

# Компьютерная графика

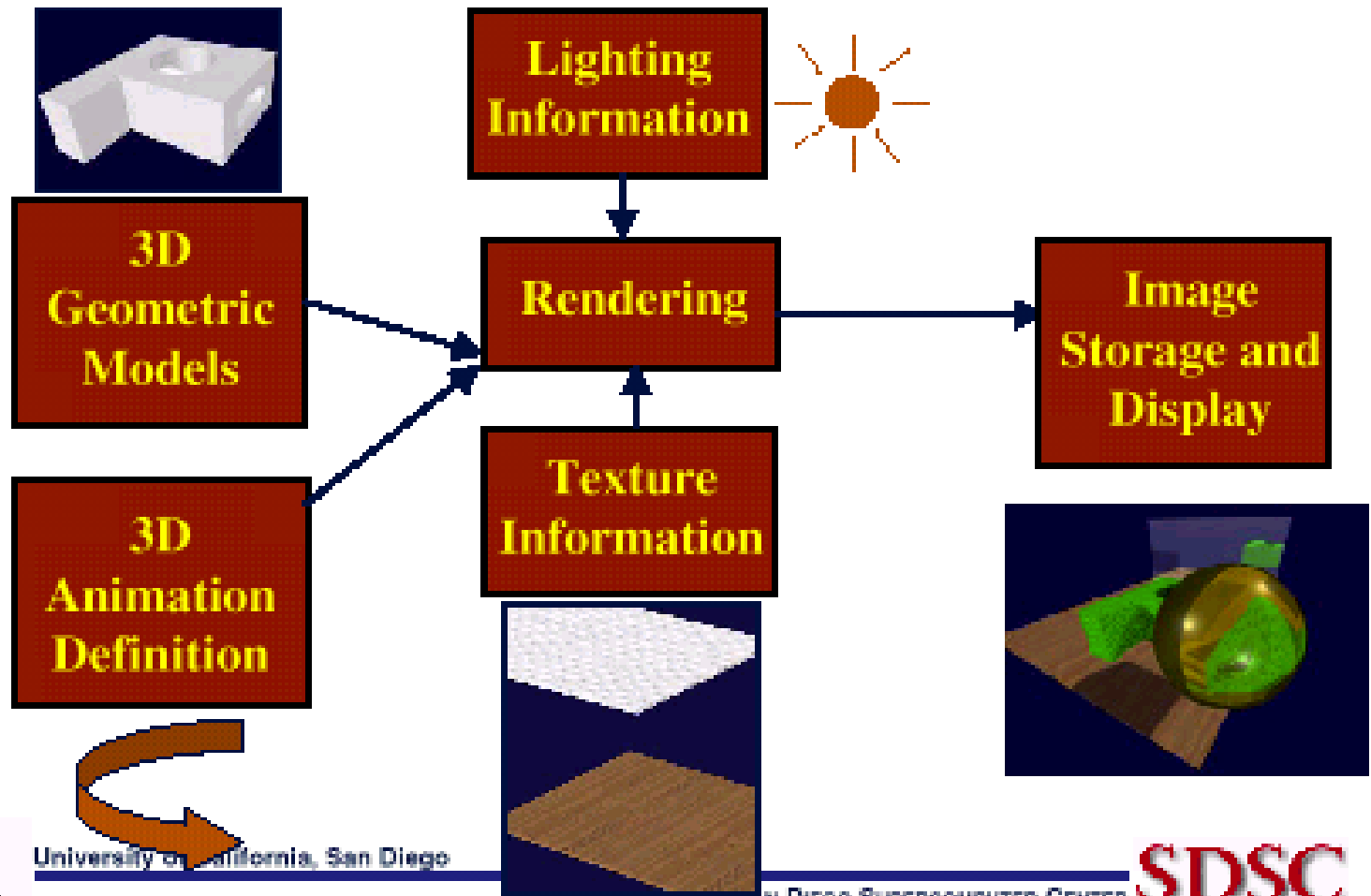
---

## Лекция 9

# Локальные и глобальные модели освещения. Закраска Гуро и Фонга

7 апреля 2005 года

# The Graphics Process



University of California, San Diego

SAN DIEGO SUPERCOMPUTER CENTER

**SDSC**

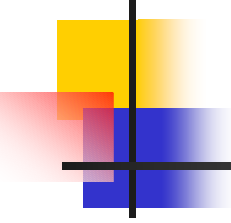


# Экранизация (Rendering)

---

Realistic displays of a scene are obtained by generating perspective projections of objects and by applying natural lighting effects to the visible surfaces. An illumination model, also called a lighting model and sometimes referred to as a shading model, is used to calculate the intensity of light that we should see at a given point on the surface of an object.

Photorealism in computer graphics involves two elements: accurate graphical representations of objects and good physical descriptions of the lighting effects in a scene. Lighting effects include light reflections, transparency, surface texture, and shadows.



# Задачи моделирования освещения

---

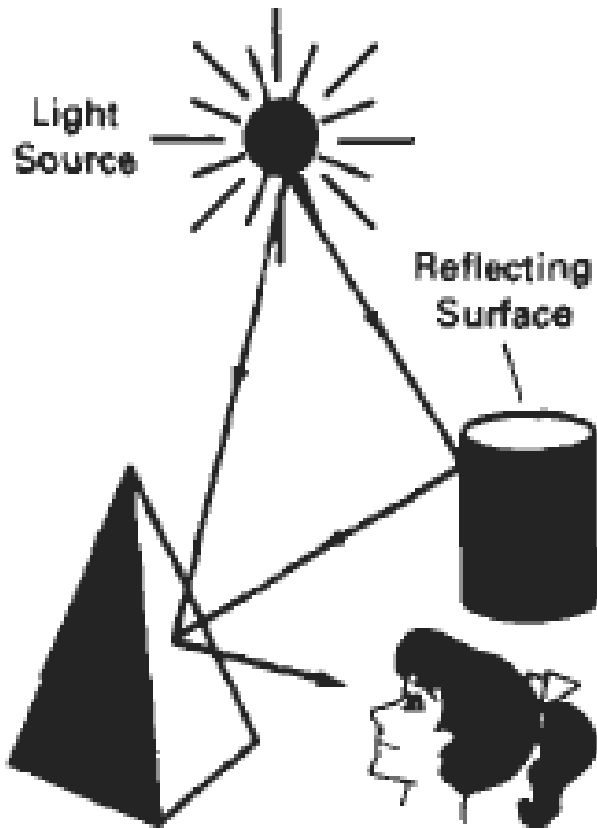
Моделирование освещения - ключевой элемент создания фотореалистичных изображений

- глаз воспринимает освещение и «реконструирует» трехмерную форму

В графике необходимо моделировать:

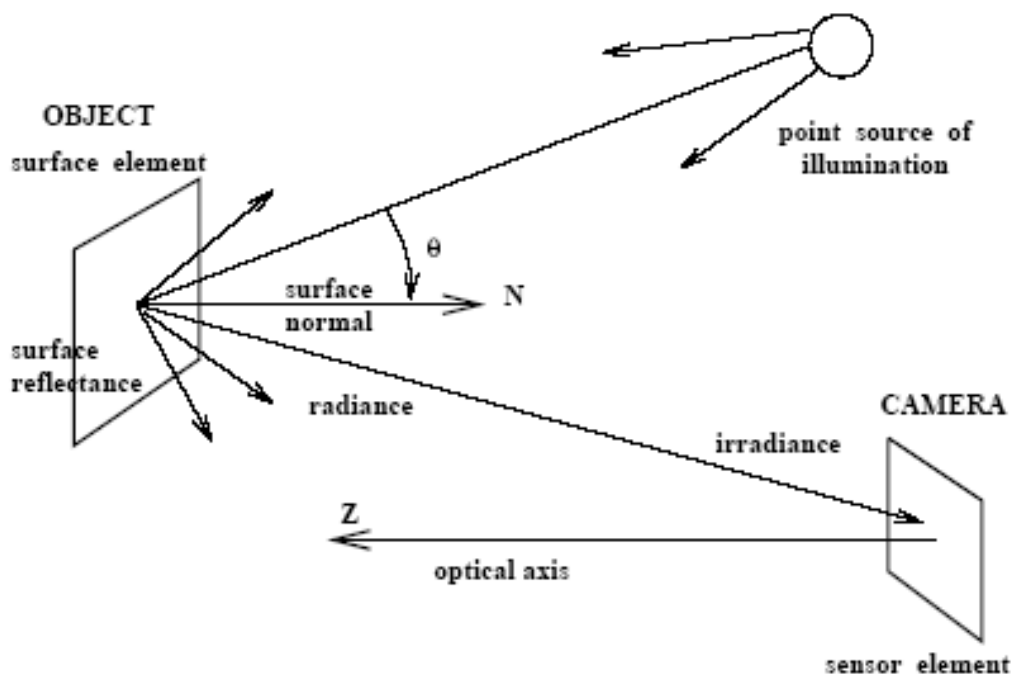
- свойства источника световой энергии
- отражающие, преломляющие и т.п. свойства поверхностей и сред
- свойства сетчатки глаза

# Источники света



When we view an opaque nonluminous object, we see reflected light from the surfaces of the object. The total reflected light is the sum of the contributions from light sources and other reflecting surfaces in the scene. Thus, a surface that is not directly exposed to a light source may still be visible if nearby objects are illuminated. Sometimes, light sources are referred to as **light-emitting sources**; and reflecting surfaces, such as the walls of a room, are termed **light-reflecting sources**. We will use the term **light source** to mean an object that is emitting radiant energy, such as a light bulb or the sun.

# Отражение света



Modeling the colors and lighting effects that we see on an object is a complex process, involving principles of both physics and psychology. Fundamentally, lighting effects are described with models that consider the interaction of electromagnetic energy with object surfaces. Once light reaches our eyes, it triggers perception processes that determine what we actually "see" in a scene.



