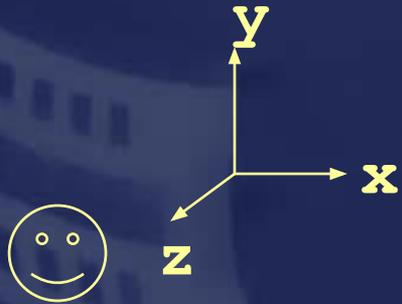


Рисуем тор

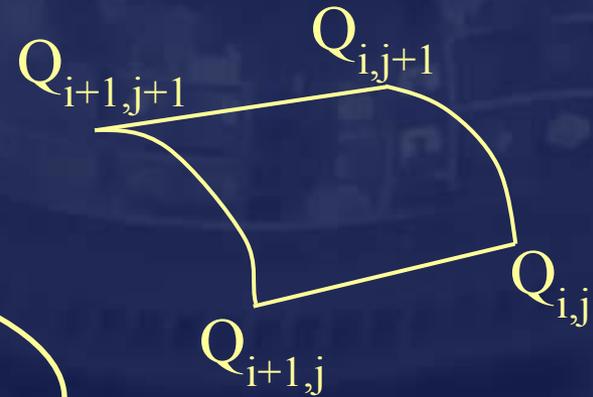
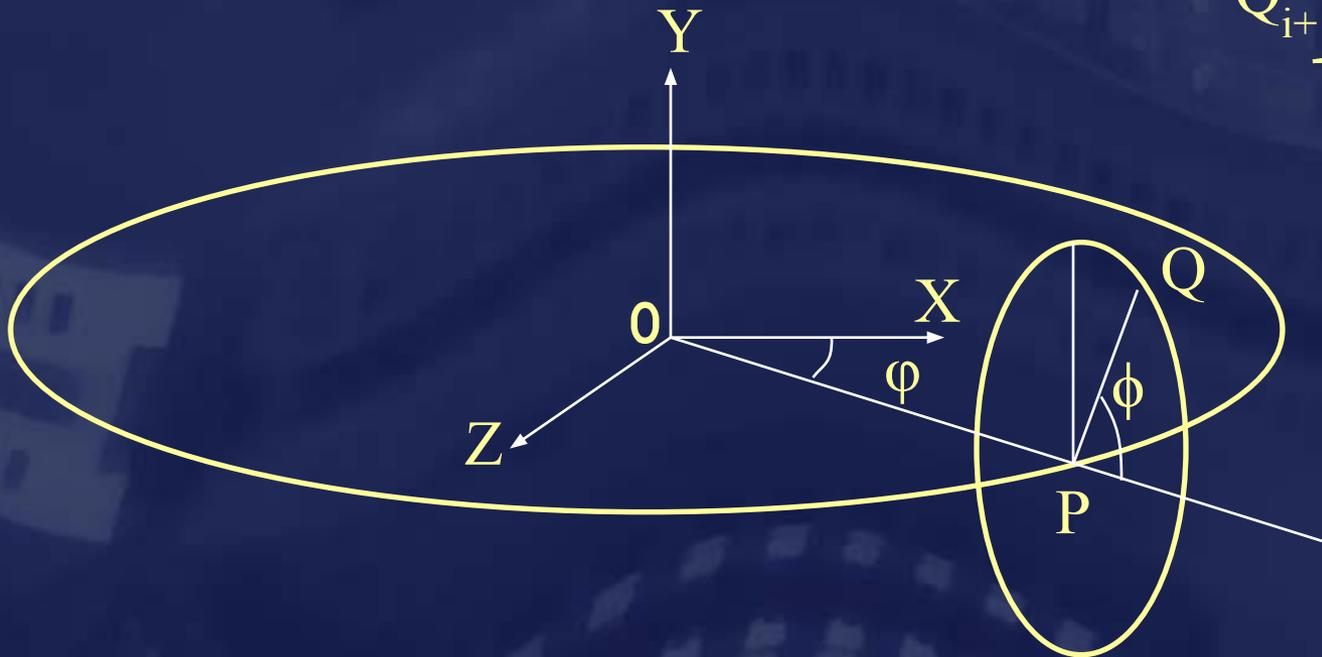


