

# Интерактивная Компьютерная Графика

Часть 4-1  
(тени)

# Расчет теней

1. Вершинный (vertex shadows):
  - (+): простота, скорость
  - (–): интерполяция тени по примитиву
2. Проекционный (projected shadow):
  - (+): простота, возможность смягчения теней
  - (–): висячие тени, наложение («двойное смешение» - reblending)
3. Тектурный
  - Теневые объемы (shadow volumes, стенил-тени)
    - (+): корректность
    - (–): затратность, неинвариантность, резкость
  - Карты теней (shadow maps, shadow buffers)
    - (+): скорость
    - (–): ступенчатость
  - Карты освещения (light maps)
    - (+): скорость на этапе рендеринга
    - (–): статичность

# Расчет теней

(shadow maps, shadow buffers)

Генерируется сцена с позиции источника света

Результат записывается в текстуру (карту теней)

Генерируется сцена с позиции камеры

«Исправленные» текстуры с тенями накладываются на объект

Достоинства:

- (+) регулируемая точность
- (+) аппаратная поддержка

Недостатки:

- (-): размер текстуры ограничен
- (-): ступенчатость

# Расчет теней (shadow volumes)

«Теневой объем» (**Shadow Volume**) – это особый невидимый объект, представляющий собой объемную тень, отбрасываемую объектом.

Т.е. все, что попадает в тень от исходного объекта, находится внутри его «теневого объема»

Фактически, «теневой объем» – это сам объект, вытянутый по направлению от источника

# Расчет теней (shadow volumes)

Строим теневой объем в виде усеченной пирамиды  
(с позиции источника света)

Рисуем все объекты без теней со включенным Z-буфером

Рисуем в S-буфер «теневой объем» так,  
чтобы ненулевое значение получают затеняемые пиксели

Рисуем сцену со включенным S-буфером

Достоинства:

(+) корректность

Недостатки:

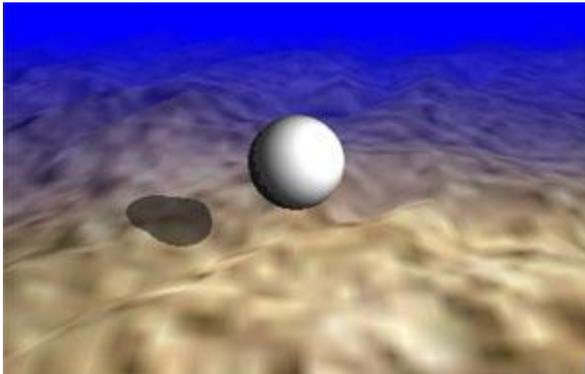
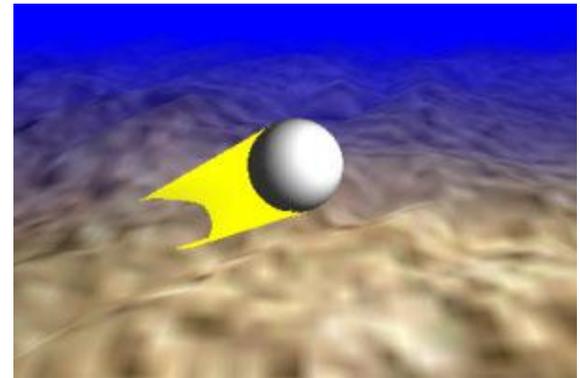
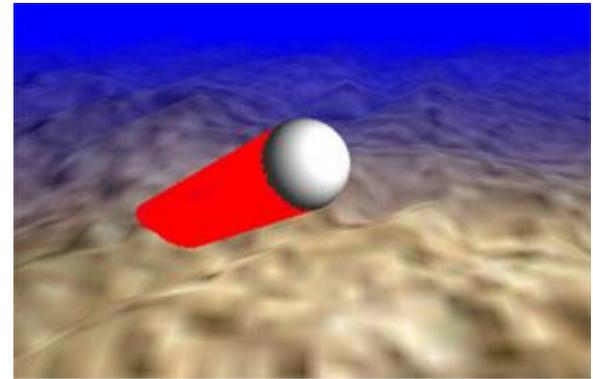
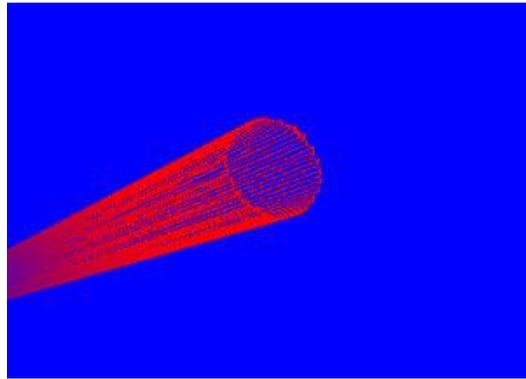
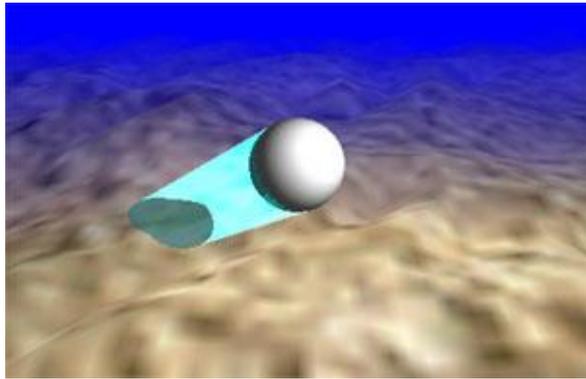
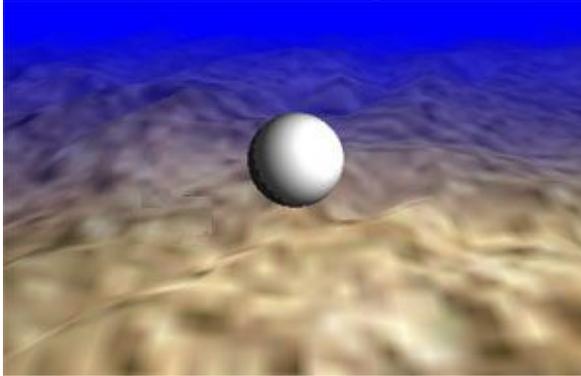
(-) затратность

