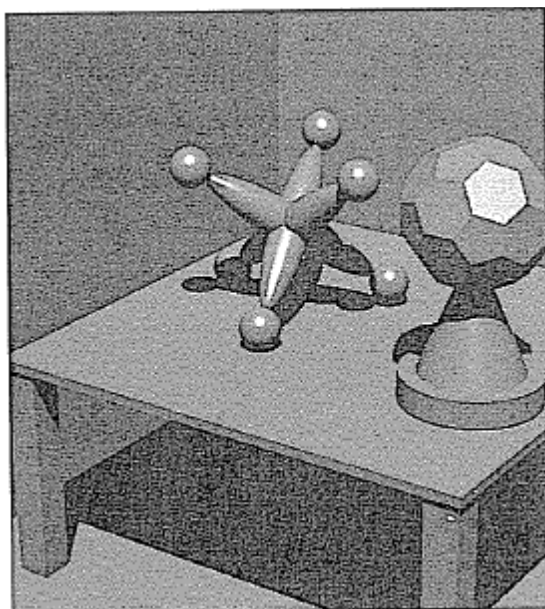
<sup>۲</sup> Smooth shading

مرحله بعد افزودن درخشندگی می باشد.



شکل ۴

برای واقعی تر شدن می توان سایه های ایجاد شده را نیز ایجاد نمود.



شکل ۵

افزودن بافت مرحله بعدی برای نمایش واقعی تر جسم می باشد.



شکل ۶ نگاشت بافت بر روی اشیاء

یک مدل سایه زنی برای محاسبه شدت نوری که هنگام دیدن سطح باید مشاهده شود استفاده می شود. در ابتدا نور بدون رنگ را در نظر می گیریم. محاسبات شدت بستگی به خواص اپتیکی سطوح، مکان نسبی سطوح، و تمایل آنها نسبت به منبع نور دارد.

### مدلسازی شدت‌های نور

نوری که از سطوح منعکس می شود از منابع نوری مختلفی حول شی می آید. اگر شی شفاف باشد، نوری که از منابع پشت شی نیز وجود دارند دیده می شود. منابع نوری که یک شی را نورافشانی می کنند عبارتند از سطوح نور پخش‌شان، و سطوح منعکس کننده نور. انعکاسهای مختلف که از اشیاء منعکس کننده نور وجود دارند یک نورافشانی یکنواخت بنام نور وافر، یا نور پس زمینه را ایجاد می کنند.

نحوه تعامل نور برخورد شده به یک سطح

- مقداری جذب و تبدیل به گرما می شود
- مقداری منعکس می شود
- مقداری به داخل شی منتقل می شود (همانند شیشه)

اگر همه نور برخورد شده جذب شود شی سیاه به نظر می رسد. (بدنه سیاه). فرض می کنیم که دو نوع انعکاس نور برخورد شده به یک جسم وجود دارد:

- انعکاس پراکنده<sup>۴</sup> - نور برخورد شده بطور یکنواخت در همه جهتها بازتابانده می شود.
- انعکاسهای درخشانده<sup>۵</sup> - همانند آینه و بسیار جهتدار

بسیاری از سطوح ترکیبی از این دو می باشند.

<sup>3</sup> black body

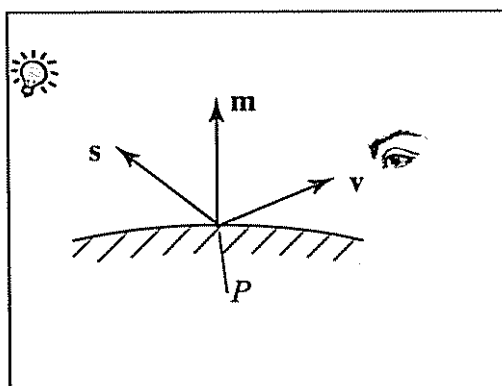
<sup>4</sup> Diffuse

<sup>5</sup> Specular

### اجزاء هندسی محاسبه نور منعکس شده

برای محاسبه نوری که از یک نقطه P به چشم می رسد به شناخت سه بردار اصلی نیازمندیم:

- بردار عمود m به سطح در نقطه P،
- بردار v از P به چشم بیننده، و
- بردار s از P به منبع نور



شکل ۷

### نور وافر

در ساده ترین مدل نورافشانی، هر شیء با استفاده از یک شدت ذاتی نمایش داده می شود. یک منبع بدون جهت و پراکنده که نتیجه انعکاسهای نور از سطوح بسیاری است که در محیط هستند نور وافر نامیده می شود. در این صورت:

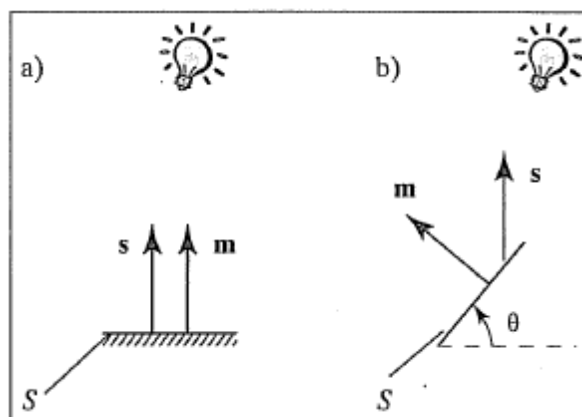
$$I = I_a \rho_a$$

که  $\rho_a$  ضریب انعکاس نور وافر می باشد.

### نور پراکنده

هنگامی که نور از یک منبع به یک سوی یک وجه برخورد می کند بخشی از آن بصورت یکنواخت در همه جهات پراکنده می شود. بخشی از این بازتابش به چشم بیننده خواهد رسید با شدتی که با  $I_d$  نمایش داده می شود. نکته مهم در مورد نور پراکنده آن است که این شدت مستقل از جهت نقطه P به چشم بیننده است. این خاصیت بنام پراکندگی تمام جهته شناخته می شود، چرا که نور چنان بصورت یکنواخت در همه جهات پراکنده می شود که تمایل وجه نسبت به بیننده بدون تأثیر است. بنابر این  $I_d$  مستقل از v و m می باشد (بجز آنکه  $v \cdot m < 0$  که در این صورت  $I_d = 0$ ). از سوی دیگر، میزان نوری که وجه را نورافشانی می کند به تمایل وجه با منبع نور وابسته است.

