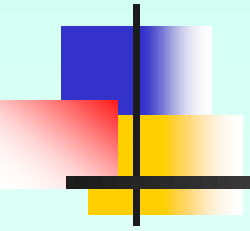
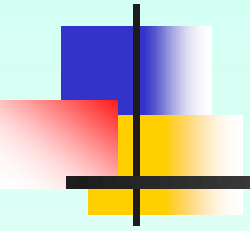


COMPUTER GRAPHICS

ЗАСОБИ ПРОГРАМУВАННЯ КОМП’ЮТЕРНОЇ ГРАФІКИ

Лек. 15 2021 ІПЗ-19



COMPUTER GRAPHICS

Ray Tracing

ТРАССИРОВКА ЛУЧЕЙ

- Фотореалистическая компьютерная графика
- Прямая трассировка лучей.
- Обратная трассировка лучей.
- Вычислительная сложность трассировки.

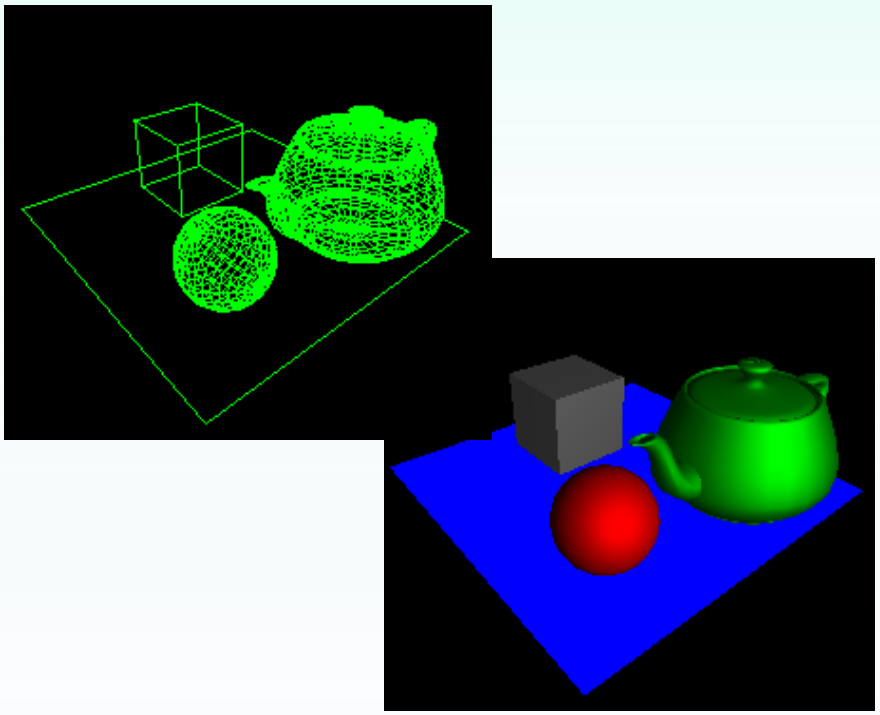
СКОРОСТЬ СИНТЕЗА VS ФОТОРЕАЛИЗМ