# Освещение и отражение

---

## Illumination and Reflection





# Освещение и отражение

---

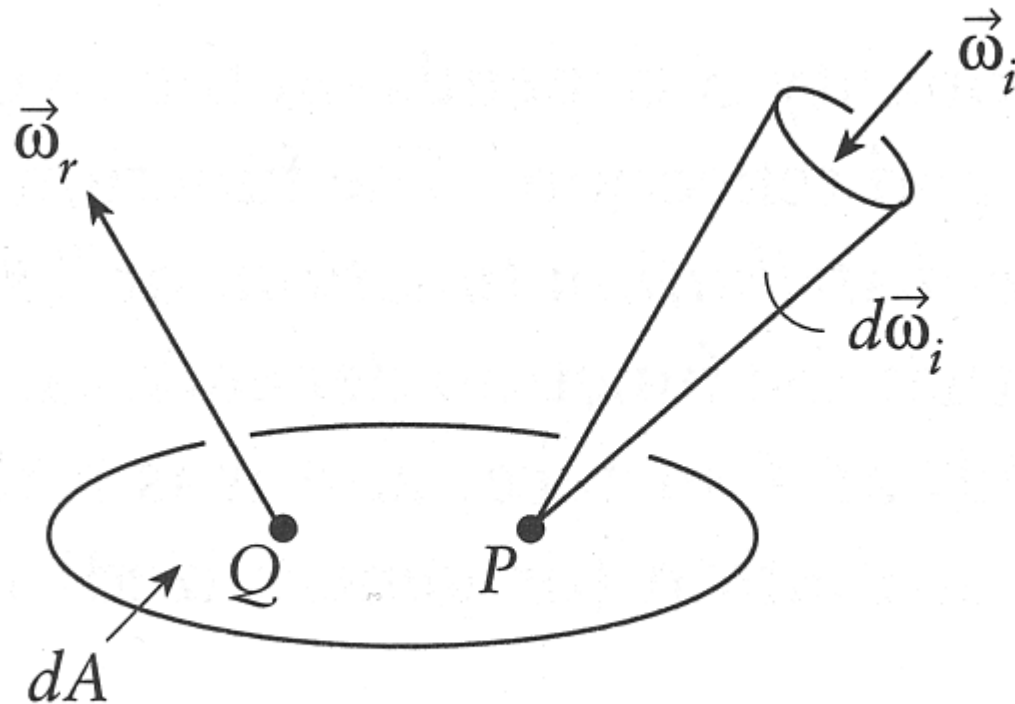
В реальном мире видимый цвет определяется количеством энергии видимого спектра, которая попадает на чувствительные элементы сетчатки глаза

Как правило, мы видим **отраженный** свет, т.е. энергию, отраженную материалом поверхности.

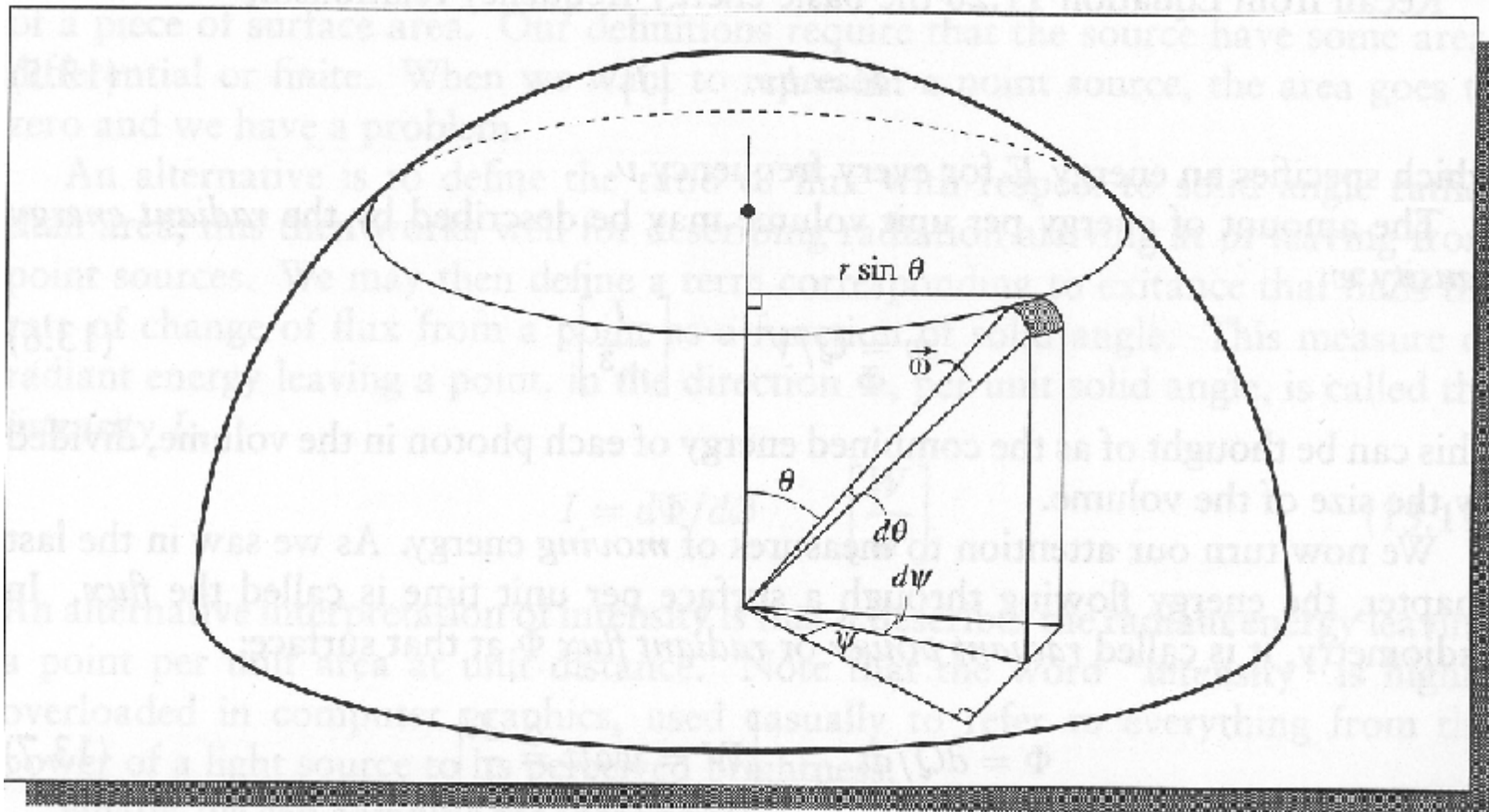
Отражающие характеристики поверхности определяются отражающими способностями по отношению к волнам различной длины