<sup>6</sup> Ambient



شکل ۸

طبق قانون لمبرت<sup>۷</sup> داریم:

$$I_d = I_s \rho_d \cos \theta$$

$$= I_s \rho_d \frac{s \cdot m}{|s| |m|}$$

که  $I_s$  شدت منبع نور و  $\rho_d$  ضریب انعکاس پراکندگی می باشد. همانگونه که مشاهده می شود رابطه فوق مستقل از زاویه بین  $m$  و  $v$  می باشد. در حالتیکه عمود به سطح به دور از بیننده است، سطح دیده نمی شود در این صورت:

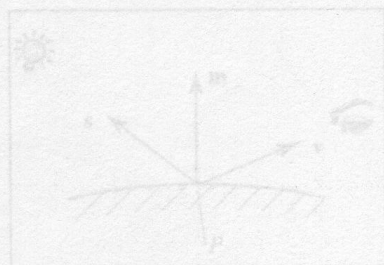
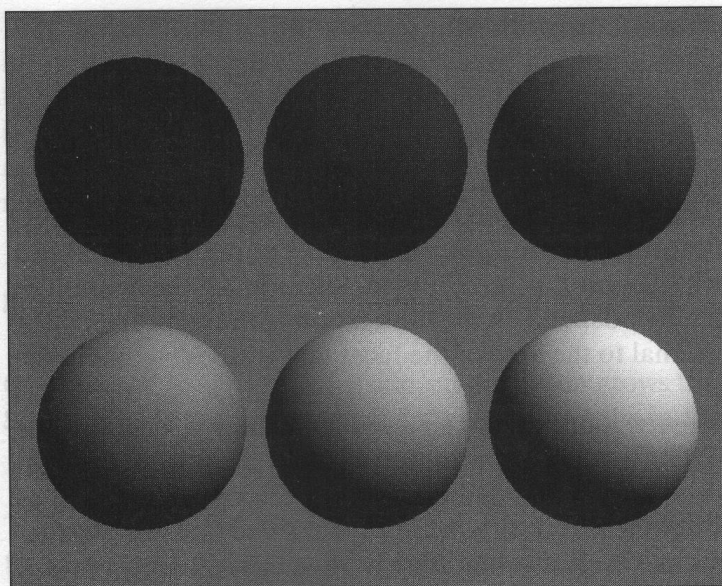
$$I_d = I_s \rho_d \max(\cos \theta, 0)$$

$$= I_s \rho_d \max\left(\frac{s \cdot m}{|s| |m|}, 0\right)$$

شکل ۹ یک چگونگی ظاهر شدن یک کره را برای ضریب انعکاس پراکندگی  $0, 0,2, 0,4, 0,6, 0,8, 1,0$  و نمایش می دهد. شدت نور منبع  $1,0$  می باشد.

<sup>7</sup> Lambert

**FIGURE 8.11** Spheres with various reflection coefficients shaded with diffuse light.



**FIGURE 8.8** Important directions used in computing reflected light.

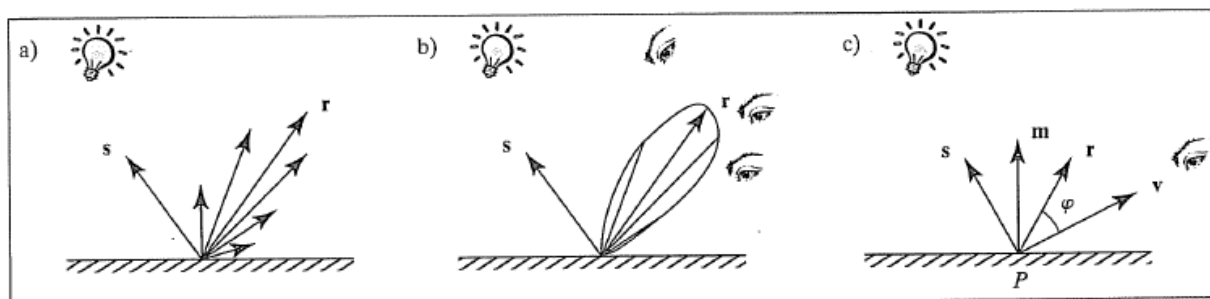
شکل ۹

### انعکاس درخشنده

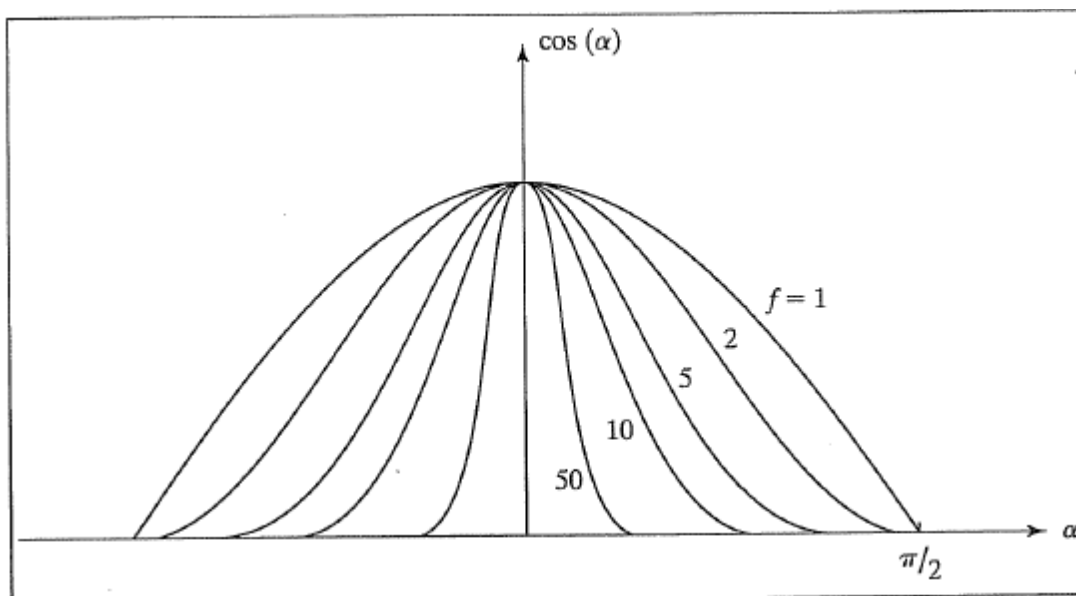
در مورد آینه کامل: نور منعکس شده بصورت یک اشعه با زاویه ای برابر نسبت به عمود از سمت دیگر شیء خارج می شود.