$$P(\phi) = (R_1 \cos \phi, 0, R_1 \sin \phi)$$

$$\bar{n}(\phi, \varphi) = (\cos \phi \cos \varphi, \sin \varphi, \sin \phi \cos \varphi)$$

$$Q(\phi, \varphi) = P(\phi) + R_2 \bar{n}(\phi, \varphi)$$

$$\phi, \varphi \in [0, 2\pi]$$

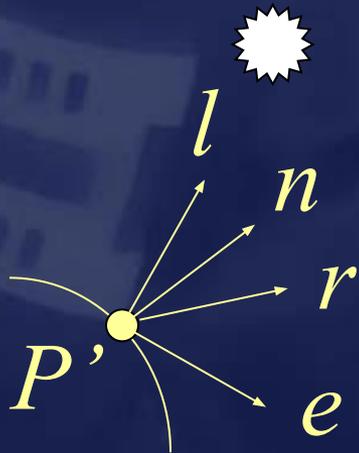


$$Q_{i,j} = Q(2\pi i / N_1, 2\pi j / N_2), i = 0, 1, \dots, N_1, j = 0, 1, \dots, N_2.$$

Уравнение освещенности по Фонгу

$$I = a_m a_l + d_m d_l (n \cdot l) + s_m s_l (e \cdot r)^{h_s}$$

- ❑ Фоновое освещение не имеет источника и зависит только от сцены
- ❑ При диффузном освещении свет от источника равномерно рассеивается во всех направлениях.
- ❑ При зеркальном освещении свет от источника отражается от поверхности в одном направлении. Зеркальная освещенность дополнительно зависит от положения наблюдателя..



$$(a \cdot b) = \begin{cases} (a, b), & (a, b) \geq 0 \\ 0, & (a, b) < 0 \end{cases}$$

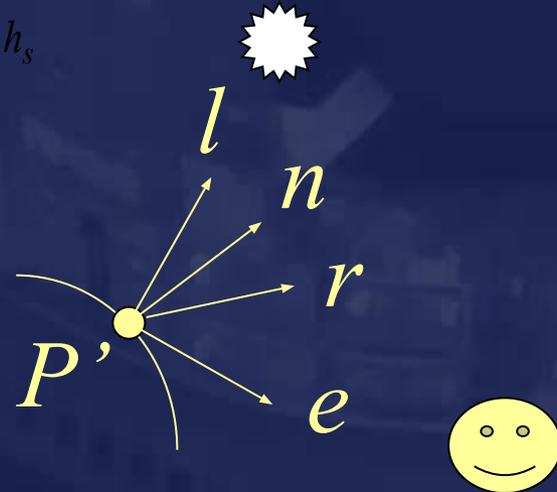
$$r = \text{reflect}(l, n)$$

Модели Блинна и Шлика

- ❑ Вычисление отраженного вектора – трудоемкая операция (Блинн)

$$I = a_m a_l + d_m d_l (n \otimes l) + s_m s_l (n \otimes h)^{h_s}$$

$$h = \frac{l + e}{\|l + e\|}$$



- ❑ Возведение в степень также работает не очень быстро... (Шлик)

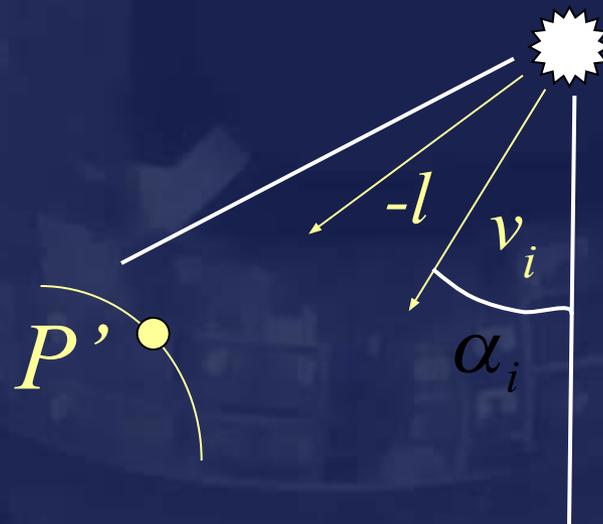
$$(n \otimes h)^{h_s} \sim \frac{D}{h_s - Dh_s + D}, D = (n \otimes h)$$