(-) неинвариантность относительно положения наблюдателя

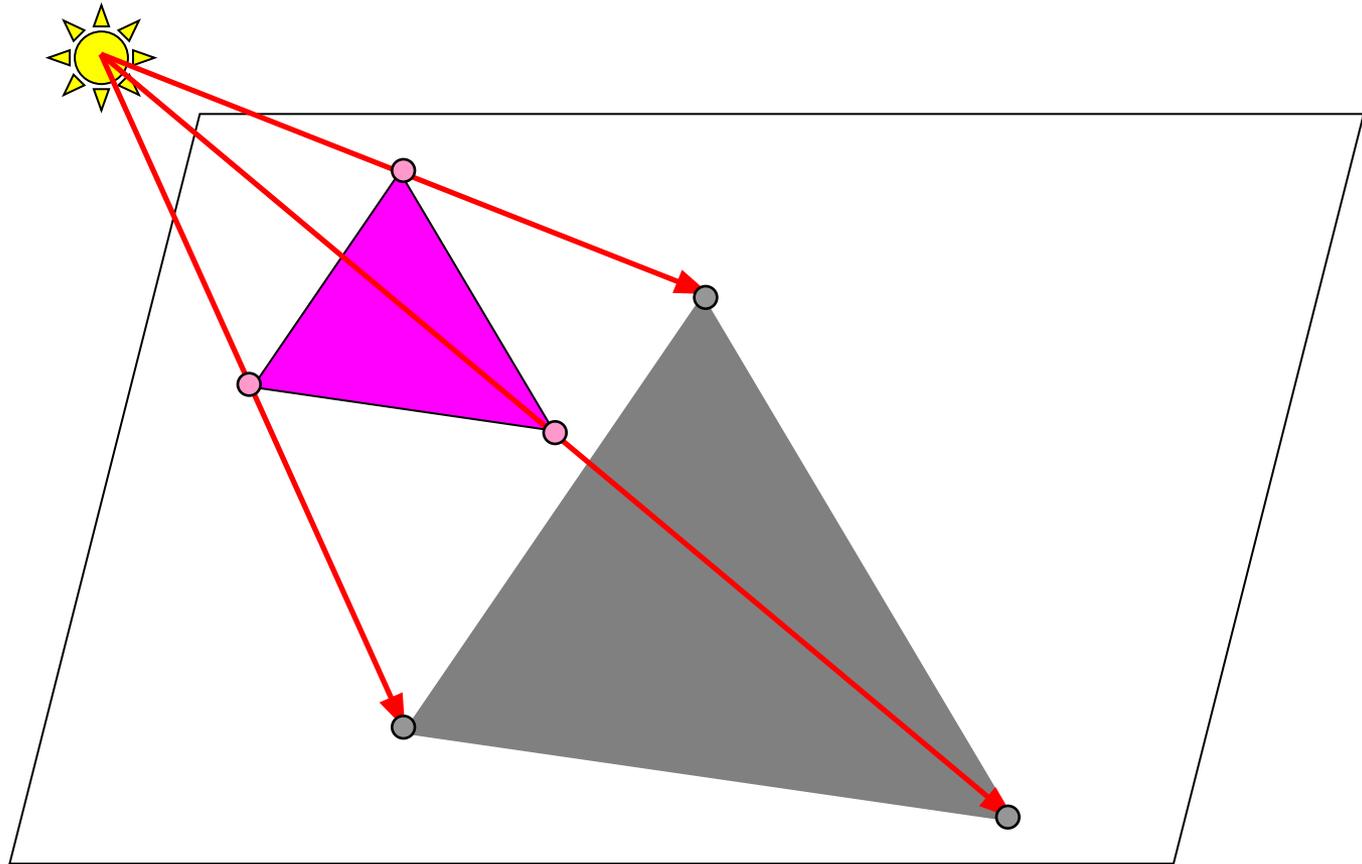
(-) резкость теней

# Расчет теней (пример)

(shadow volumes)



# Проекционный расчет теней (projected shadow)



# Проекционный расчет теней

Цель: спроецировать точку **A**  
на плоскость **XZ**  
по вектору, выпущенному из точки **L**

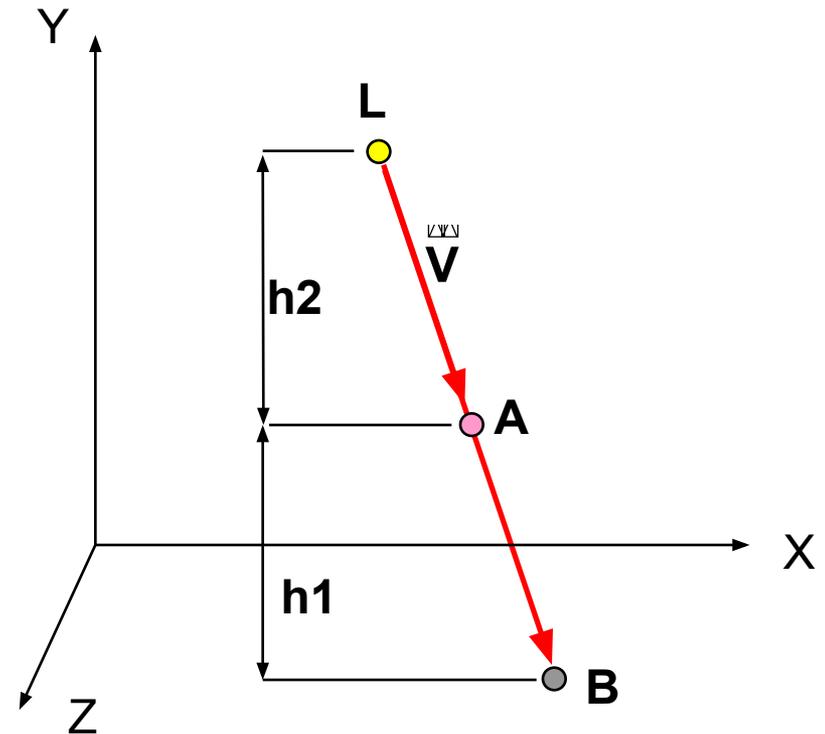
**h1** – расстояние от **A** до **XZ**

**h1+h2** – расстояние от **L** до **XZ**

Вектор от **L** до **A**:  $\vec{V} = \vec{A} - \vec{L}$

Вектор от **L** до **XZ**:  $\vec{W} = \vec{V} \cdot \frac{h1+h2}{h2}$

$\vec{B} = \vec{L} + \vec{W}$



# Проекционный расчет теней

Цель: спроецировать точку **A**  
на произвольную плоскость **P**  
(проходящую через начало координат)  
по вектору, выпущенному из точки **L**