در مورد اشیاء درخشان دیگر: نور در جهات مختلف منعکس می شود و بیشترین مقدار آن در راستای انعکاس آینه کامل است. هر چه از این راستا فاصله گرفته شود میزان بازتابش کاهش می یابد.



شکل ۱۰



شکل ۱۱ افت نور درخشنده با زاویه

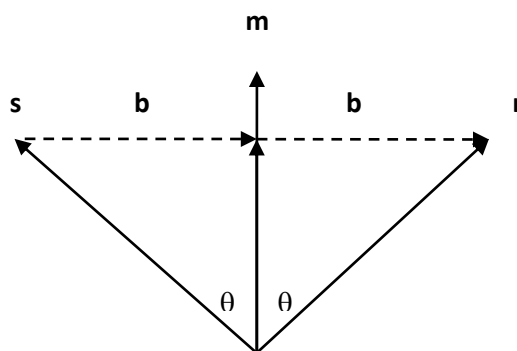
در مدل فانگ<sup>۸</sup>، میزان نور منعکس شده با افزایش زاویه  $\varphi$  کاسته می شود، و در رابطه است با

بنابر این:

$$I_{sp} = I_s \rho_s \cos^f(\varphi)$$

$$= I_s \rho_s \left( \frac{r \cdot v}{|r| |v|} \right)^f$$

$\rho_s$  ضریب انعکاس درخندگی می باشد. هر چه مقدار  $f$  بیشتر باشد به آینه کامل نزدیکتر خواهیم بود.

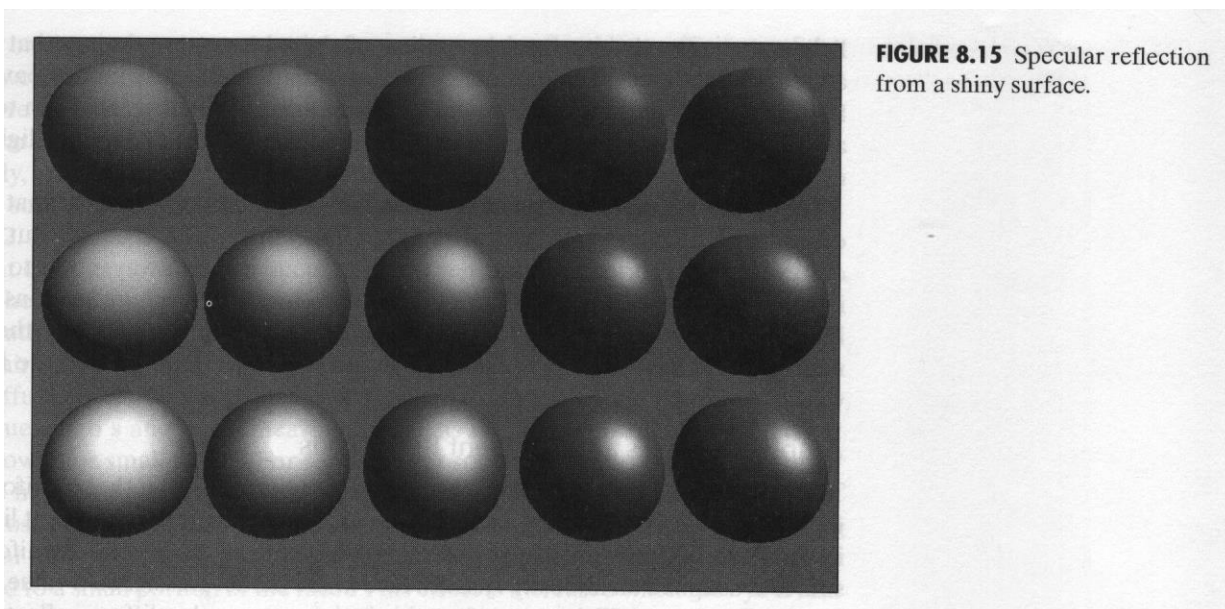


می توان نشان داد که بردار  $r$  بصورت زیر بدست می آید:

$$r = -s + 2 \frac{(s \cdot m)}{|m|^2} m$$

<sup>8</sup> Phong





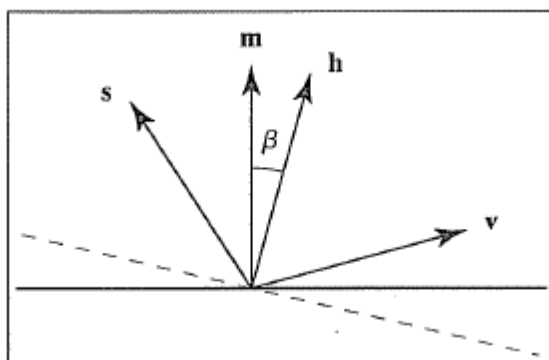
**FIGURE 8.15** Specular reflection from a shiny surface.

شکل ۱۲

در شکل ۱۲ ضریب انعکاس درخشندگی از بالا به پائین افزایش می یابد.  $f$  از چپ به راست اضافه می شود.

### افزایش کارآئی

محاسبه  $r$  زمانبر می باشد. می توانیم از بردار نیمساز بین  $s$  و  $h$  استفاده کنیم.  $(h = s + v)$ . هنگامی که  $\beta$  برابر صفر باشد  $v$  در جهت  $r$  قرار گرفته و بیشترین انعکاس را داریم. هر چه  $\beta$  افزایش یابد میزان بازتابش کاهش می یابد.