# BRDF: bidirectional reflectance distribution function

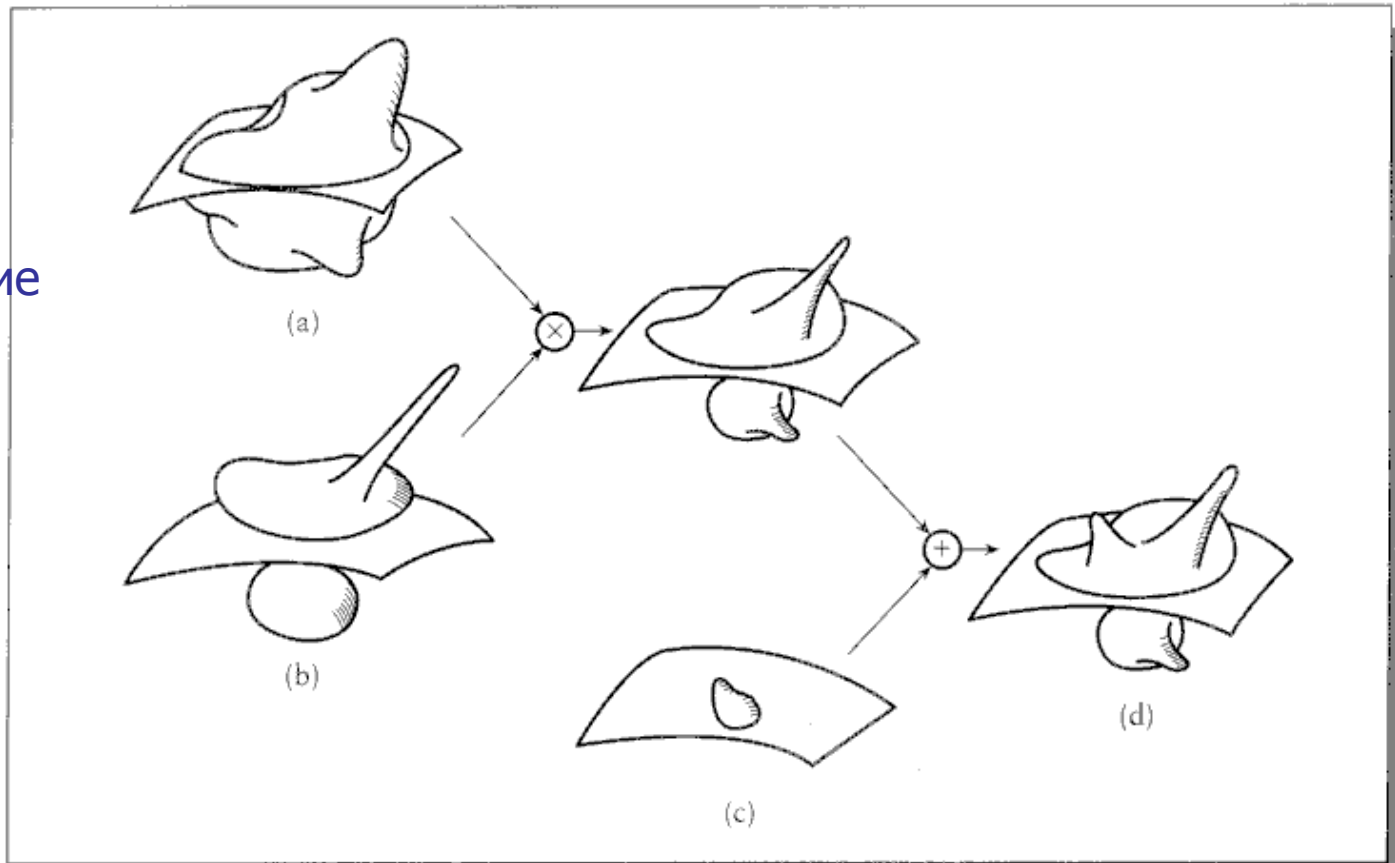


# Двухнаправленная функция распределения отражения

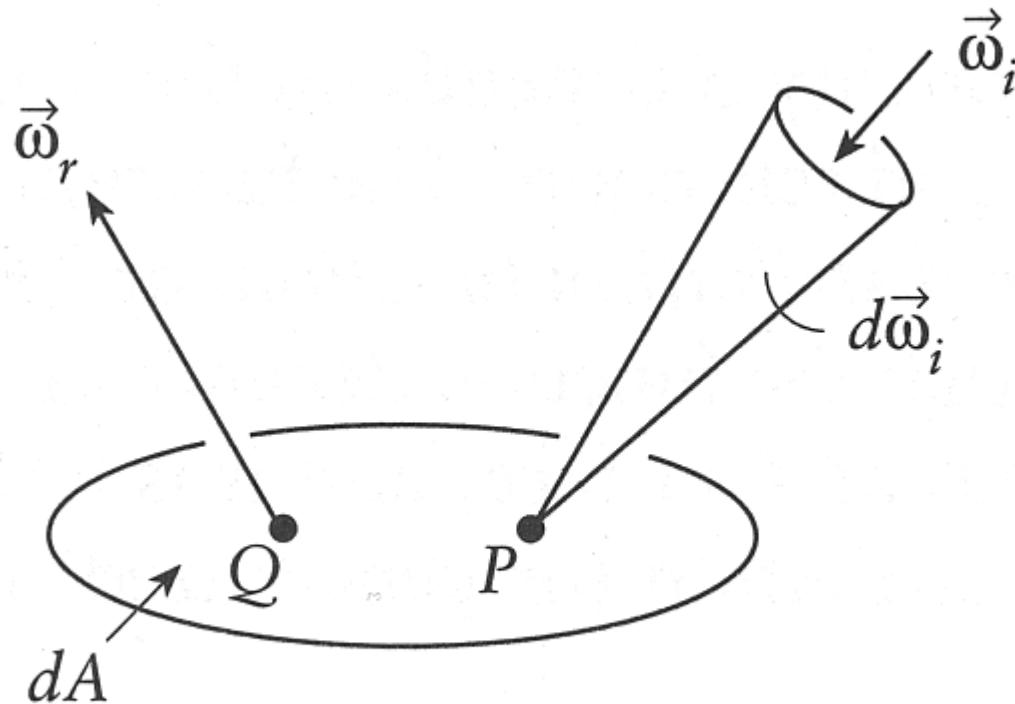


# Функции освещения и отражения

- (a) Освещение  
Illumination
- (b) Светорассеивание  
BRDF
- (c) Эмиссия  
Emission
- (d) Излучение  
Radiance



# BRDF: bidirectional reflectance distribution function





# Уравнение отражения

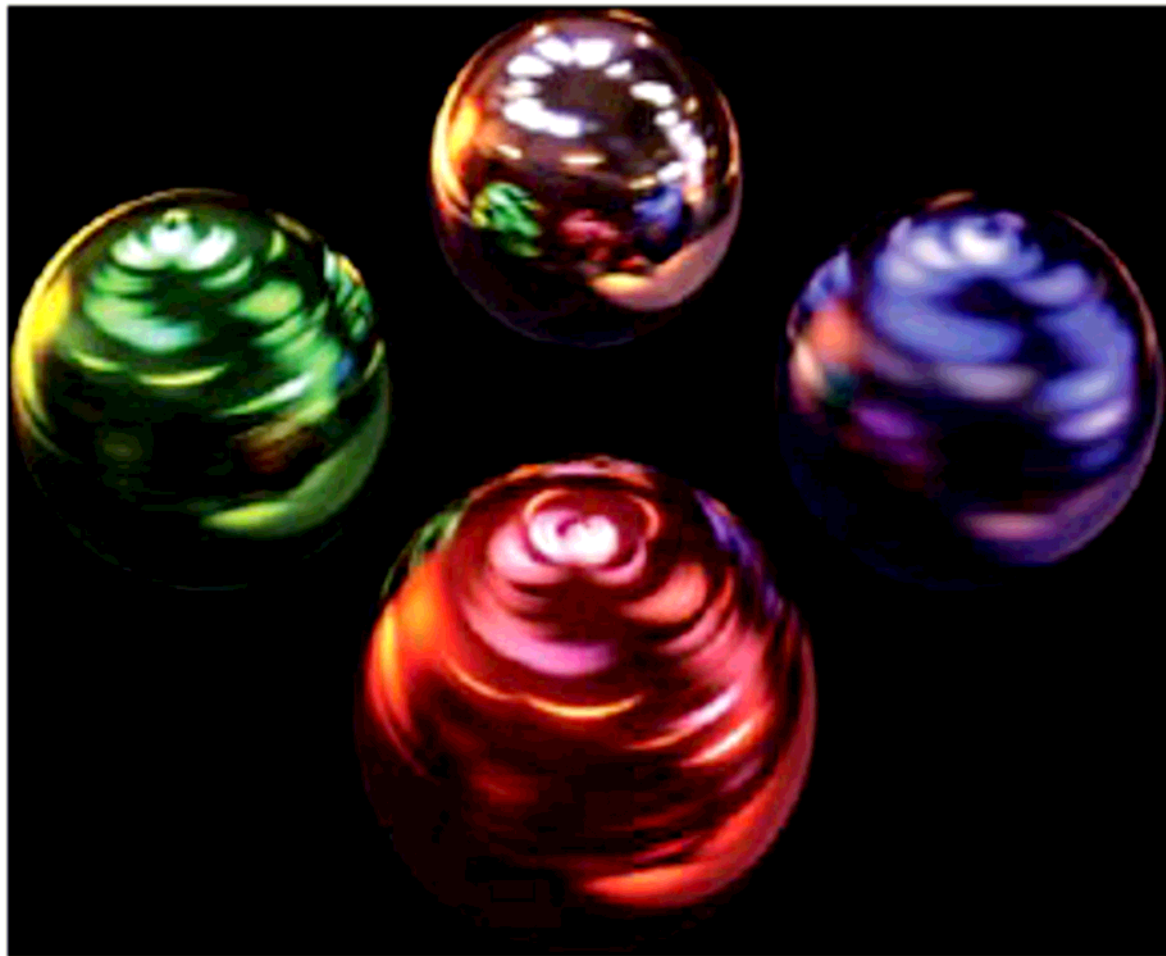
---

$$\begin{aligned} L^r(\vec{\omega}_r) &= \int_{\Omega_i} f_r(\vec{\omega}_i \rightarrow \vec{\omega}_r) L^i(\vec{\omega}_i) d\vec{\omega}_i^N \\ &= \int_{\phi_i=0}^{2\pi} \int_{\theta_i=0}^{\pi/2} f_r((\theta_i, \psi_i) \rightarrow (\theta_r, \psi_r)) L^i(\theta_i, \psi_i) \cos \theta_i \sin \theta_i d\theta_i d\psi_i \end{aligned}$$

$$dA = (r \sin \theta d\psi)(r d\theta) = r^2 \sin \theta d\theta d\psi$$

# Анизотропная функция светорассеяния (BRDF)

- (a) Точность  
Accuracy
- (b) Выразительность  
Expressiveness
- (c) Скорость  
Speed





# Локальные и глобальные модели

---

Источниками энергии могут быть не только источники света, но и другие отражающие объекты (и даже части того же самого объекта)

Такие взаимоотношения сложно учитывать, поэтому выделяют **локальные** и **глобальные** модели освещения

**Локальные модели** при вычислении освещения в данной точке учитывают только положение этой точки относительно первичных источников света

# Локальные и глобальные модели

Illumination models in computer graphics are often loosely derived from the physical laws that describe surface light intensities. To minimize intensity calculations, most packages use empirical models based on simplified photometric calculations. More accurate models, such as the radiosity algorithm, calculate light intensities by considering the propagation of radiant energy between the surfaces and light sources in a scene. We first take a look at the basic illumination models often used in graphics packages; then we discuss more accurate, but more time-consuming, methods for calculating surface intensities.