Уравнение освещенности OpenGL

$$c = e_m + a_m a_s + \sum_{i=0}^{n-1} att_i \cdot spot_i \cdot (a_m a_i + d_m d_i (n \boxtimes l) + s_m s_i (n \boxtimes h)^{h_m})$$

$$att_i = \frac{1}{k_{c,i} + k_{l,i} r + k_{q,i} r^2},$$

$$spot_i = \begin{cases} 1, \alpha_i = \pi, \\ 0, (v_i, -l) < \cos(\alpha_i), \\ (v_i, -l), (v_i, -l) \geq \cos(\alpha_i). \end{cases}$$



$spot_i$ – коэффициент направленности

att_i – коэффициент затухания

a_s – фоновое освещение

a_i, s_i, d_i – свойства i -го источника освещения

e_m, a_m, s_m, d_m, h_m – свойства материала

Установка параметров освещения 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
```

Установка параметров освещения. Часть 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;
```

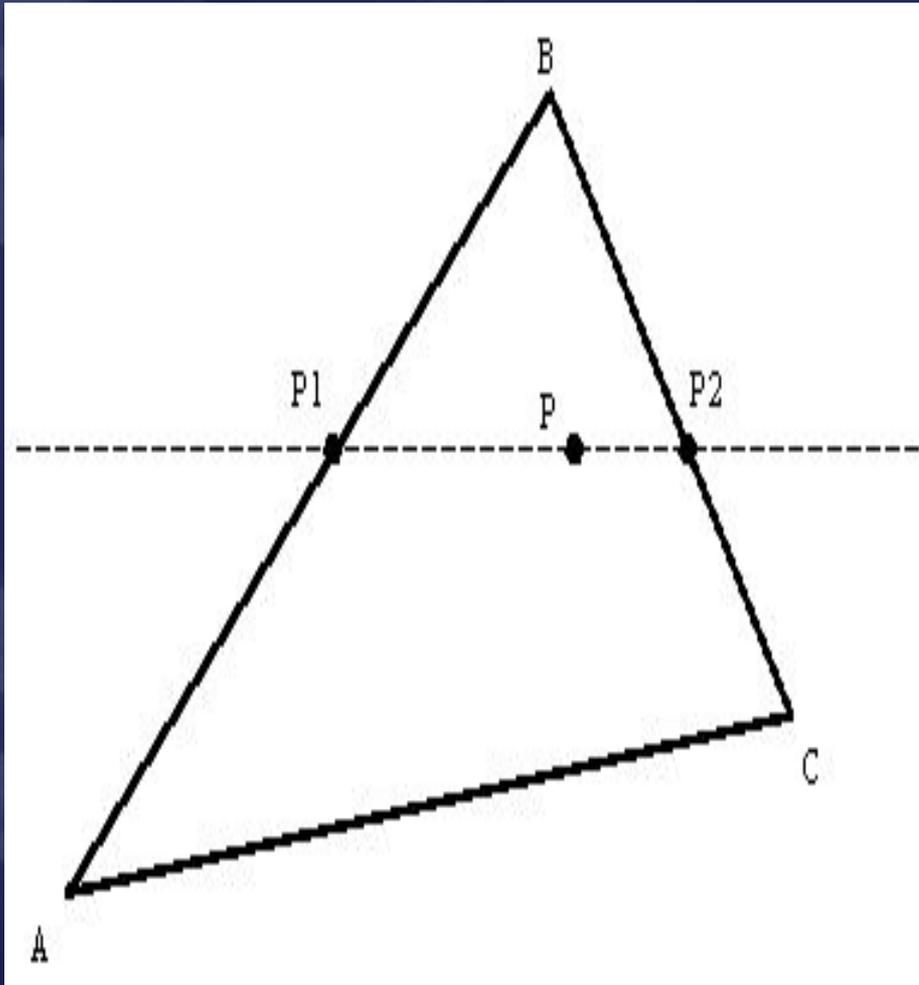
- Включаем требуемые источники освещения

```
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
```