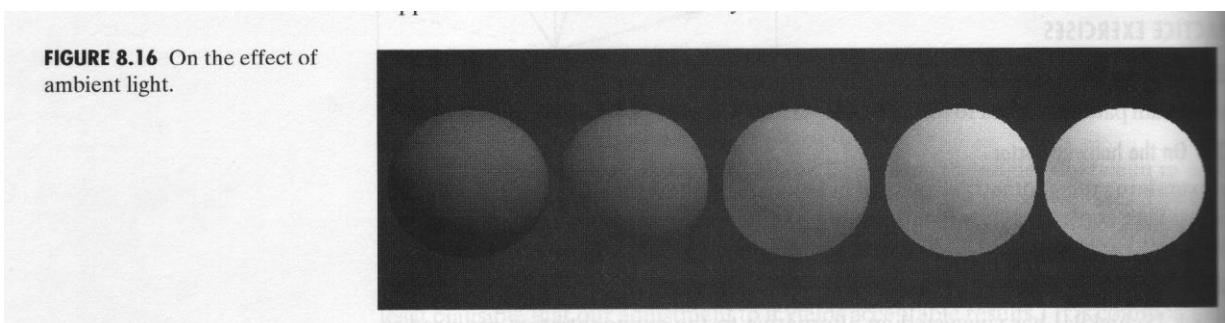
شکل ۱۳

$$I_{sp} = I_s \rho_s \max(\cos^f(\beta), 0)$$

$$= I_s \rho_s \max\left(\left(\frac{h \cdot m}{|h| |m|}\right)^f, 0\right)$$

### افزودن نور وافر

در برخی موارد که نقطه روی سطح بدور از منبع نور است کاملاً در تاریکی قرار می گیرد. به همین علت معمولاً مقداری نور وافر در نظر می گیرند تا اطراف شی بهتر مشخص باشد.



شکل ۱۴

### ترکیب نورها

در حالت کلی می توان نوری که از نقطه P به بیننده می رسد را ترکیبی از همه انواع نورها دانست:

$$I = I_a \rho_a + I_s \rho_d \max(\cos(\theta), 0) + I_s \rho_s \max(\cos^f(\beta), 0)$$

### اضافه کردن رنگ

بسط مدل سایه زنی به نور رنگی ساده است. مدل سایه زنی به هر یک از رنگهای اصلی جداگانه اعمال می شود.

$$I_r = I_{ar} \rho_{ar} + I_{sr} \rho_{dr} \max(\cos(\theta), 0) + I_{sr} \rho_{sr} \max(\cos^f(\beta), 0)$$

$$I_g = I_{ag} \rho_{ag} + I_{sg} \rho_{dg} \max(\cos(\theta), 0) + I_{sg} \rho_{sg} \max(\cos^f(\beta), 0)$$

$$I_b = I_{ab} \rho_{ab} + I_{sb} \rho_{db} \max(\cos(\theta), 0) + I_{sb} \rho_{sb} \max(\cos^f(\beta), 0)$$

ضریب انعکاس وافر برای هر رنگ:  $\rho_{ab}$ ,  $\rho_{ag}$ ,  $\rho_{ar}$  و

ضریب انعکاس پراکنده:  $\rho_{db}$ ,  $\rho_{dg}$ ,  $\rho_{dr}$  و

ضریب انعکاس درخشانده:  $\rho_{sb}$ ,  $\rho_{sg}$ ,  $\rho_{sr}$  و

### اشیاء ساخته شده از مواد مختلف

Material	Ambient: $\rho_{ar}, \rho_{ag}, \rho_{ab}$	Diffuse: $\rho_{dr}, \rho_{dg}, \rho_{db}$	Specular: $\rho_{sr}, \rho_{sg}, \rho_{sb}$	Exponent: $f$
Black Plastic	0.0 0.0 0.0	0.01 0.01 0.01	0.50 0.50 0.50	32
Brass	0.329412 0.223529 0.027451	0.780392 0.568627 0.113725	0.992157 0.941176 0.807843	27.8974
Bronze	0.2125 0.1275 0.054	0.714 0.4284 0.18144	0.393548 0.271906 0.166721	25.6
Chrome	0.25 0.25 0.25	0.4 0.4 0.4	0.774597 0.774597 0.774597	76.8
Copper	0.19125 0.0735 0.0225	0.7038 0.27048 0.0828	0.256777 0.137622 0.086014	12.8
Gold	0.24725 0.1995 0.0745	0.75164 0.60648 0.22648	0.628281 0.555802 0.366065	51.2
Pewter	0.10588 0.058824 0.113725	0.427451 0.470588 0.541176	0.3333 0.3333 0.521569	9.84615
Silver	0.19225 0.19225 0.19225	0.50754 0.50754 0.50754	0.508273 0.508273 0.508273	51.2
Polished silver	0.23125 0.23125 0.23125	0.2775 0.2775 0.2775	0.773911 0.773911 0.773911	89.6

### استفاده از منابع نور در OpenGL

اجازه تعریف تا ۸ منبع نور GL\_LIGHT0 الی GL\_LIGHT7

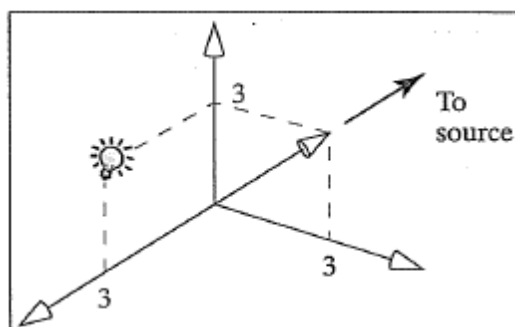
```
GLfloat myLightPosition[] = {3.0, 6.0, 5.0, 1.0};
glLightfv(GL_LIGHT0, GL_POSITION, myLightPosition);
glEnable(GL_LIGHTING); // enable lighting in general
glEnable(GL_LIGHT0); // enable source GL_LIGHT0
```

برخی منابع مثل چراغ مطالعه در صحنه هستند و برخی مثل خورشید بسیار دور هستند

$(x, y, z, 1)$ : a local light source at the position  $(x, y, z)$

$(x, y, z, 0)$ : a vector to an infinitely remote light source in the direction  $(x, y, z)$

مثال: شکل ۱۵ یک منبع محلی در  $(0,3,3,1)$  و یک منبع خارجی در طول بردار  $(3,3,0,0)$  را نشان می دهد.



شکل ۱۵

نورهای بسیار دور اغلب نورهای جهتی نامیده می شوند. برای نورهای جهتی، جهت  $S$  در محاسبات همواره ثابت است. می توان رنگهای متفاوتی به یک منبع نور برای هر یک از سه نوع نور وافر، پراکنده، و درخشنده آن نسبت داد. البته شاید عجیب باشد که به منبع نسبت می دهیم. ولی باعث می شود که هر موقع که بخواهیم آنرا روشن و خاموش کنیم.

```
GLfloat amb0[] = {0.2, 0.4, 0.6, 1.0}; // define some colors
GLfloat diff0[] = {0.8, 0.9, 0.5, 1.0};
GLfloat spec0[] = {1.0, 0.8, 1.0, 1.0};
glLightfv(GL_LIGHT0, GL_AMBIENT, amb0); // attach them to LIGHT0
glLightfv(GL_LIGHT0, GL_DIFFUSE, diff0);
glLightfv(GL_LIGHT0, GL_SPECULAR, spec0);
```

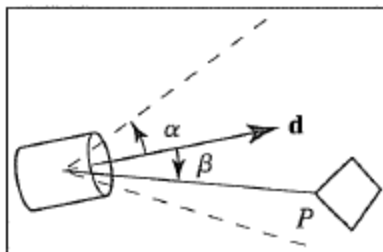
رنگها در فرمت RGBA(red,green,blue,alpha) داده می شوند. Alpha برای ترکیب رنگها استفاده می شود فعلاً آنرا ۱ در نظر می گیریم. منابع نور مقادیر پیش فرضی دارند.