# Типы функций отражения

---

## Диффузное отражение

- матовый пластик, дерево и т.п.
- модель Ламберта

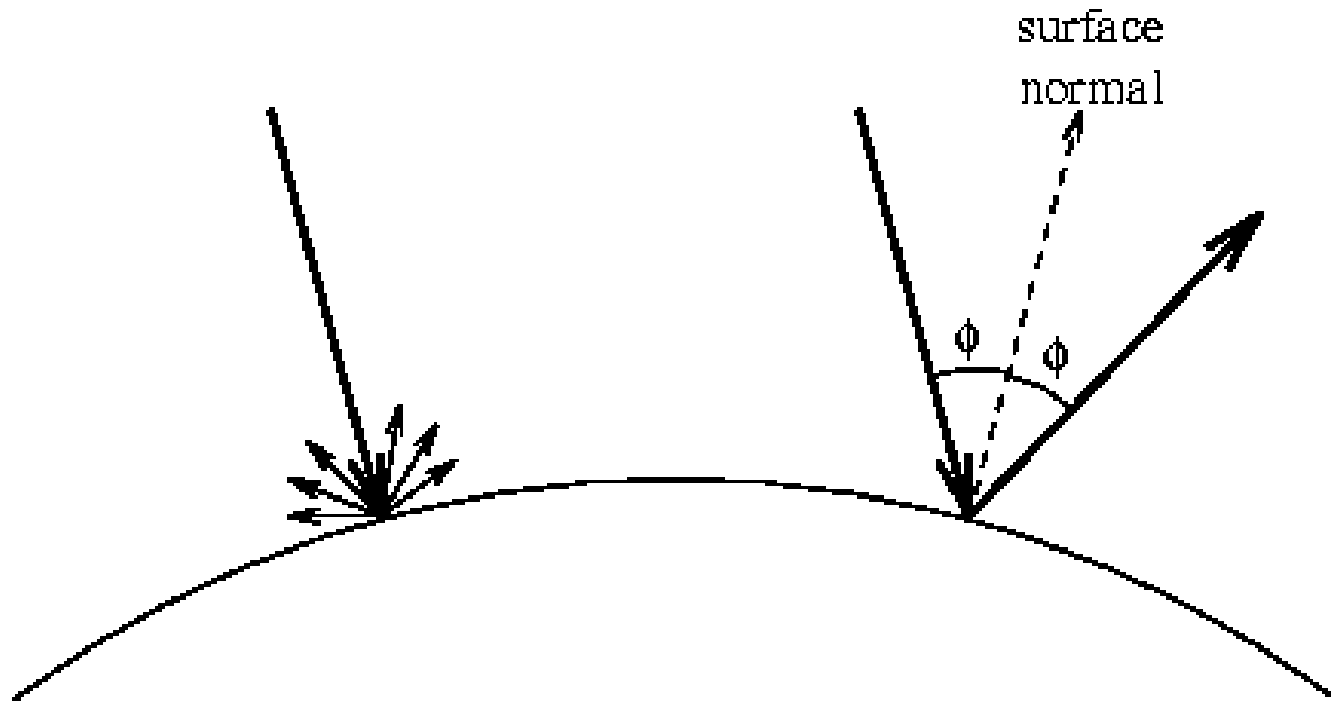
## Идеально зеркальное отражение

- зеркало
- модель отражения

## Зеркальное отражение

- блики на объекте
- модели Фонга и Блинна

# Диффузное и зеркальное отражение



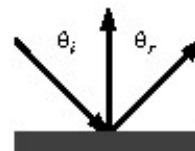
# Функции отражения

## Types of Reflection Functions

---

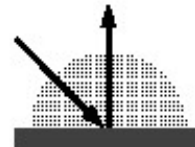
### Ideal Specular

- Mirror
- Reflection Law



### Ideal Diffuse

- Matte
- Lambert's Law

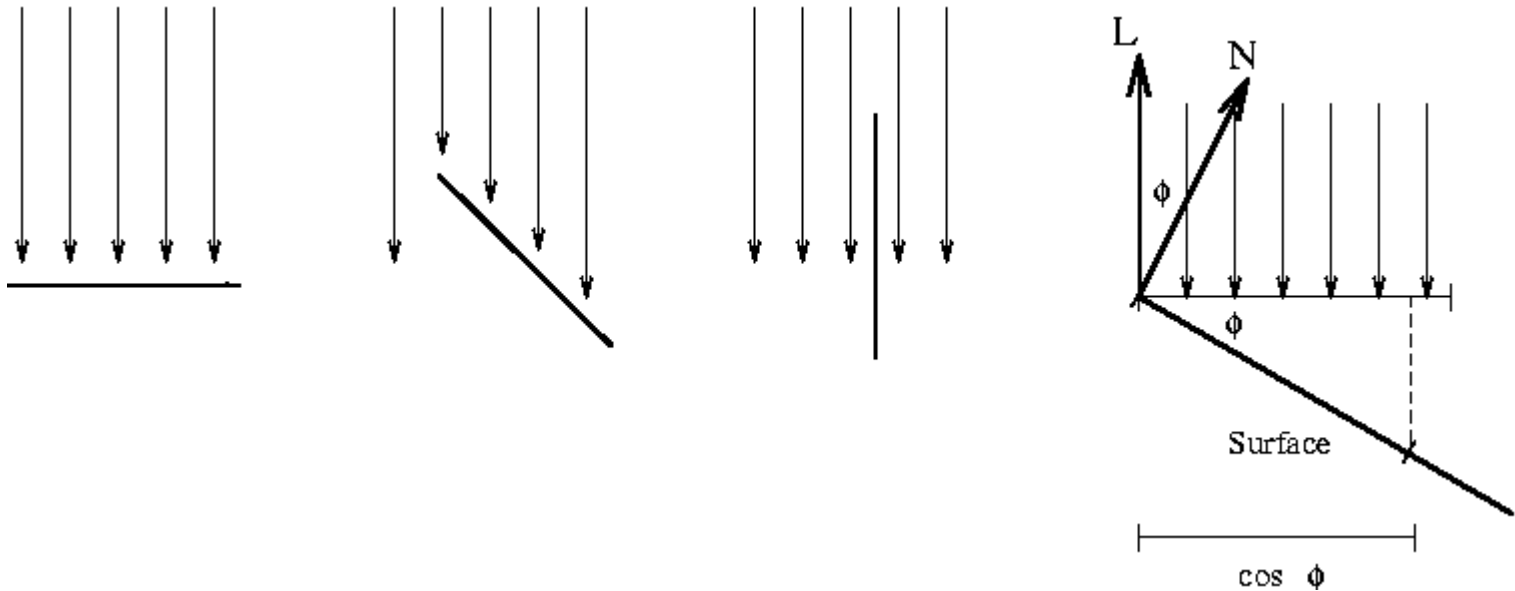


### Specular

- Glossiness and Highlights
- Phong and Blinn Models



# Диффузное отражение



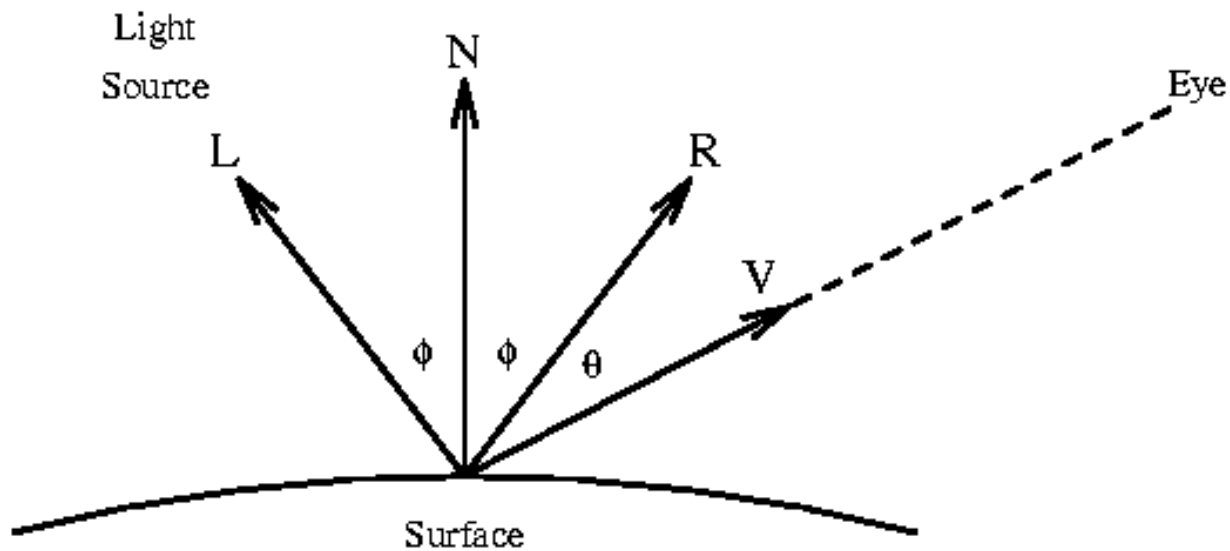
$$I_r = k_{dr} \frac{I_{pr}}{d} \cos \phi$$

$$I_g = k_{dg} \frac{I_{pg}}{d} \cos \phi$$

$$I_b = k_{db} \frac{I_{pb}}{d} \cos \phi$$

# Зеркальное отражение

## Phong's Rule



$$I_r = I_{pr} k_s (\cos \theta)^n \quad I_g = I_{pg} k_s (\cos \theta)^n \quad I_b = I_{pb} k_s (\cos \theta)^n$$

# Определение видимых поверхностей (рассеянный “ambient” свет)



# Уравнение освещенности Фонга (модель Фонга)

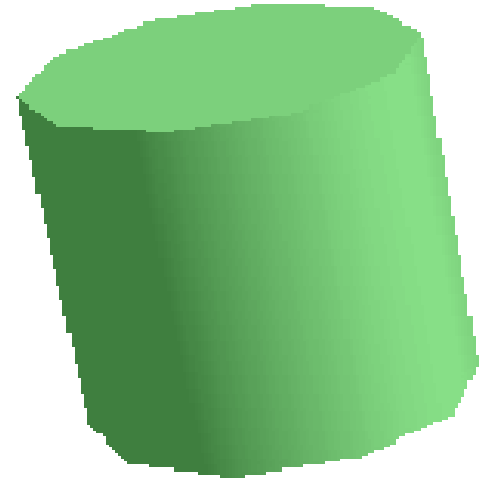
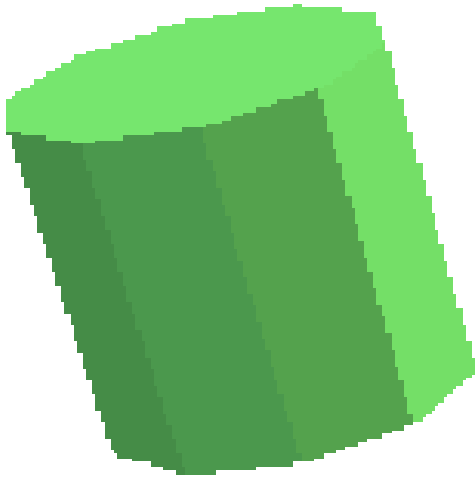
$$I_r = I_{ar}k_{df} + \frac{I_{pr}}{d}(k_{df}(L \cdot N) + k_s(R \cdot V)^n)$$