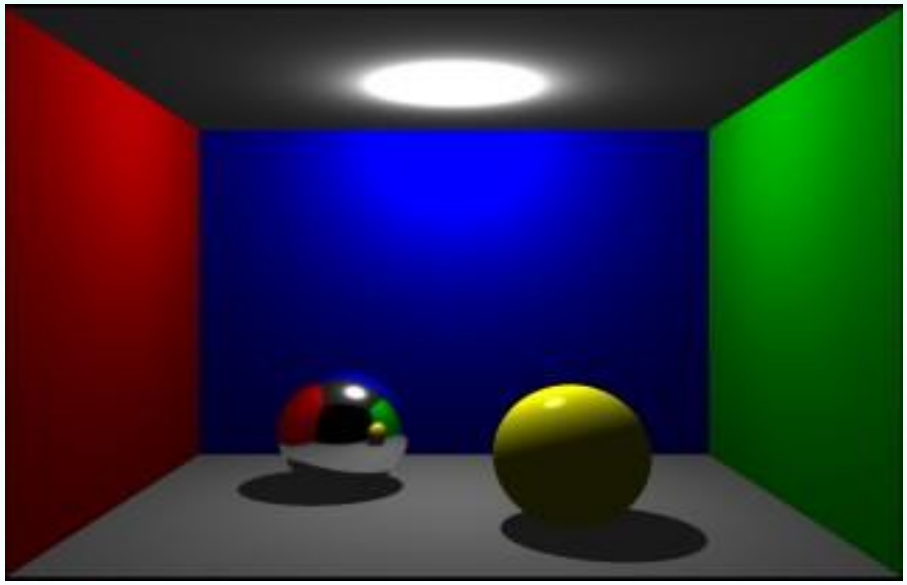
Скорость

- Растеризация
- Текстурирование

Фотореализм

- **Трассировка лучей**
- Излучательность





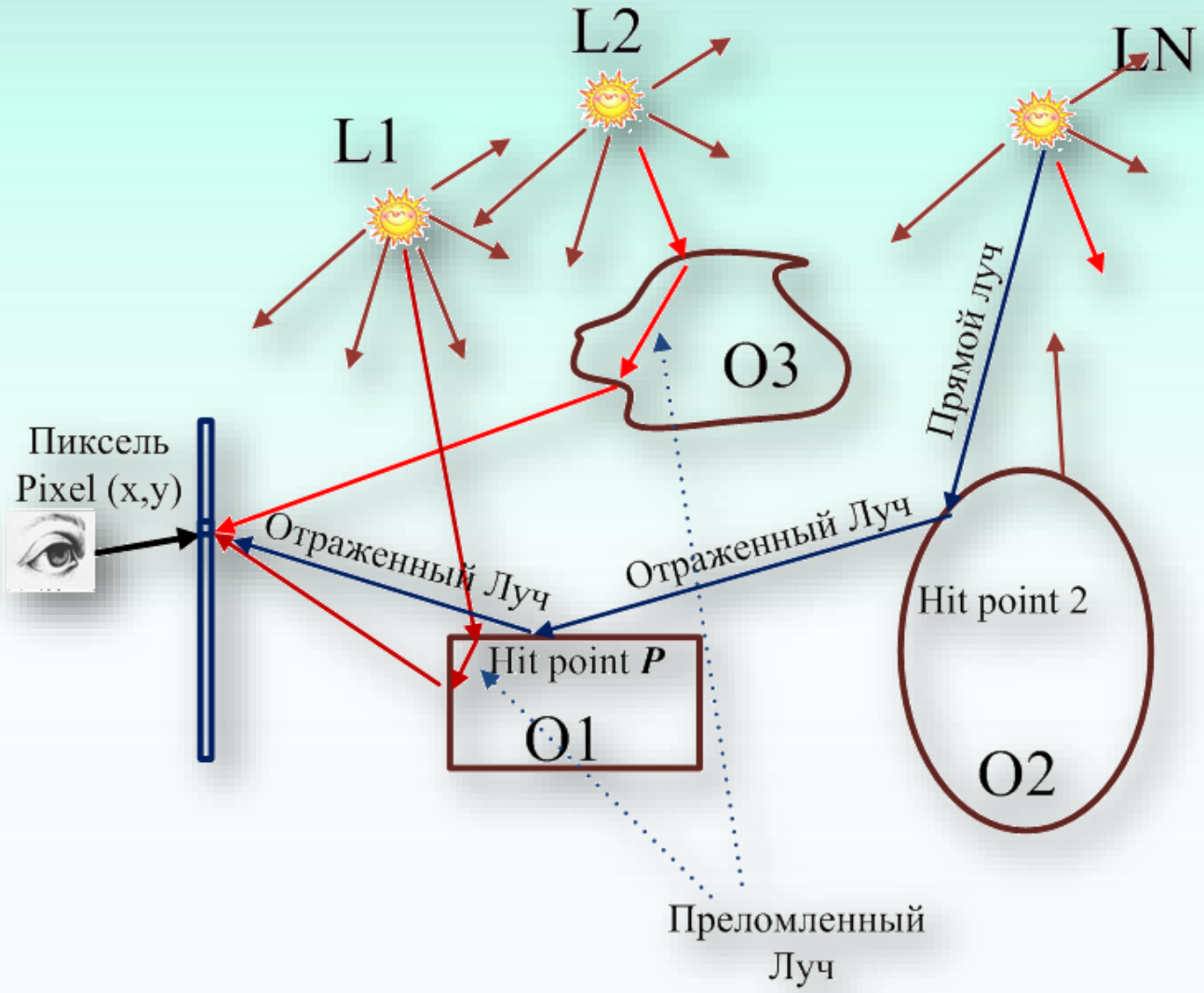
Ray Tracing. Трассировка лучей.

Трассировка лучей (Ray tracing; рейтрейсинг) — **Общее:** один из методов **геометрической оптики** — исследование оптических систем путём отслеживания взаимодействия отдельных лучей с поверхностями.