- برای همه منابع نور پیش فرض نور وافر:  $(0,0,0,1)$  سیاه
- برای LIGHT0 پیش فرض نور پراکنده و نور درخشنده  $(1,1,1,1)$  سفید
- برای منابع دیگر مقادیر نور پراکنده و نور درخشنده سیاه است.

## نورافکن

بطور پیش فرض منابع نوری در OpenGL منابع نقطه ای هستند. نور در همه جهات بطور یکنواخت پراکنده می شود. می توان آنها را نورافکن کرد یعنی فقط در جهات خاصی نور پراکنده شود.

<sup>9</sup> spotlight



شکل ۱۶

نور در جهت  $d$  منتشر می شود. نور رسیده به  $P$  با ضریب  $\cos^{\epsilon}(\beta)$  تضعیف می شود. مقدار  $\epsilon$  توسط کاربر انتخاب می شود.

```
glLightf(GL_LIGHT0, GL_SPOT_CUTOFF, 45.0); // a cutoff angle of 45°
glLightf(GL_LIGHT0, GL_SPOT_EXPONENT, 4.0); // ε = 4.0
GLfloat dir[] = {2.0, 1.0, -4.0}; // the spotlight's direction
glLightfv(GL_LIGHT0, GL_SPOT_DIRECTION, dir); // send the
direction vector, dir
```

مقادیر پیش فرض:  $\epsilon=0$ ,  $\alpha=180^\circ$ ,  $d=(0,0,-1)$

### تضعیف نور با فاصله

OpenGL اجازه می دهد که سرعت تضعیف نور با فاصله گرفتن از منبع را مشخص نمائیم.

ضریب تضعیف:

$$atten = \frac{1}{k_c + k_l D + k_q D^2}$$

$D$  فاصل بین منبع و رأس مورد نظر می باشد. میزان تضعیف را می توان بصورت ثابت، خطی، و یا درجه دو مدل نمود. توسط دستور

زیر مقدار ضریب تضعیف ثابت برابر  $2.0$  در نظر گرفته می شود.

```
glLightf(GL_LIGHT0, GL_CONSTANT_ATTENUATION, 2.0);
```

پارامترهای دیگر:

GL\_LINEAR\_ATTENUATION and GL\_QUADRATIC\_ATTENUATION.

پیش فرض:  $k_q=0$  و  $k_l=0$ ,  $k_c=1$

### مدل نورافشانی

OpenGL اجازه نشان دادن سه پارامتر را برای مشخص کردن قوانین کلی برای اعمال مدل نورافشانی می دهد.

الف: رنگ نور وافر جهانی

می توان یک نور وافر جهانی در صحنه مستقل از هر منبعی داشت.

```
GLfloat amb[] = {0.2, 0.3, 0.1, 1.0};
glLightModelfv(GL_LIGHT_MODEL_AMBIENT, amb);
```

مقدار پیش فرض: (0.2,0.2,0.2,1.0). باعث می شود که اشیاء حتی اگر هیچیک از توابع نوری صدا زده نشده باشند دیده شوند.

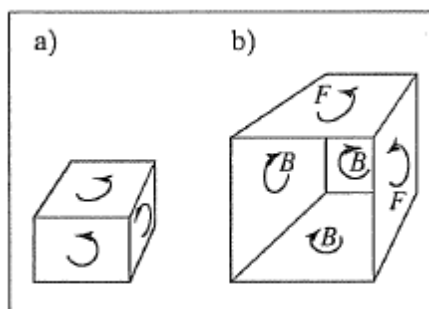
ب: نقطه دید محلی است یا دور

OpenGL انعکاس درخشنده را با استفاده از بردار  $h$  محاسبه می کند. مقادیر واقعی  $S$  و  $V$  برای هر نقطه تفاوت می کنند. اگر نور جهتی باشد  $S$  ثابت است ولی هنوز  $V$  تغییر می کند. اگر  $V$  را برای همه رؤس ثابت در نظر بگیریم سرعت نمایش افزایش می یابد. این حالت پیش فرض بوده و OpenGL بردار  $v=(0,0,1)$  را در نظر می گیرد. اگر بخواهیم مقدار واقعی حساب شود:

```
glLightModeli(GL_LIGHT_MODEL_LOCAL_VIEWER, GL_TRUE);
```

ج: آیا هر دو سمت یک وجه بدرستی سایه زنی می شوند؟

هر وجه چندضلعی دو رو دارد. معمولاً رؤس CCW چنانکه از خارج منحنی نگریسته شوند لیست می شوند. معمولاً دوربین خارج یک شیء را می بیند. OpenGL فقط وجه جلو و وجه عقب می شناسد. یک وجه جلو رؤس CCW لیست شده چنانکه با چشم دیده می شود.



شکل ۱۷

```
glLightModeli(GL_LIGHT_MODEL_TWO_SIDE, GL_TRUE);
```

## خواص مواد در OpenGL

اثر یک منبع نور هنگامی دیده می شود که از سطح شیء منعکس شود. می توان ضرائب انعکاس را توسط دستور `glMaterial()` معین نمود.

مثال:

```
GLfloat myDiffuse[] = {0.8, 0.2, 0.0, 1.0};
glMaterialfv(GL_FRONT, GL_DIFFUSE, myDiffuse);
(pdr, pd $\bar{g}$ , pd $\bar{b}$ ) = (0.8, 0.2, 0.0)
```

در فرمت RGBA

پارامتر اول:

GL\_FRONT: set the reflection coefficient vector for front faces  
 GL\_BACK: set it for back faces  
 GL\_FRONT\_AND\_BACK: set it for both front and back faces

پارمتر دوم:

GL\_AMBIENT: set the ambient reflection coefficients  
 GL\_DIFFUSE: set the diffuse reflection coefficients  
 GL\_SPECULAR: set the specular reflection coefficients  
 GL\_AMBIENT\_AND\_DIFFUSE: set both the ambient and diffuse reflection coefficients to the same values. This is for convenience, since the ambient and diffuse coefficients are so often chosen to be the same.  
 GL\_EMISSION: set the emissive color of the surface.