$$I_g = I_{ag}k_{df} + \frac{I_{pg}}{d}(k_{df}(L \cdot N) + k_s(R \cdot V)^n)$$

$$I_b = I_{ab}k_{df} + \frac{I_{pb}}{d}(k_{df}(L \cdot N) + k_s(R \cdot V)^n)$$

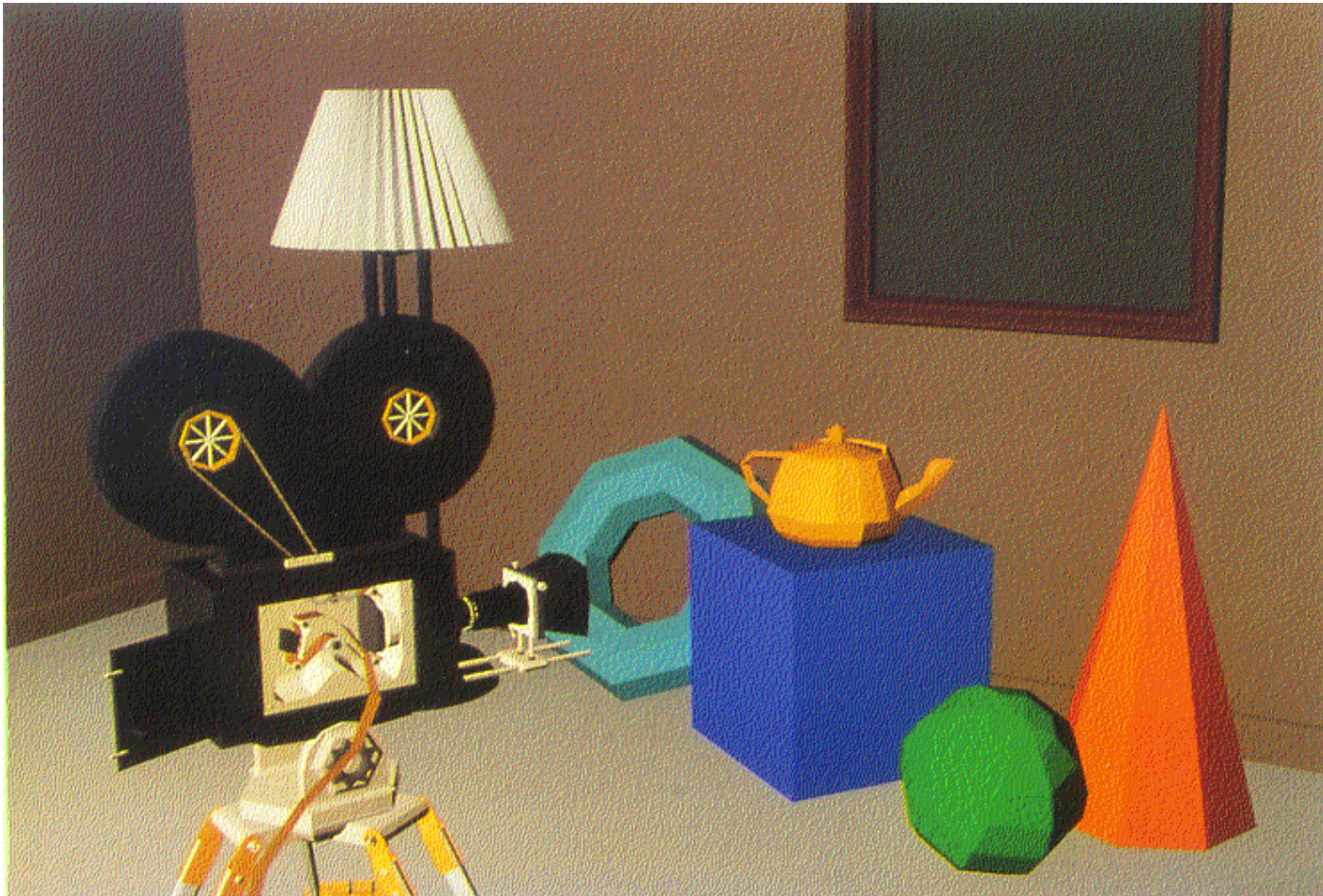
# Интерполяционная закрашка (Interpolative Shading)

- Аппроксимация объекта плоскими гранями
- Вычисление нормалей
- Гладкая интерполяция нормалей

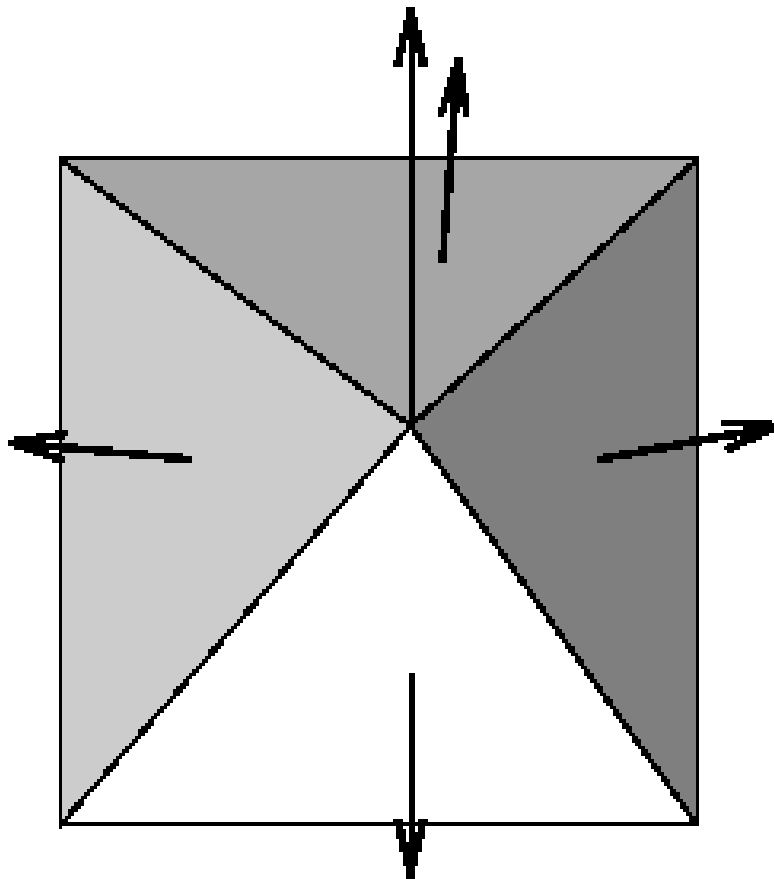




# Закраска индивидуальных полигонов (плоская “flat” закразка)

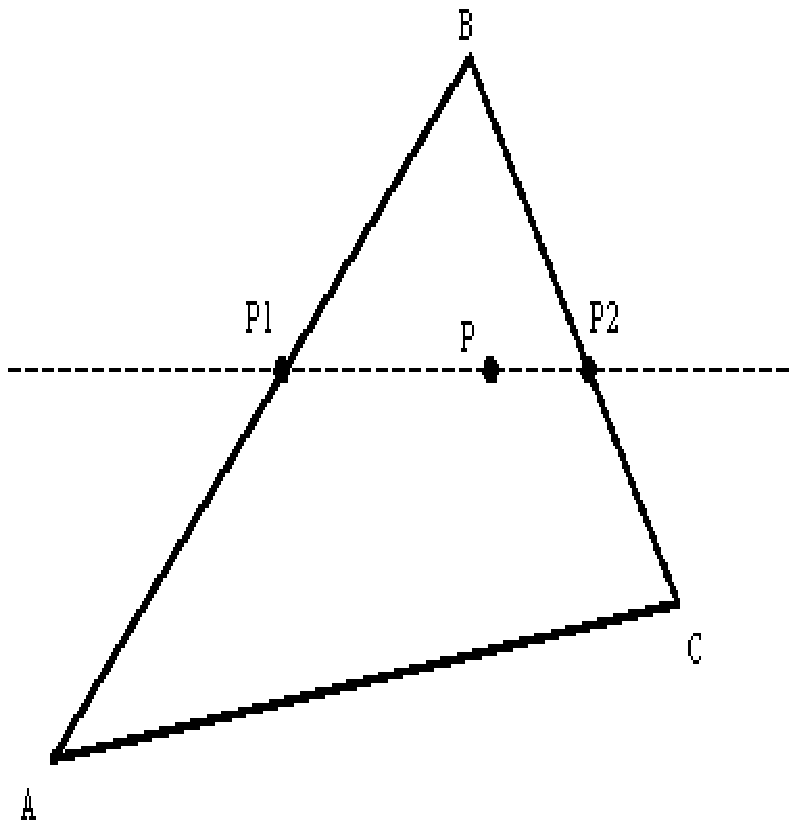


# Вычисление нормалей



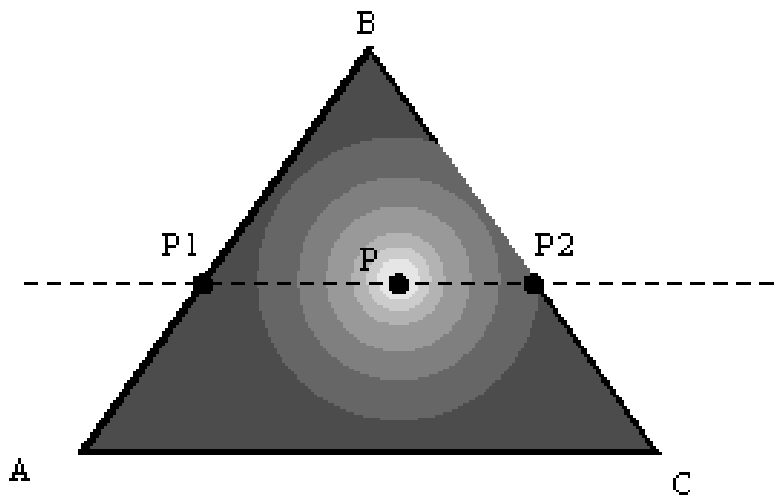
- Нормали в вершинах вычисляются усреднением нормалей смежных граней

# Интерполяция цвета

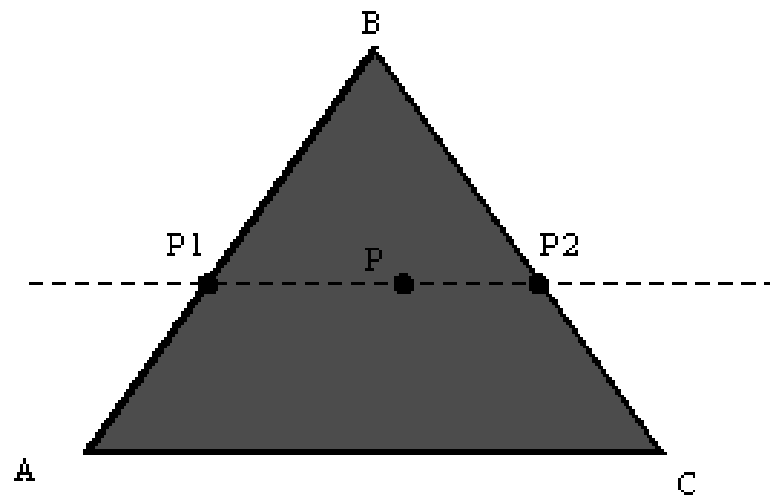


- Вычислить цвет (RGB) в каждой вершине.
- Вычислить цвет в точках P1 и P2:  
 $s = \|P1 - B\| / \|A - B\|$   
 $C(P1) = s(C(A)) - (1-s)(C(B))$
- Вычислить цвет в т. P:  
 $s = \|P - P2\| / \|P1 - P2\|$   
 $C(P) = s(C(P1)) - (1-s)(C(P2))$