Интерполяция цвета



- Вычислить цвет (RGB) в каждой вершине.
- Вычислить цвет в точках P1 и P2:

$$s = \|P1 - B\| / \|A - B\|$$

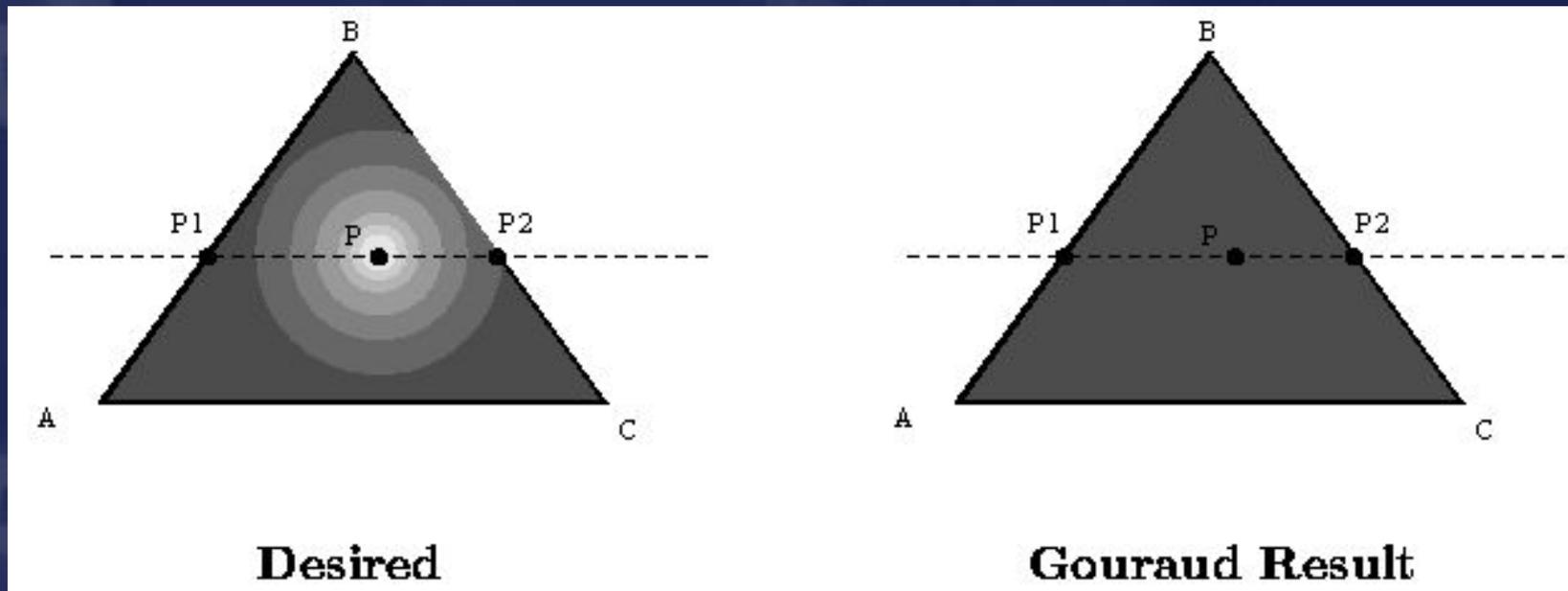
$$C(P1) = s(C(A)) - (1-s)(C(B))$$

- Вычислить цвет в т. P:

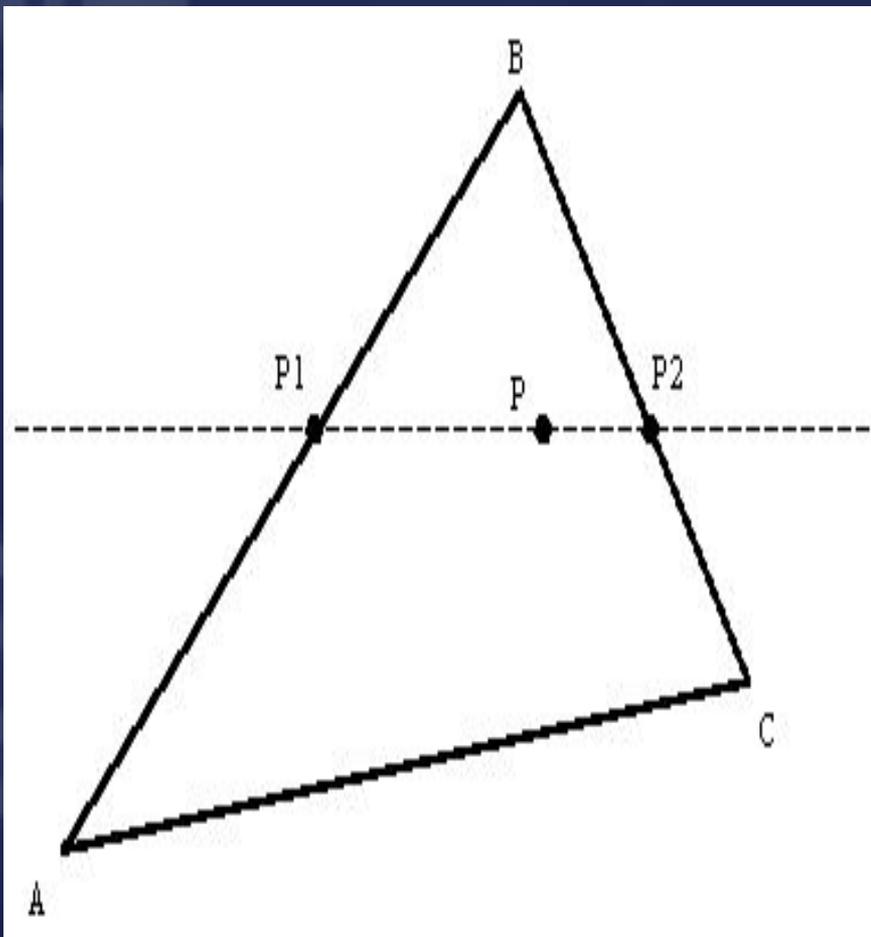
$$s = \|P - P2\| / \|P1 - P2\|$$

$$C(P) = s(C(P1)) - (1-s)(C(P2))$$

Недостатки закраски по Гуро

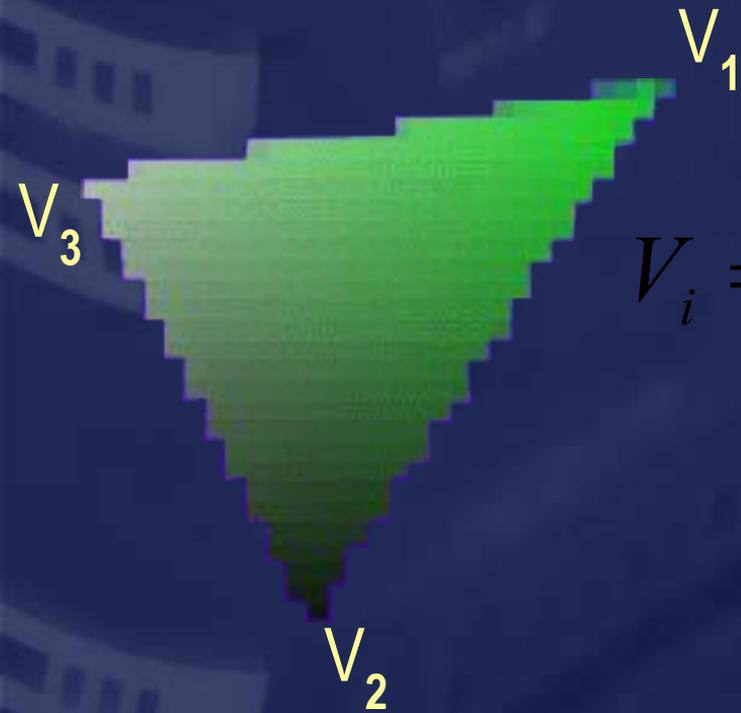


Интерполяция нормали



- 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.