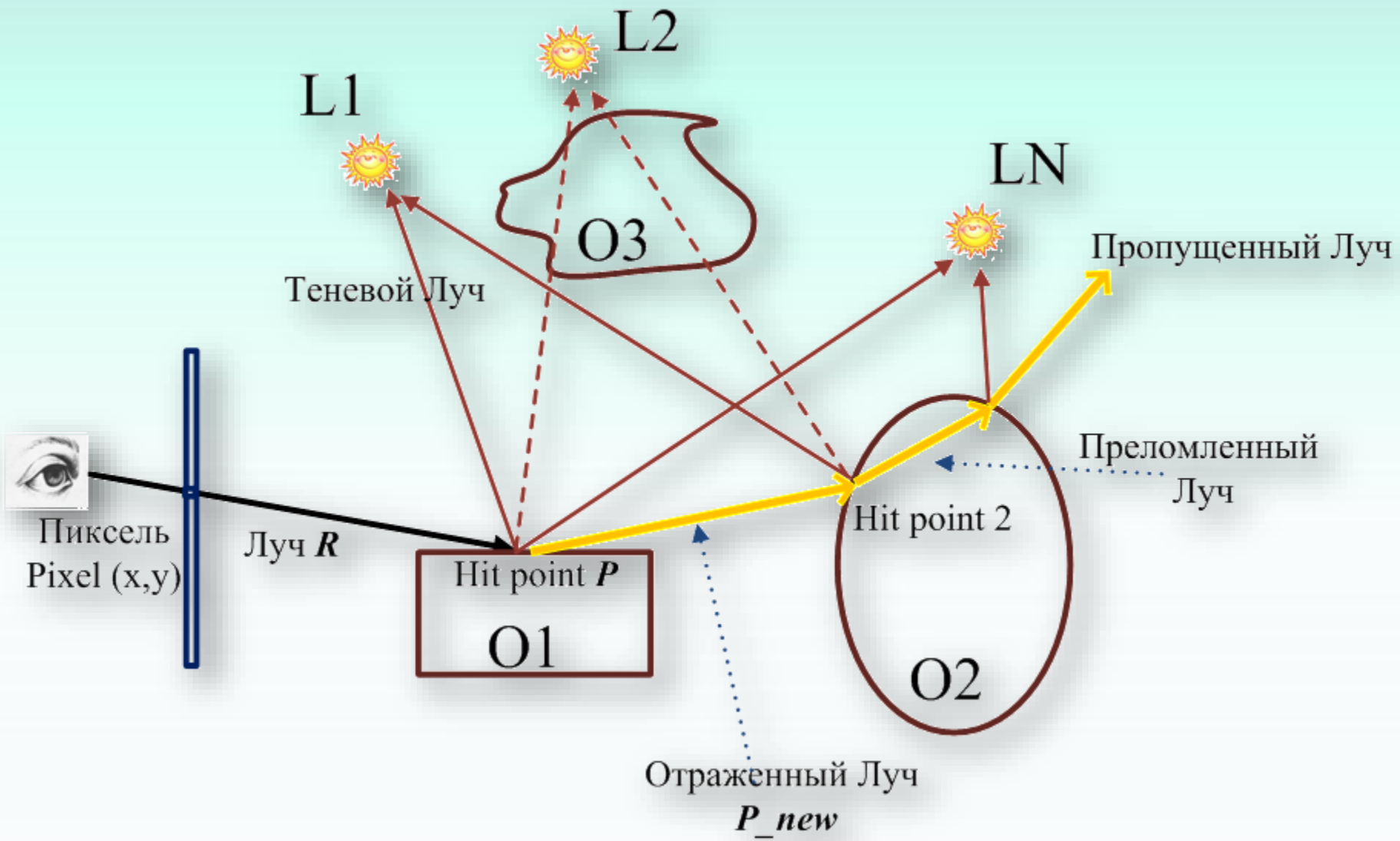
В компьютерной графике — технология построения изображения трёхмерных моделей, при которых отслеживается прямая траектория распространения луча от источника к экрану (**прямая трассировка**) или обратная траектория распространения луча от экрана к источнику (**обратная трассировка**).

Трассировка лучей в КГ — это решение для создания реалистичного освещения, отражений и теней, обеспечивающее более высокий уровень реализма по сравнению с традиционными способами рендеринга.