# Исчезают блики

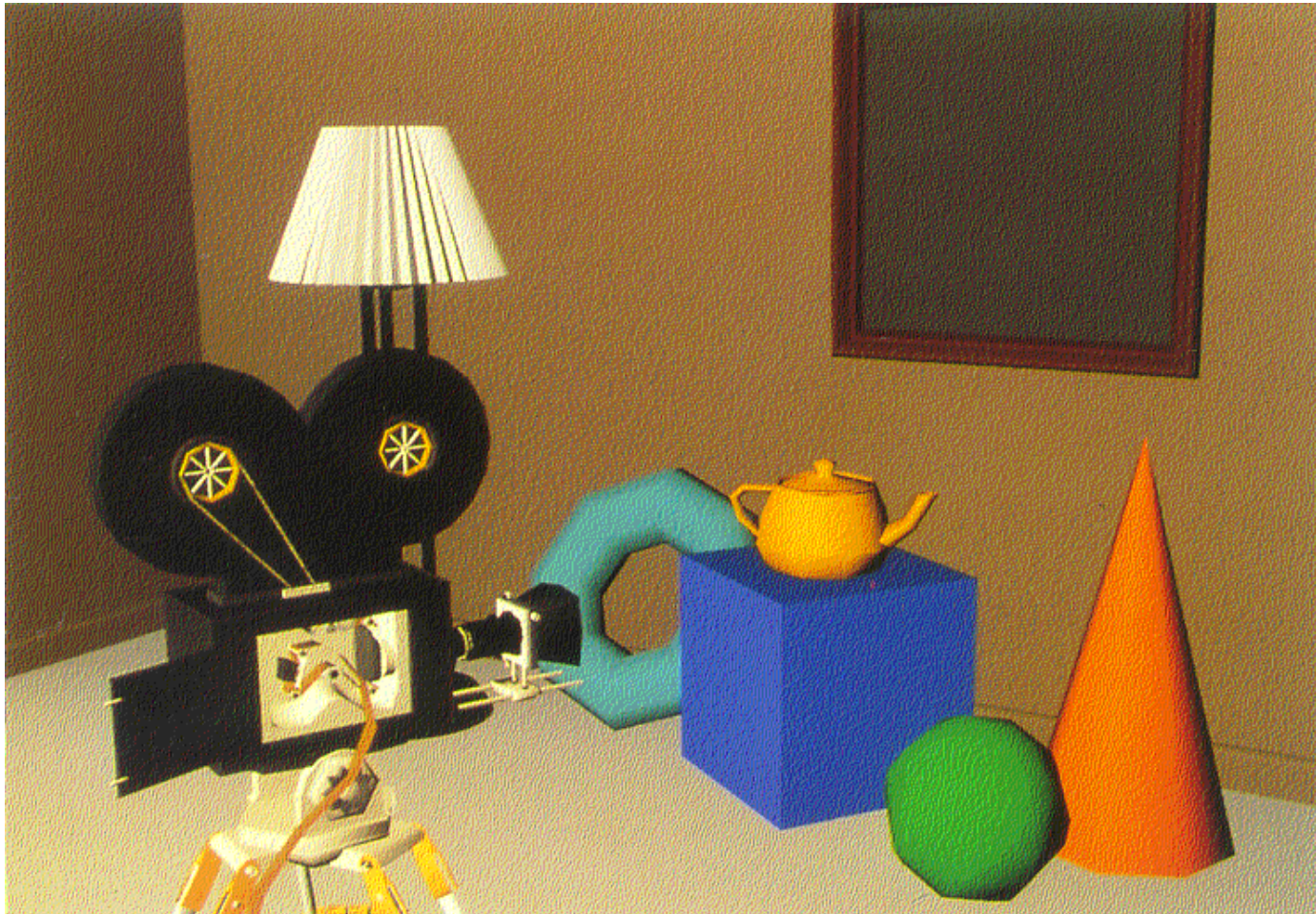


**Desired**

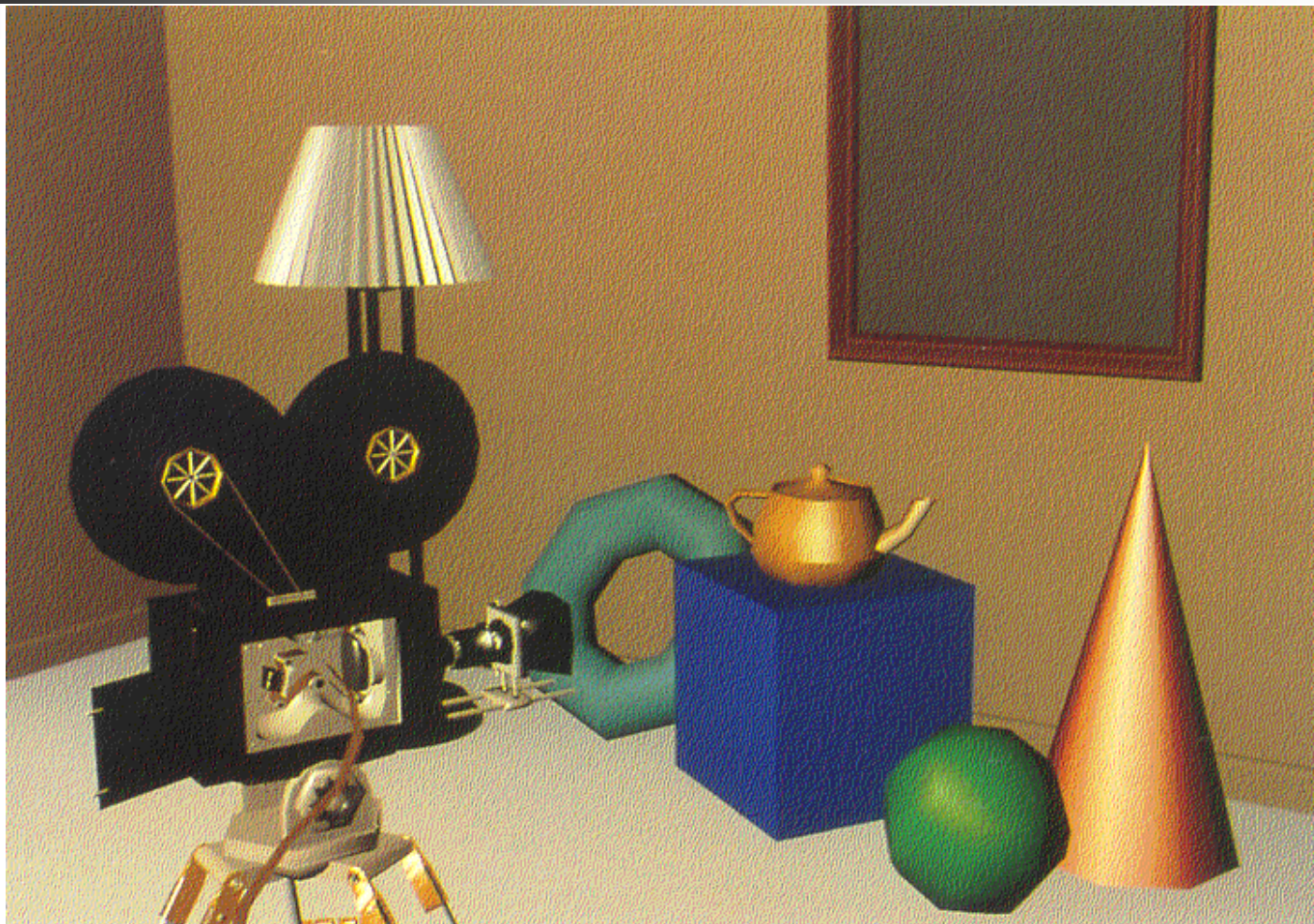


**Gouraud Result**

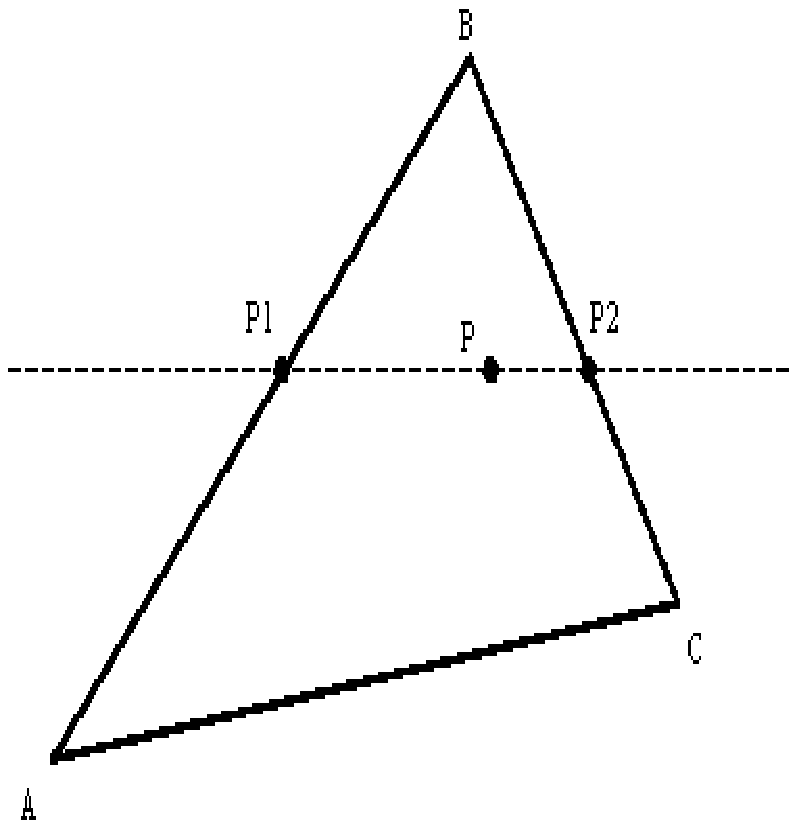
# Закраска Гуро (Gougaud) (диффузное отражение)