Растреризация



$$V_i = \{x_i, y_i, z_i, RGBA_i, \dots\}, i = 1, 2, 3$$

- Интерполяция координаты z

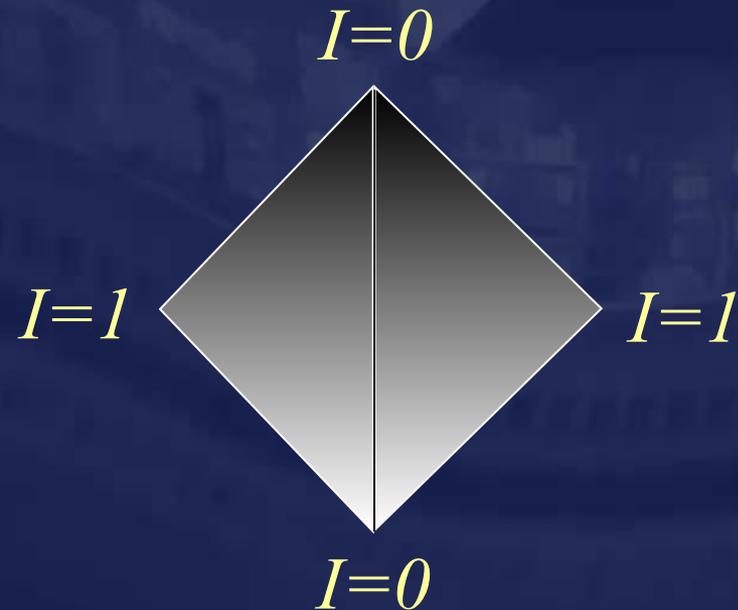
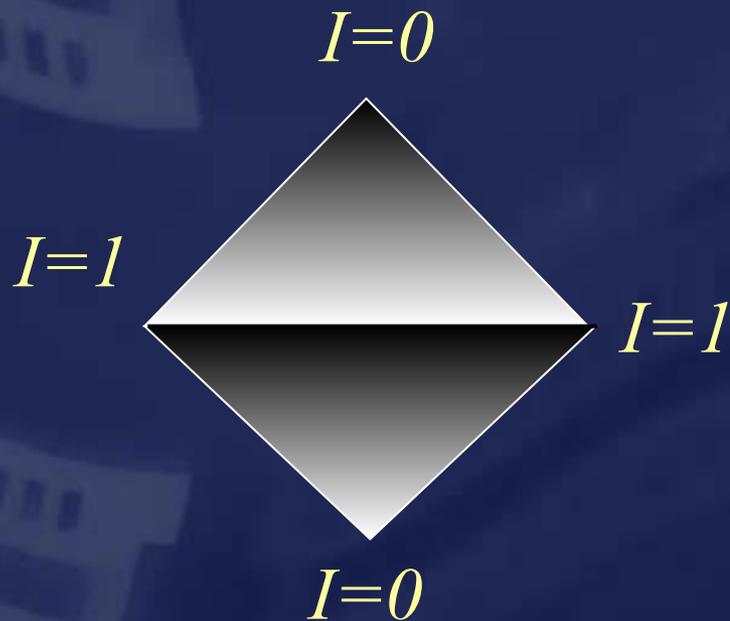
$$z(x, y) = L(x, y, z_i)$$

- Интерполяция цвета вдоль примитива - закраска по Гуро

$$RGBA(x, y) = L(x, y, RGBA_i)$$

Ошибки линейной интерполяции

- Освещенность зависит от способа разбиения на примитивы



- Поле нормалей лучше задавать в виде текстуры!