ПРЯМАЯ ТРАССИРОВКА ЛУЧЕЙ



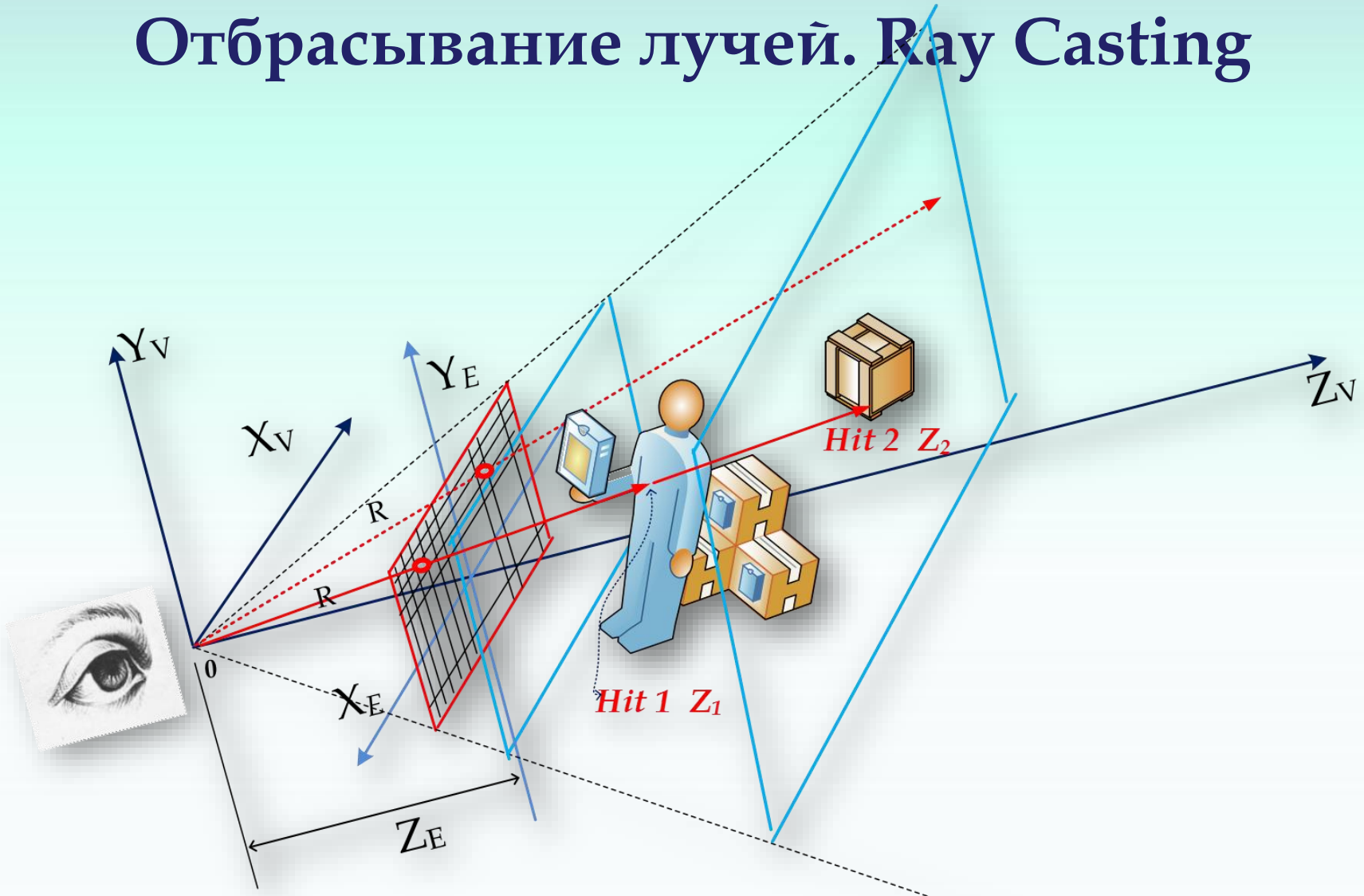
ОБРАТНАЯ ТРАССИРОВКА ЛУЧЕЙ



Turner Whitted (1979)

ОБРАТНАЯ ТРАССИРОВКА ЛУЧЕЙ

Отбрасывание лучей. Ray Casting



ОТБРАСЫВАНИЕ ЛУЧЕЙ
ray casting

От наблюдателя через **каждый** пиксель растра экрана проводится луч (***R,*** ray). Ищется пересечение луча с **каждым** объектом сцены (попавшим в пирамиду видимости). Все точки пересечения сортируются по ***Z***. Цвет пикселя устанавливается равным цвету точки пересечения с минимальным ***Z.***

Для каждого пикселя !

ОБРАТНАЯ ТРАССИРОВКА ЛУЧЕЙ

Алгоритм

ForEach pixel(x,y)
   R = ray from eyepoint *E* through pixel(x,y)
   Image [x,y] = **raytracer**(*R*)