محاسبه رنگ یک نقطه در OpenGL:

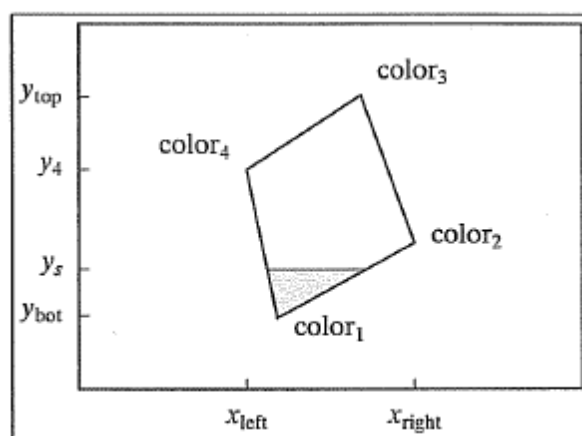
$$I_r = e_r + I_{mr} \rho_{ar}$$

$$+ \sum_i \text{atten}_i \times \text{spot}_i \times (I_{ar}^i \rho_{ar} + I_{dr}^i \rho_{dr} \times \text{lambert}_i + I_{spr}^i \rho_{sr} \times \text{phong}_i)$$

$e_r$ : نور متصاعد شده قرمز، و  $I_{mr}$ : نور وافر جهانی می باشد. پس از آن روی اثر همه منابع نوری جمع گرفته می شود. برای سبز و آبی هم بطور مشابه محاسبه می شود.

### مدلهای سایه زنی برای چندضلعیها

امکان سایه زنی هر نقطه به روشهای گفته شده وجود دارد ولی بسیار زمانبر است. برای چند ضلعیها می توان از روشهای ساده تری استفاده نمود. یک چندضلعی بصورت خط به خط (از یک خط پیمایشی به یک خط پیمایشی دیگر) ترسیم می شود. برای هر پیکسل در هر خط پیمایشی باید رنگ آن محاسبه شده و نمایش داده شود. تفاوت روشهای مختلف در چگونگی یافتن رنگ هر پیکسل می باشد.



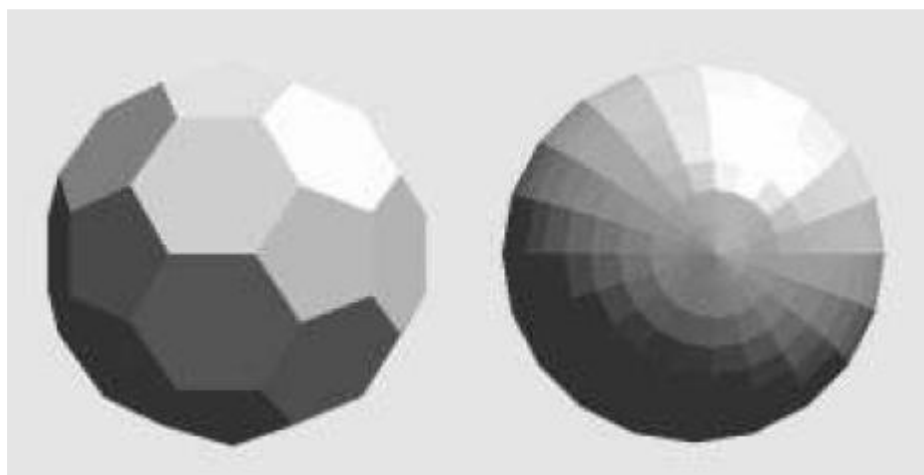
شکل ۱۸

```
for (int y = y_bot; y <= y_top; y++) // for each scanline
{
    <.. find x_left and x_right ..>
    for (int x = x_left; x <= x_right; x++) // for each relevant pixel
        across this scanline
        {
            <.. find the color c for this pixel ..>
            <.. put c into the pixel at (x, y) ..>
        }
}
```

### سایه زنی ثابت

در این حالت برای همه نقاط هر چندضلعی یک شدت یکسان در نظر گرفته می شود.

`glShadeModel(GL_FLAT);`



شکل ۱۹

همانگونه که مشاهده می شود کره بصورت تکه تکه نشان داده شده است.

### سایه زنی نرم

سایه زنی نرم تلاش در محو کردن لبه های بین وجوه بوسیله محاسبه رنگ در نقاط بیشتری از هر سطح می نماید. یکی از روشها سایه زنی گوراد<sup>۱</sup> می باشد. در این روش، رنگ پیکسلهای چپ و راست هر خط پیمایشی بصورت درونیابی یافته شده و سپس رنگ هر پیکسل در هر خط پیمایشی بصورت درونیابی بین رنگهای چپ و راست محاسبه می شود. برای افزایش کارآئی می توان رنگ هر پیکسل را بصورت افزایشی بر اساس مقدار رنگ پیکسل قبلی محاسبه نمود:

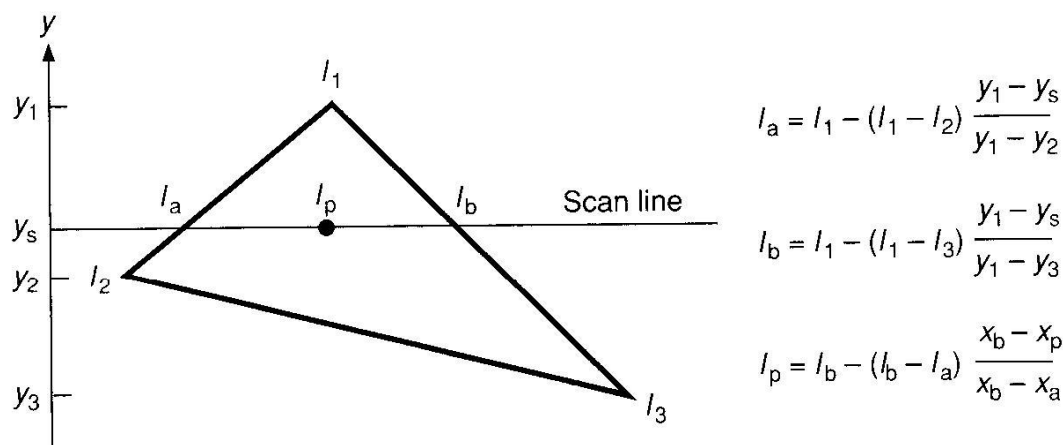
$$c(x + 1) = c(x) + \frac{color_{right} - color_{left}}{x_{right} - x_{left}}$$