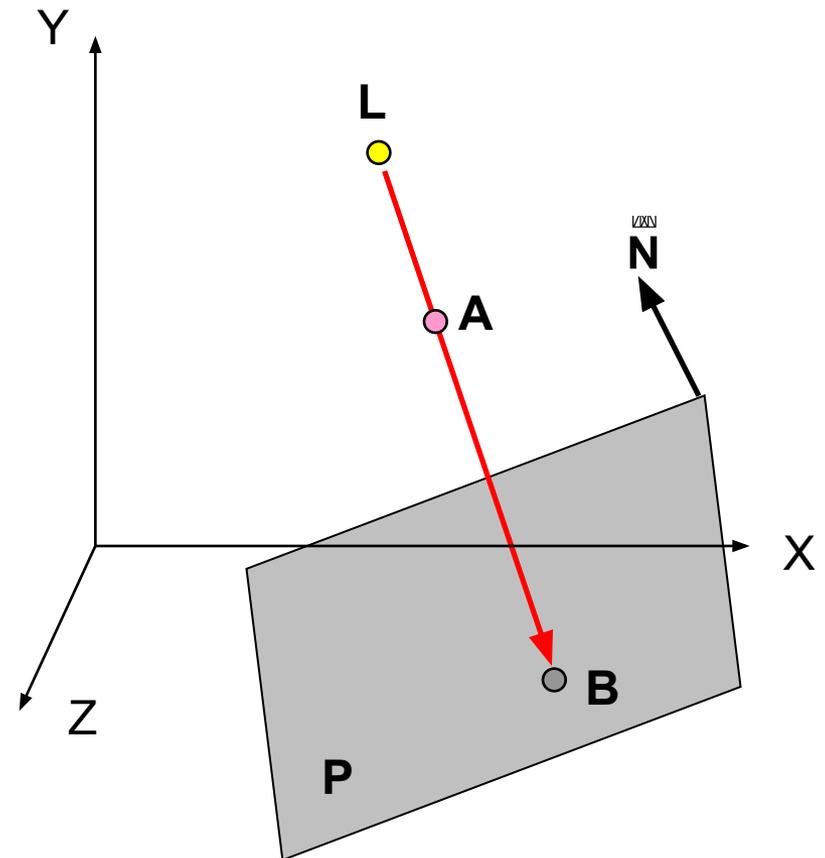
Вектор нормали к плоскости **P** :

$$\vec{N} = (N_x, N_y, N_z)$$

Вектор от **L** до **P** :

$$\vec{W} = (\vec{A} - \vec{L}) \cdot \frac{N_x \cdot A_x + N_y \cdot A_y + N_z \cdot A_z}{\sqrt{N_x^2 + N_y^2 + N_z^2}}$$