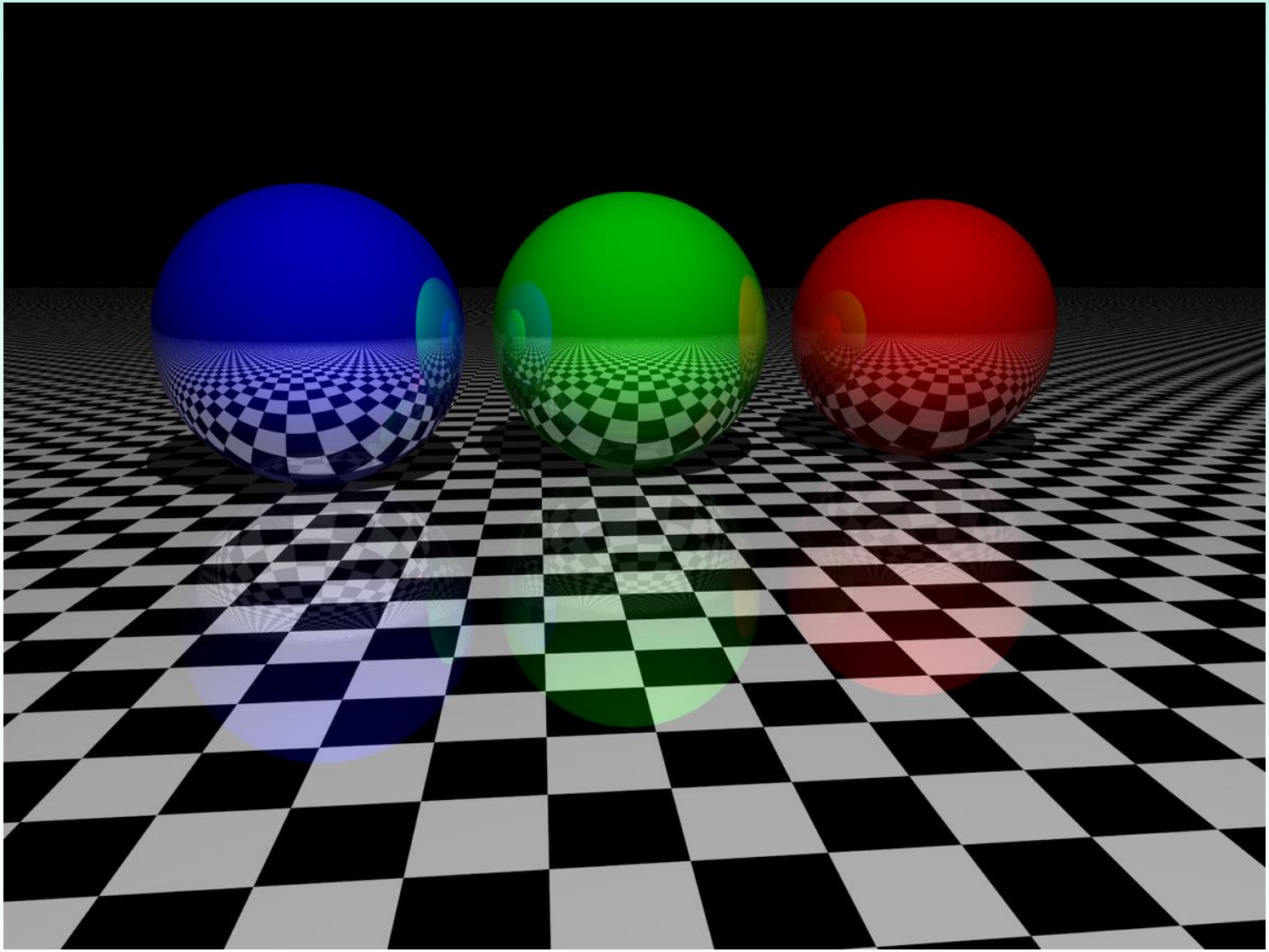
raytracer (*R*):
   P = **raycast** (*R*) // точка первого пересечения
   return **lightfrom** (*P*,*R*) //свет покидающий *P*
    //в направлении противоположенном *R*

ОБРАТНАЯ ТРАССИРОВКА ЛУЧЕЙ

Алгоритм

Lightfrom (*P*,*R*):
  color = emitted light from *P* in direction opposite *R*
  Foreach light_source *L*:
    If *L* is visible from *P*:
      Contribution = light from *S* scattered at *P* in direction opposite *R*
      color += *contribution*
    If scattering at *P* is specular: //*зеркальное*
      R_new = reflected ray at *P* //*новая_точка*
      color += **raytracer** (*R_new*)
    return *color*

ОБРАТНАЯ ТРАССИРОВКА ЛУЧЕЙ



ОБРАТНАЯ ТРАССИРОВКА ЛУЧЕЙ

ОБРАТНАЯ ТРАССИРОВКА ЛУЧЕЙ

Достоинства

- Возможность рендеринга гладких объектов без аппроксимации их полигональными поверхностями (треугольниками);
- Вычислительная сложность метода слабо зависит от сложности сцены;
- Высокая алгоритмическая распараллеливаемость вычислений — можно параллельно и независимо трассировать два и более лучей, разделять участки (зоны экрана) для трассирования на разных узлах кластера и т.д;
- Отсечение невидимых поверхностей, перспектива и корректное изменения поля зрения являются логическим следствием алгоритма.

ОБРАТНАЯ ТРАССИРОВКА ЛУЧЕЙ

Недостаток.

Высокая вычислительная сложность.

Разрешение экрана = количество исходных лучей M

HD 1920×1080 → 2 073 200 $M \approx 2*10^6$

QHD 3840 × 2160 → 8 294 400 $M \approx 8*10^6$

Количество объектов (граней) N. Например $N = 10^3$

Проверок на пересечение $N* M$

Время проверки $T \approx 10^{-3}$ (1 мкс !!!!)