# Закраска Гуро (Gougaud) (зеркальное “sresular” отражение)

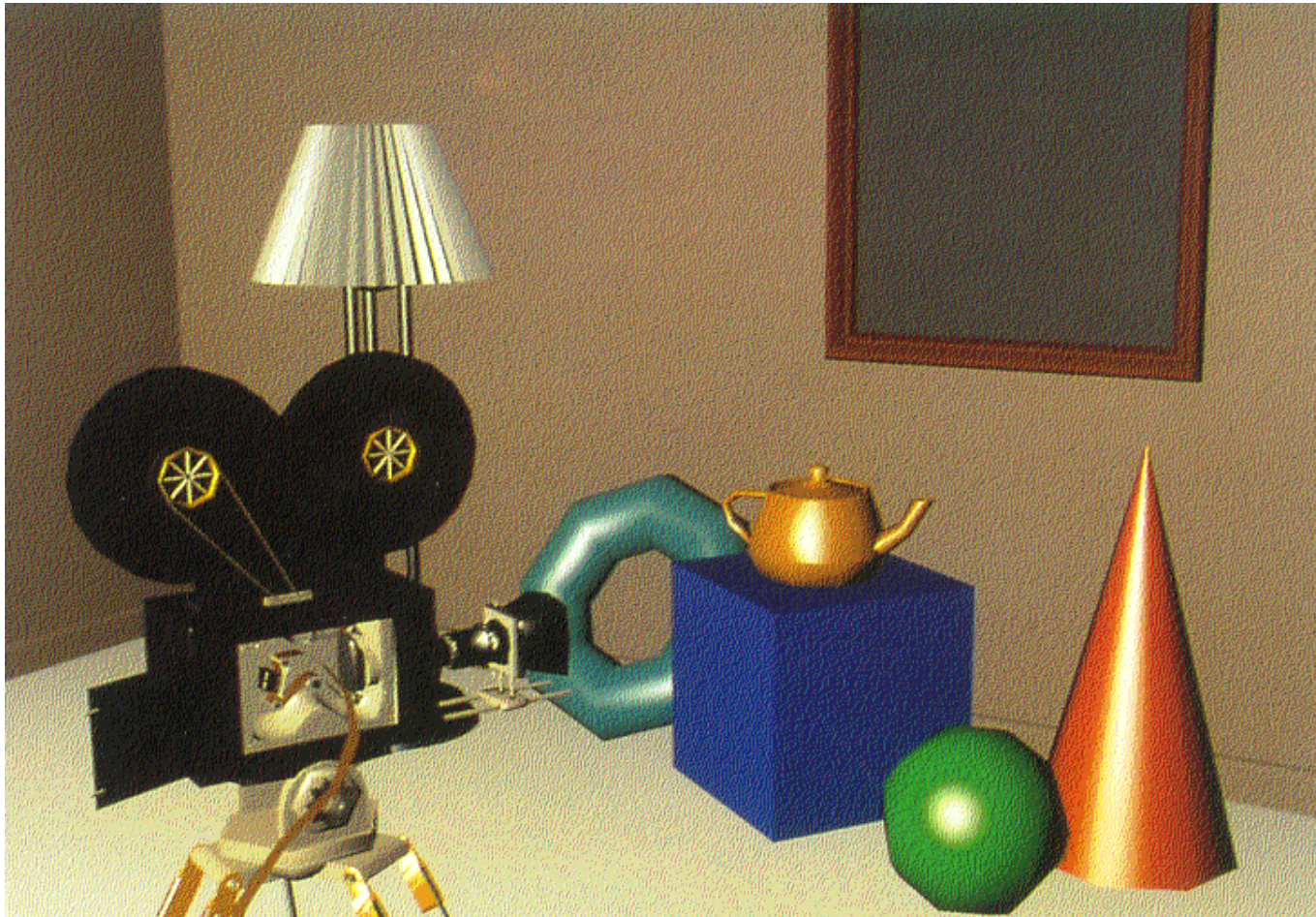


# Интерполяция нормалей



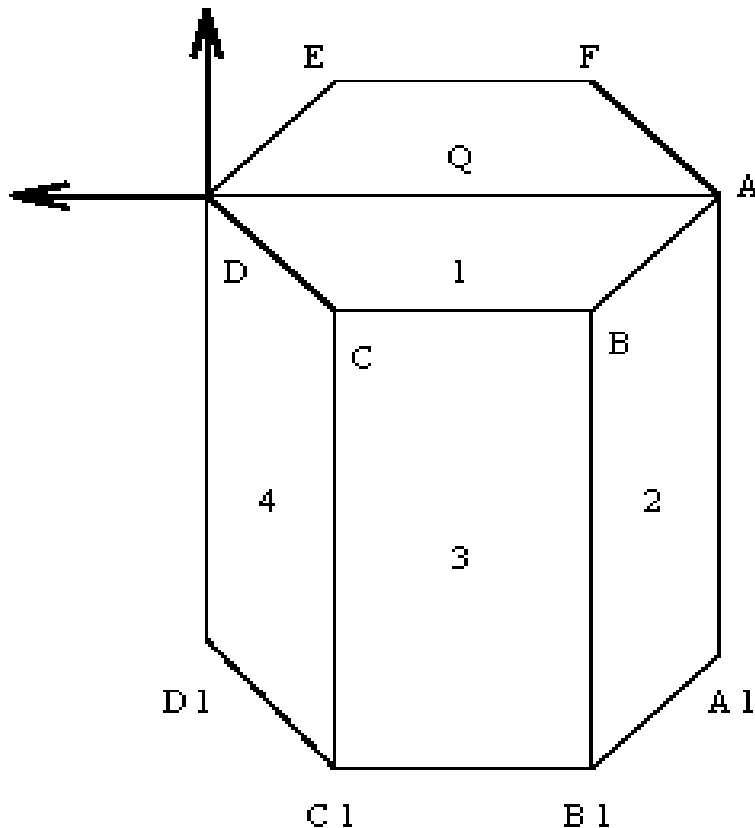
- Вычислить нормали (RGB) в каждой вершине.
- Вычислить нормаль в точках P1 и P2:  
 $s = \|P1 - B\| / \|A - B\|$   
 $N(P1) = s(N(A)) - (1-s)(N(B))$
- Вычислить нормаль в т. P:  
 $s = \|P - P2\| / \|P1 - P2\|$   
 $N(P) = s(N(P1)) - (1-s)(N(P2))$
- Вычислить цвет в точке P.

# Закраска Фонга (Phong) (зеркальное отражение)





# «Клонирование» вершин

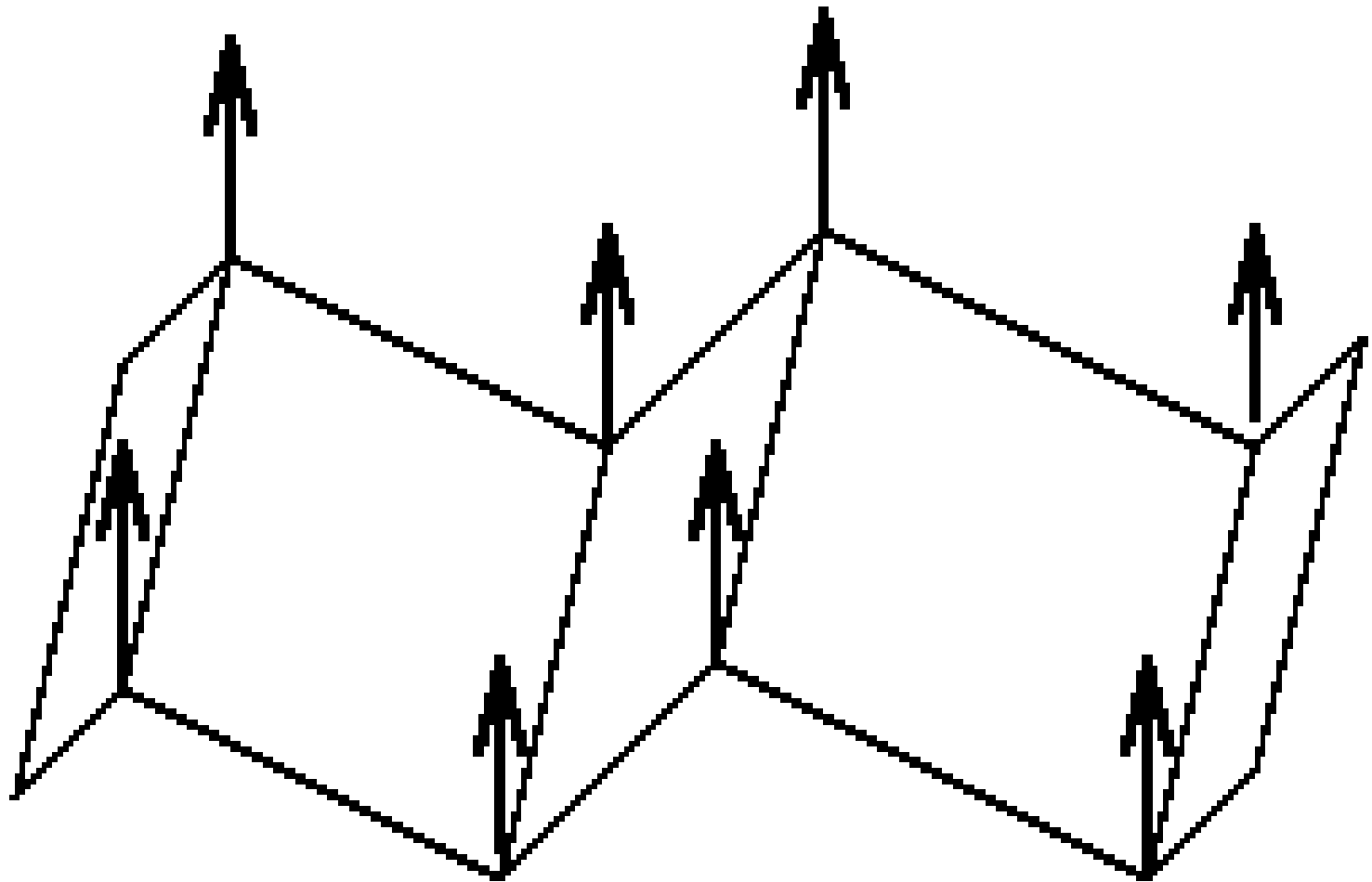


- В вершине D нужно иметь две нормали
- Одна нормаль для гладкой закрашки боковой поверхности
- Другая нормаль для закрашки торца
- На острых ребрах нормали дублируются