<sup>۱</sup> Gouraud Shading

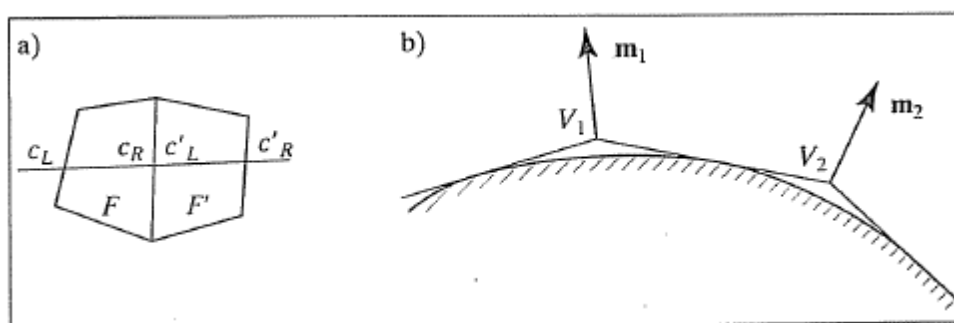


OpenGL از روش سایه زنی گوراد استفاده می کند.

`glShadeModel(GL_SMOOTH);`



**Fig. 16.19** Intensity interpolation along polygon edges and scan lines.

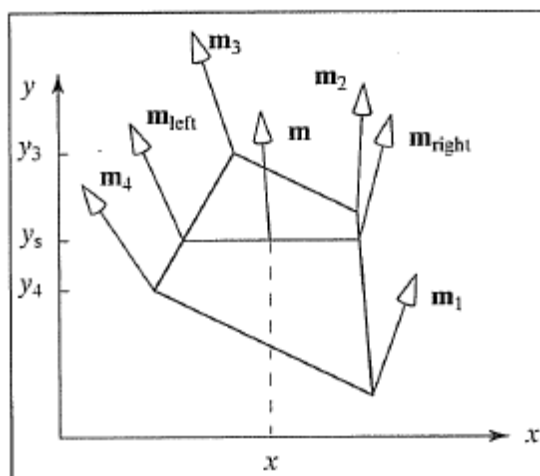


شکل ۲۰

هنگام سایه زنی سطوح نرم، عمودی که به رأس نسبت داده می شود بر اساس عمود واقعی که به سطح نرم عمود است محاسبه می شود نه عمودی که به چندضلعی تقریب زننده عمود است. سایه زنی گوراد نوربالاها را خوب محاسبه نمی کند.

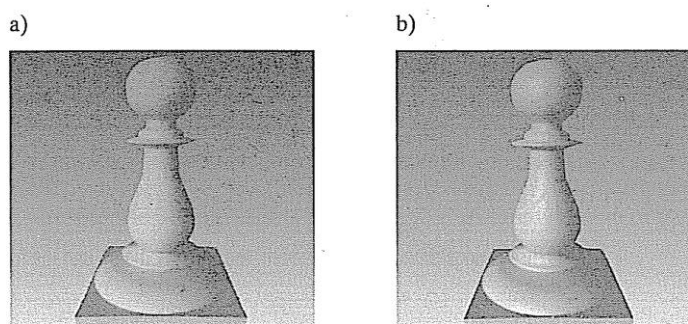
### سایه زنی فانگ

در روش فانگ عمود چپ و راست در هر خط پیمایشی بوسیله درونیایی یافته شده و در هر پیکسل در هر خط پیمایشی هم عمود بوسیله درونیایی بین عمودهای چپ و راست بدست می آید. رنگ هر نقطه با استفاده از این عمودها محاسبه می شود. این باعث می شود نوربالاها بهتر نمایش داده شوند.



شکل ۲۱

شکل ۲۲ تفاوت بین سایه زنی گوراد (a) و فانگ (b) را نشان می دهد.



شکل ۲۲

مشکل سایه زنی فانگ زمانبر بودن بیشتر آن نسبت به روش گوراد می باشد.

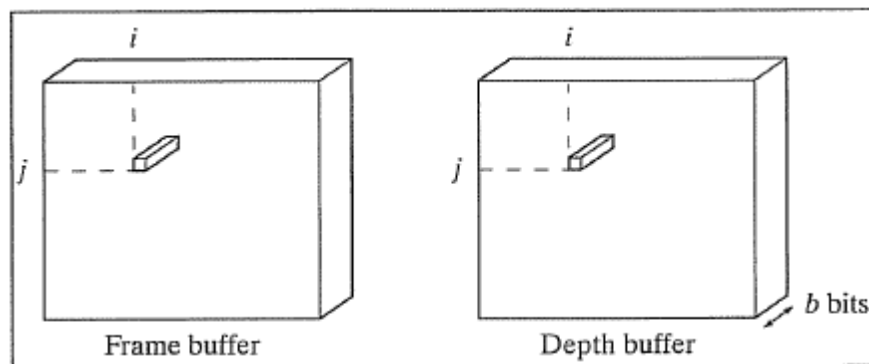
باید دقت کرد که OpenGL با مکان و جهت نور همانند مکان هر شیء دیگری برخورد می کند. تبدیلهای بر روی مکان و جهت نور اثر دارند.

### حذف سطح مخفی

حذف سطوح مخفی باعث واقعیت به نظر رسیدن اشیاء می شود. روشهای مختلف برای حذف سطوح مخفی وجود دارد. یکی از ساده ترین آنها روش میانگیر عمق یا میانگیر-Z می باشد.

### میانگیر عمق

در هنگام نمایش مقدار میانگیر عمق  $d[i][j]$  شامل شبه عمق نزدیکترین شیئی است که تاکنون در آن پیکسل با آن مواجه شده ایم. هنگام نمایش هر نقطه از هر وجه چک می شود که آیا شبه عمق آن نقطه کمتر از عمق ذخیره شده برای آن نقطه (پیکسل) در میانگیر عمق هست یا نه؟ اگر بله - رنگ آن نقطه جایگزین رنگ  $p[i][j]$  شده و شبه عمق جدید نیز در  $d[i][j]$  ذخیره می گردد.