Время расчета 1 кадра не менее $M* 10^{-3}$ ***сек***

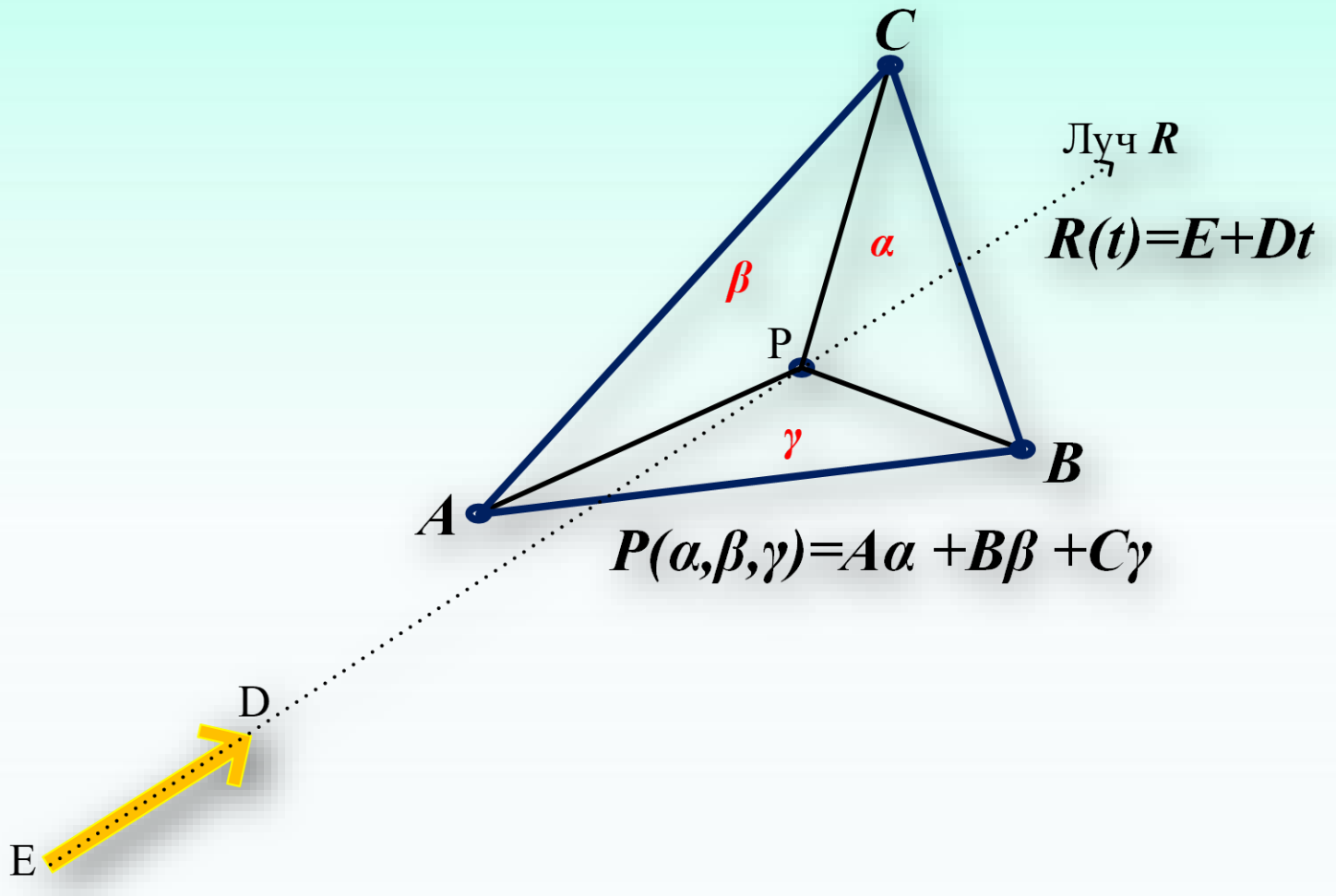
HD $\approx 2*10^3 \approx$ 32 минуты

QHD $\approx 8*10^3 \approx$ 4 часа

NVIDIA Turing (https://www.nvidia.com/ru-ru/geforce/turing)

ПОИСК ТОЧКИ ПЕРСЕЧЕНИЯ

Барицентрический тест



ПОИСК ТОЧКИ ПЕРСЕЧЕНИЯ

Барицентрический тест

E - *исходная точка луча*

D - *направление луча*

A, B, C - *вершины треугольника*

P - *точка пересечения*

t - *параметр точки пересечения*

α, β, γ; барицентрические координаты

$\alpha = S(\Delta BCP) / S(\Delta ABC)$,

$\beta = S(\Delta APC) / S(\Delta ABC)$,

$\gamma = S(\Delta ABP) / S(\Delta ABC)$;

$\alpha+\beta+\gamma=1$ -> *Условие пересечения:*

$t > 1;\ 0 < \alpha < 1$ AND $0 < \beta < 1$ AND $0 < 1 - \alpha - \beta < 1$

ПОИСК ТОЧКИ ПЕРСЕЧЕНИЯ

Барицентрический тест

Определение *P* - точки пересечения.
Уравнение плоскости, содержащей *A, B C*

$$N \cdot P + d = 0$$

где:

N* – направляющий вектор плоскости, содержащей *A, B, C;

***d* = -*N*·*A*, свободный член уравнения плоскости.**

Т.е.

$$N \cdot P(t^*) + d = 0 \text{ или } N \cdot (E + Dt^*) + d = 0$$

И

$$t^* = -(d + N \cdot E) / N \cdot R;\ t > 0\ !!!\ \rightarrow P(t^*) = E + Dt^*$$

ПОИСК ТОЧКИ ПЕРСЕЧЕНИЯ

Барицентрический тест

$$S(\Delta ABC) = |N| / 2$$

$$\alpha = S(\Delta BCP) / S(\Delta ABC),\ S(\Delta BCP) = |(B-P) \times (C-P)| / 2$$

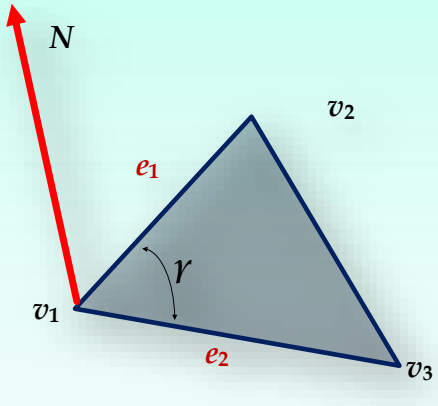
$$\beta = S(\Delta APC) / S(\Delta ABC),\ S(\Delta APC) = |(C-P) \times (A-P)| / 2$$

$$\gamma = S(\Delta ABP) / S(\Delta ABC),\ S(\Delta ABP) = |(A-P) \times (B-P)| / 2$$

Если внутри -> $\alpha+\beta+\gamma=1$

Условия попадания $\alpha > 0$ AND $\beta > 0$ AND $\alpha+\beta < 1$

ВЕКТОРНОЕ ПРОИЗВЕДЕНИЕ (НОРМАЛЬ к ГРАНИИ в 3D)



Нормаль – векторное произведение векторов. Вектора 2 ребра треугольника

$e_1=(v_2-v_1)$, $e_2=(v_3-v_1)$

$N= e_1 X e_2 = |e_1| * |e_2| * sin(\gamma)$

$$\boldsymbol{N} = \begin{bmatrix} i & j & k \\ V1_x & V1_y & V1_z \\ V2_x & V2_y & V2_z \end{bmatrix}$$

$$N_x = V1_y * V2_z - V2_y * V1_z$$
$$N_y = V2_x * V1_z - V1_x * V2_z$$
$$N_z = V1_x * V2_y - V2_x * V1_y$$

ПОИСК ТОЧКИ ПЕРСЕЧЕНИЯ

Барицентрический тест. Пример

E - наблюдатель [0, 0, 0]
Pix - пиксель [-0.5, 0.9, 1]
A вершина = [3, 4, 5]
B вершина = [-3, 4, 5]
C вершина = [-1, 1, 5]

D направление луча = Pix – E = [-0.5, 0.9, 1]
Вектор e_1 = B-A = [-3, 4, 5]-[3, 4, 5] = [-6, 0, 0]
Вектор e_2 = C-A = [-1, 1, 5]-[3, 4, 5] = [-4, -3, 0]

ПОИСК ТОЧКИ ПЕРСЕЧЕНИЯ

Барицентрический тест. Пример

$$S(\Delta ABC) = |N| / 2$$

$$N_x = e1_y * e2_z - e2_y * e1_z = 0$$
$$N_y = e2_x * e1_z - e1_x * e2_z = 0$$
$$N_z = e1_x * e2_y - e2_x * e1_y = (-6 * -3) - (-4 * 0) = 18$$

$$S(\Delta ABC) = |N| / 2 = 18 / 2 = 9$$

P точки пересечения ? Уравнение плоскости $N \cdot P + d = 0$

$P = (E + Dt^*)$. $N \cdot (E + Dt^*) - NA = 0$. $N \cdot Dt^* - NA = 0$

$$N_x(-0.5t^*) + N_y(-0.9t^*) + N_z(1t^*) - (N_x * A_x + N_y * A_y + N_z * A_z) = 0$$

$$N_z(1t^*) - N_z * 5 = 0$$

$t^* = 5$, $P = [-0.5*5, 0.9*5, 1*5] = [-2.5, 4.5, 5]$

ПОИСК ТОЧКИ ПЕРСЕЧЕНИЯ

Барицентрический тест. Пример

$a = S(\Delta BCP) / S(\Delta ABC)$, $S(\Delta BCP) = |(B-P) \times (C-P)|/2$

Вектор e_1 = B-P = [-3, 4, 5]-[-2.5, 4.5, 5]=[-0.5, -0.5, 0]