# Волнообразная поверхность (1)

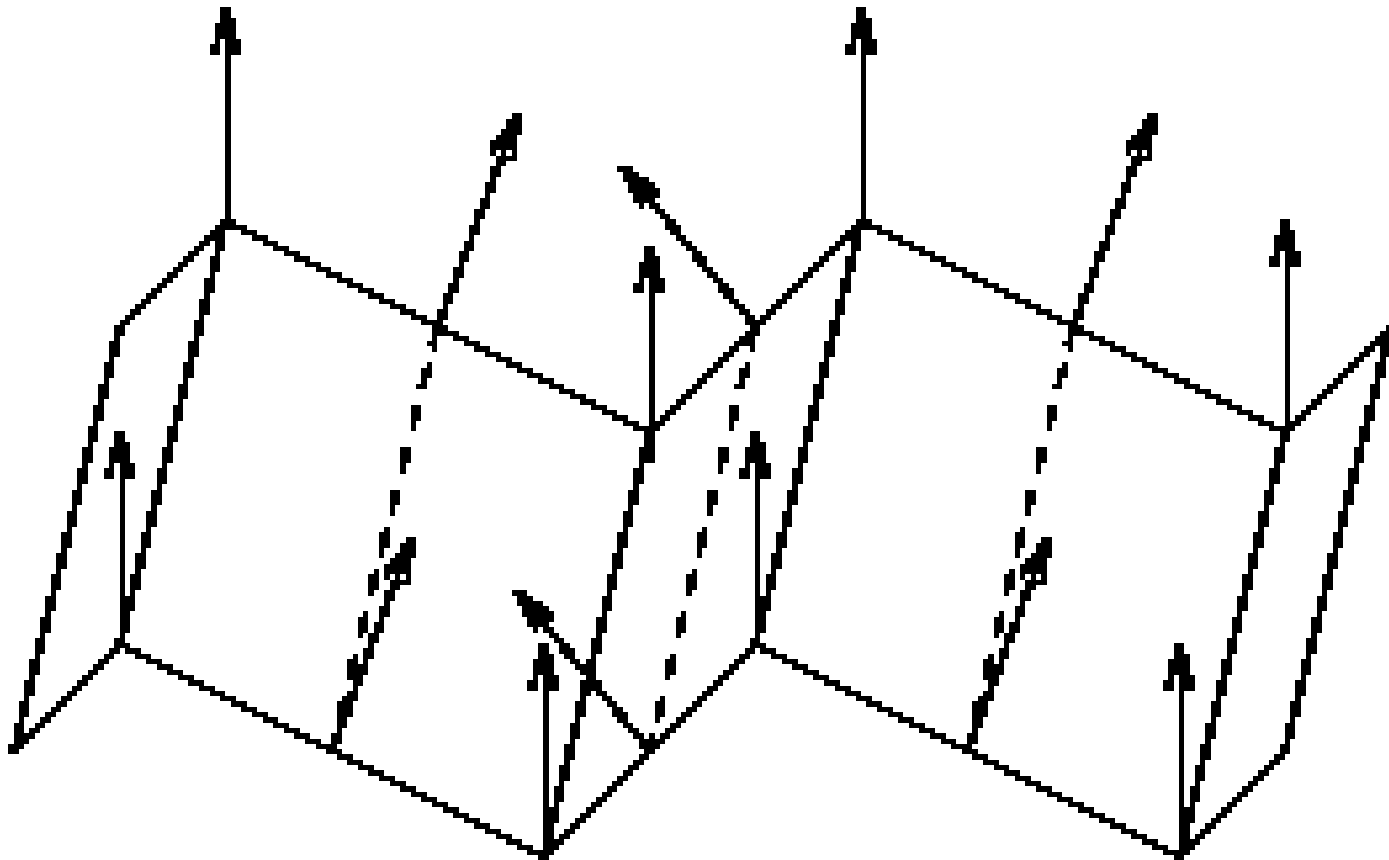
---





# Волнообразная поверхность (2)

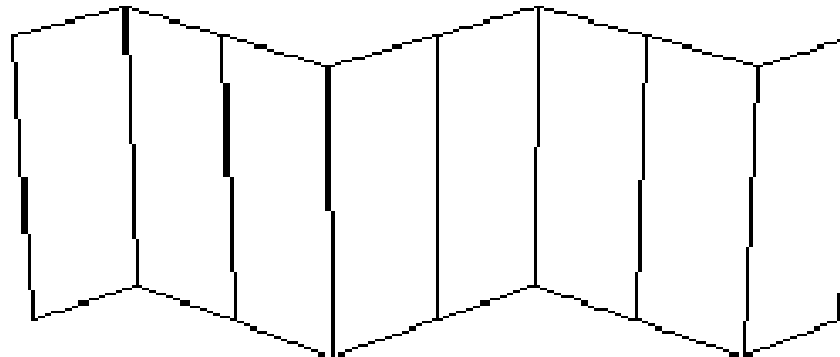
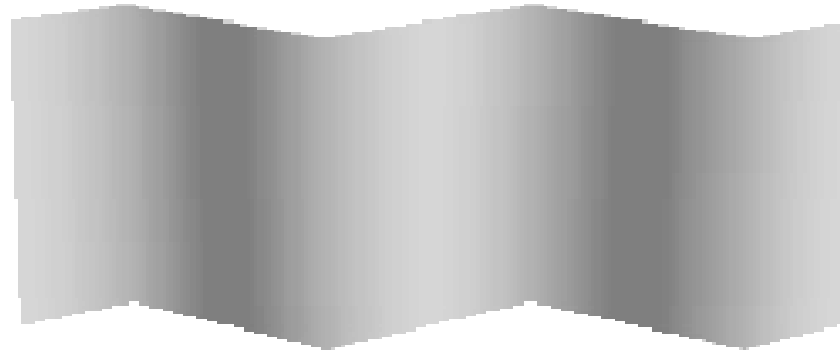
---





# Волнообразная поверхность (3)

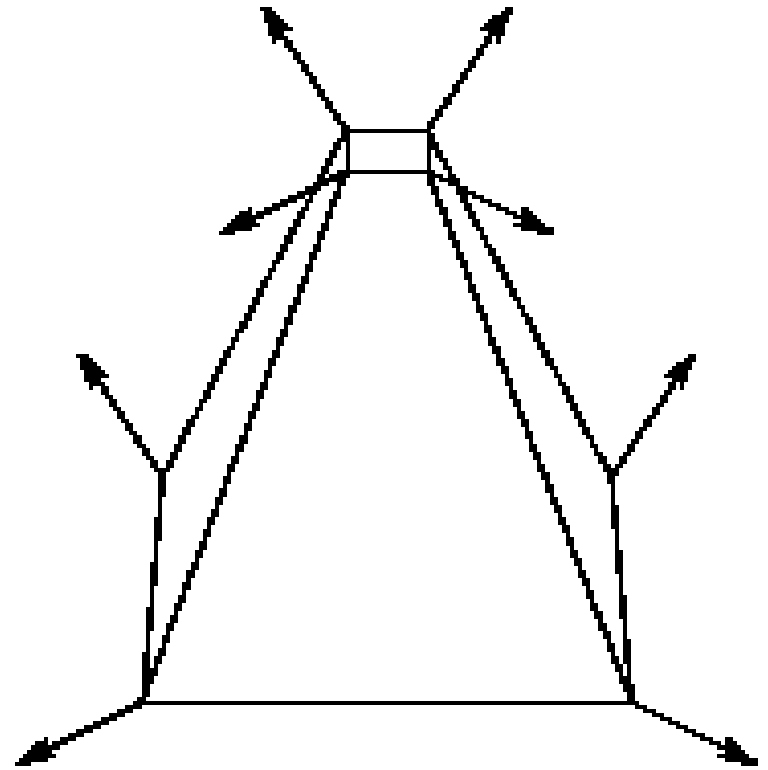
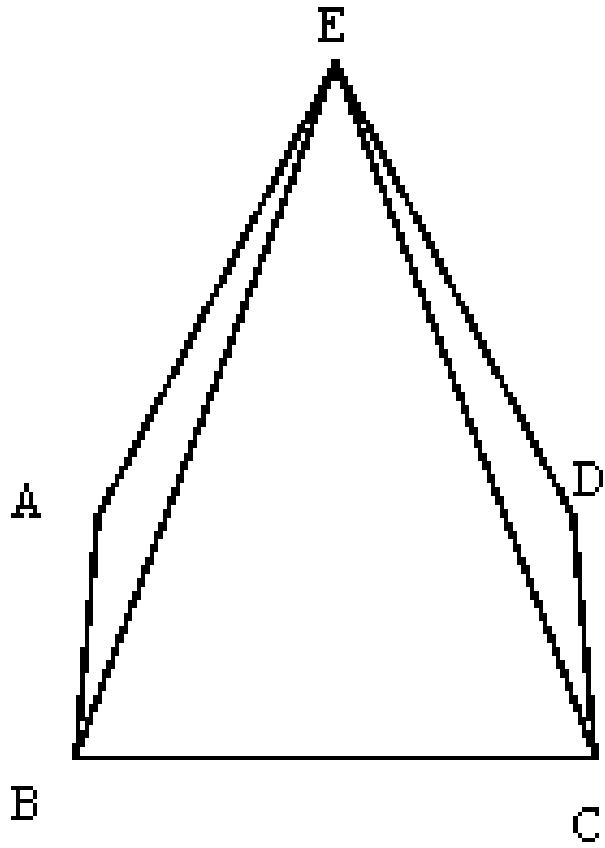
---





# Конус

---





# Поддержка в OpenGL

---

OpenGL поддерживает

- модель освещения Фонга (локальная!)
- расчет освещения на вершинах
- интерполяционная закраска Гуро

Необходимо задать:

- параметры источников света
- дополнительные атрибуты вершин
  - нормаль
  - материал

# Освещение в OpenGL

- Задаем параметры материала:

```
void glMaterialfv(GLenum face, GLenum param, GLfloat *value);  
face = {GL_FRONT|GL_BACK}  
param = {GL_AMBIENT|GL_DIFFUSE|GL_EMISSIVE|GL_SPECULAR}  
value = float[4] // RGBA
```

```
void glMaterialf(GLenum face, GL_SHININESS, GLfloat value);
```

- Задаем цвет фонового освещения:

```
void glLightModelfv(GLenum param, GLfloat *value);  
param = LIGHT_MODEL_AMBIENT  
value = float[4] // RGBA
```

- Задаем цвет источника освещения:

```
void glLightfv(GLenum light, GLenum param, GLfloat *value);  
face = {GL_LIGHT0|GL_LIGHT1|...}  
param = {GL_AMBIENT|GL_DIFFUSE|GL_SPECULAR}  
value = float[4] // RGBA
```

# Освещение в OpenGL (2)

- Задаем положение источника освещения:

```
void glLightfv(GLenum light, GL_POSITION, GLfloat *value);  
face = {GL_LIGHT0|GL_LIGHT1|...}  
value = float[4] // x,y,z,w
```

Координаты источника освещения преобразуются текущей матрицей модельного преобразования!

- Включаем расчет освещенности

```
void glEnable(GLenum type); type = GL_LIGHTING;
```

- Включаем требуемые источники освещения

```
void glEnable(GLenum type); type = GL_LIGHT0;
```

- Включаем требуемые источники освещения

```
void glShadeModel(GLenum type);  
type = GL_FLAT; - плоская закраска грани  
type = GL_SMOOTH - закраска по Гуро
```





# Литература в библиотеке

---

- *Color & Shading* (из книги Linda Shapiro & George Stockman "Computer Vision") (pdf 1.23 MB) (на англ.)
- <http://graphics.cs.msu.su/courses/cg01b/ch6.pdf>
- *D.Hearn, M.P.Baker "Computer Graphics. C Version*  
Д.Херн, М.П.Бейкер «Компьютерная графика и стандарт OpenGL», Изд. Дом «Вильямс», 2005
- Цифровая библиотека