شکل ۲۳

وجوه می توانند به هر ترتیبی رسم شوند. اگر یک وجه دورتر زودتر رسم شود، رنگ برخی از نقاط آن توسط رنگ وجوهی که بعداً جلوتر هستند جایگزین می شوند. اگر وجه جلوتر زودتر رسم شود، وجوه عقبتر از آن اصلاً رسم نمی شوند.

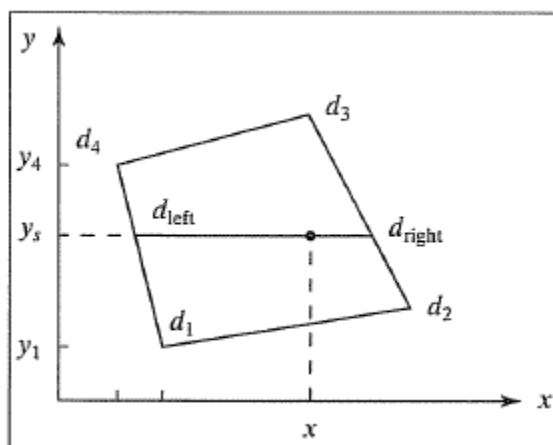
### یافتن شبه عمق

نقطه  $P=(P_x, P_y, P_z)$  طی عبور از خط لوله OpenGL تبدیلهای متعددی را متحمل می شود.

پس از تبدیل بندردید داریم:

$$(x, y, z) = \left( \frac{P_x}{-P_z}, \frac{P_y}{-P_z}, \frac{aP_z + b}{-P_z} \right)$$

مؤلفه سوم شبه عمق است.  $a$  و  $b$  به گونه ای تنظیم می شوند که اگر  $P$  روی صفحه جلو باشد مقدارش صفر و اگر روی صفحه عقب باشد مقدارش ۱ باشد. عمق هر نقطه بصورت درونیابی و افزایشی یافته می شود.



شکل ۲۴

OpenGL میانگیر عمق را پشتیبانی می کند. برای گفتن به OpenGL برای ایجاد میانگیر عمق

```
glutInitDisplayMode(GLUT_DEPTH | GLUT_RGB);
```

و آزمون عمق را نیز باید فعال کرد:

```
glEnable(GL_DEPTH_TEST);
```

هر گاه که تصویر جدیدی ایجاد می شود میانگیر عمق باید پاک شود (مقدار دهی اولیه شود)

```
glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT); // clear  
screen
```

### مراحل افزودن روشنایی به صحنه:

- تعیین بردارهای عمود برای هر رأس هر شیء. (این بردارها تمایل شیء نسبت به منبع نور را مشخص می کنند)
- ایجاد، انتخاب، و مکاندهی یک یا بیشتر منبع نور
- ایجاد و انتخاب یک مدل نورافشانی (جهت تعیین سطح نور وافر جهانی و مکان مؤثر بیننده برای محاسبات نورپردازی)
- معین کردن خواص ماده برای اشیاء درون صحنه

مثال:

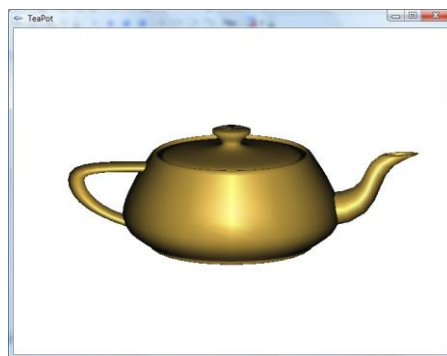
```

// In His Name the Most High
// This program draws teapot.
// Programmer : Maziar Palhang
#include <windows.h>
#include <gl/gl.h>
#include <gl/glu.h>
#include <gl/glut.h>
void myInit() {
    glClearColor(1.0f, 1.0f, 1.0f, 0.0f);
    glViewport(0, 0, 640, 480);
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    glEnable(GL_DEPTH_TEST);
    // glShadeModel(GL_FLAT);
    GLfloat position[] = {0.0f, 0.0f, 3.0f, 1.0f};
    glLightfv(GL_LIGHT0, GL_POSITION, position);

    //ظرایب طلا
    GLfloat mat_amb[] = {0.24725f, 0.1995f, 0.0745f};
    GLfloat mat_diff[] = {0.75164f, 0.60648f, 0.22648f};
    GLfloat mat_spec[] = {0.628281f, 0.555802f, 0.366065f};
    GLfloat mat_phong[] = {51.2f};
    //ظرایب مس
    // GLfloat mat_amb[] = {0.19125f, 0.0735f, 0.0225f};
    // GLfloat mat_diff[] = {0.7038f, 0.27048f, 0.0828f};
    // GLfloat mat_spec[] = {0.256777f, 0.137622f, 0.086014f};
    // GLfloat mat_phong[] = {12.8f};
    glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_amb);
    glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diff);
    glMaterialfv(GL_FRONT, GL_SPECULAR, mat_spec);
    glMaterialfv(GL_FRONT, GL_SHININESS, mat_phong);
}
void myDisplay() {
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-2, 2, -2, 2, 0.1, 20);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(0.0, 1.0, 3.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glColor3d(1.0, 0.0, 0.0);
    glutSolidTeapot(1.0); //رسم بصورت جسم صلب نه قاب سیمی
    glFlush();
}
int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(640, 480);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("TeaPot");
    glutDisplayFunc(myDisplay);
    myInit();
    glutMainLoop();
    return 0;
}

```



شکل ۲۵

باید دقت کرد که هر گاه دستورهای مشخص کردن خواص ماده صادر می شوند تا وقتی تغییر داده نشده اند برای همه اشیاء تا قبل از تغییر همان خواص استفاده می شود. لذا اگر اشیاء با خواص مختلفی در صحنه حضور دارند باید خواص ماده هر یک را بطور مجزا مشخص کرد. خواص هر منبع نور نیز بصورت جداگانه مشخص می شود.

### تمرین

- ۱- برنامه ای بنویسید که یک میز که بر روی آن یک چراغ مطالعه قرار گرفته است و سطح آن را روشن می نماید را نمایش دهد.
- ۲- بررسی نمائید که چگونه بصورت افزایشی می توان مقدار  $C(y+1)$  را از روی مقدار  $C(y)$  در روش سایه زنی گوراد بدست آورد.