$$\vec{B} = \vec{L} + \vec{W}$$

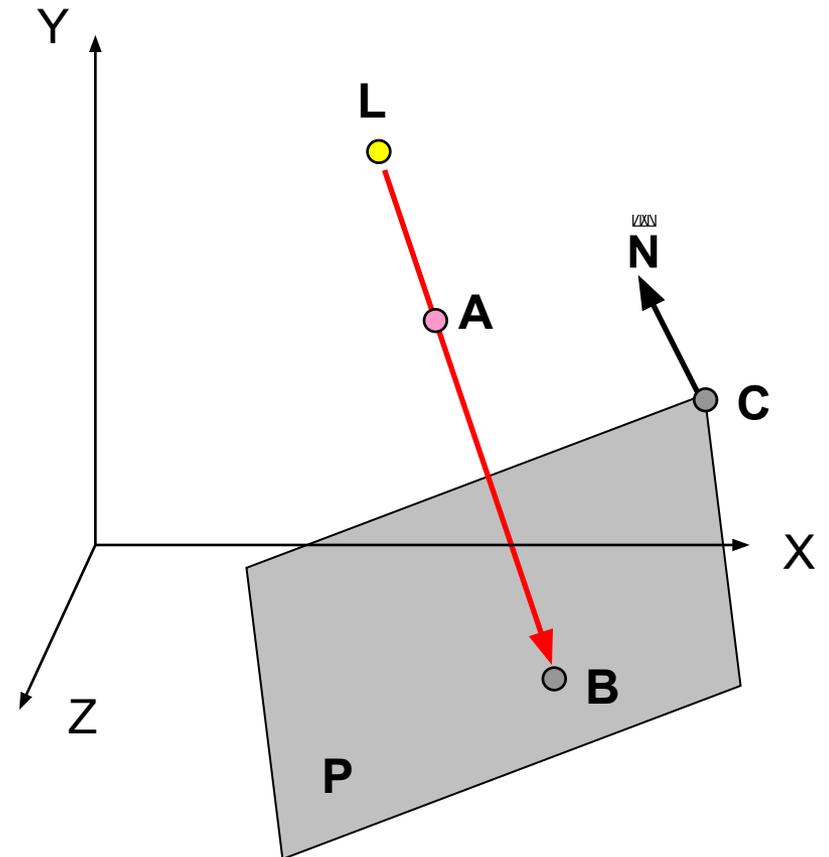


# Проекционный расчет теней

Цель: спроецировать точку **A**  
на произвольную плоскость **P**  
по вектору, выпущенному из точки **L**

$\vec{N}$  – вектор нормали к плоскости **P** в точке **C**

$$\mathbf{B} = \mathbf{L} + (\mathbf{A} - \mathbf{L}) \cdot \frac{\mathbf{N} \times (\mathbf{C} - \mathbf{L})}{\mathbf{N} \times (\mathbf{A} - \mathbf{L})}$$



# Карты освещения (light maps)

Определяется ограничивающий прямоугольник-текстура (**карта освещения**)

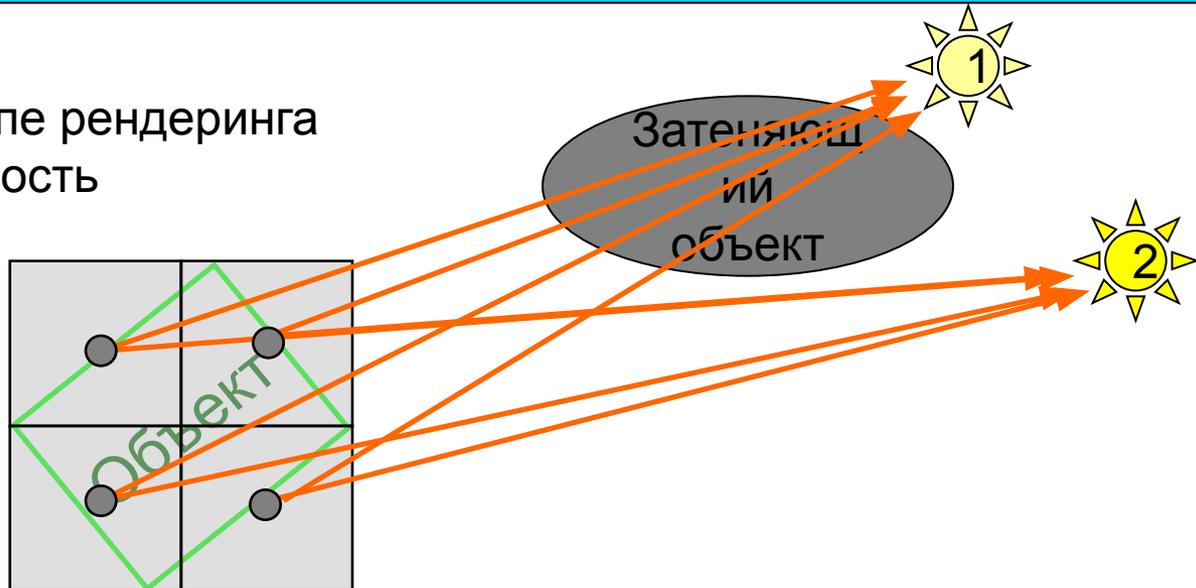
В каждом текселе (как точке объекта) определяются все видимые источники

Цвет текселя определяется по модели освещения

Полученная текстура накладывается на объект в режиме смешивания

(+): скорость на этапе рендеринга

(-): память, статичность



# Вершинные тени (vertex shadows)

В каждой вершине примитива определяются все видимые источники

Цвет в вершинах определяется по модели освещения

Примитив закрашивается по алгоритму Гуро

(+): простота, скорость, динамичность

(-): интерполяция тени