Вектор e_2 = C-P = [-1, 1, 5]-[-2.5, 4.5, 5]=[1.5, -3.5, 0]

$N_x = e1_y * e2_z - e2_y * e1_z = 0$

$N_y = e2_x * e1_z - e1_x * e2_z = 0$

$N_z = e1_x * e2_y - e2_x * e1_y$= (-0.5*-3.5) - (1.5 *-0.5) = 2.5

S(ΔBCP) = 2.5 / 2 = 1.25; a = 1.25 / 9 = 0.13889

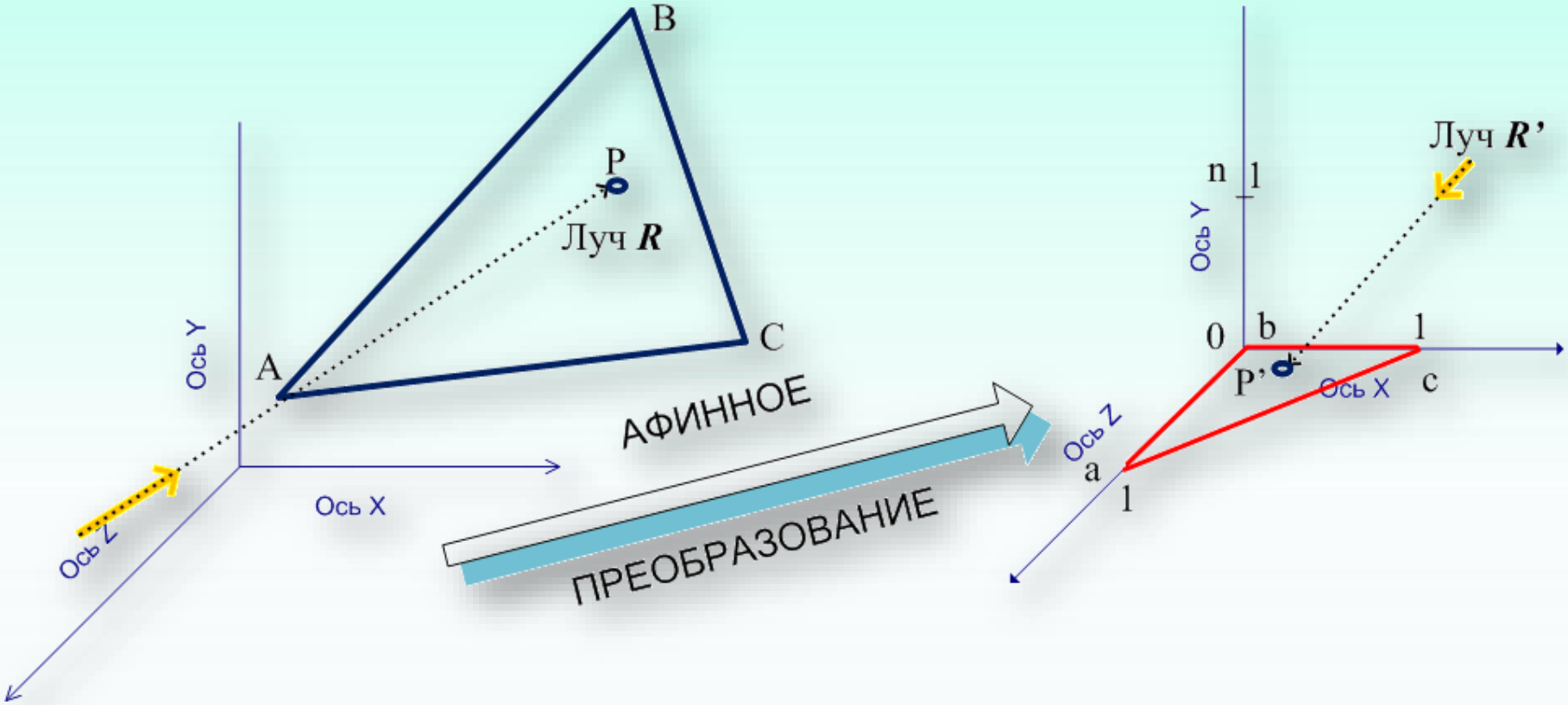
$\beta = S(\Delta APC) / S(\Delta ABC)$, $S(\Delta BCP) = |(B-P) \times (C-P)|/2$

S(ΔAPC) = 18.5 / 2 = 9.25; β = 9.25 / 9 = 1.028

t > 1 → TRUE; 0< α <1 1 → TRUE; 0< β <1 → FALSE

ПОИСК ТОЧКИ ПЕРСЕЧЕНИЯ

Unit тест



ПОИСК ТОЧКИ ПЕРСЕЧЕНИЯ

Unit тест

$T_Δ$ - преобразование из исходной системы координта в "единичную"

$T_Δ^{-1}$ - обратное преобразование из "единичной"в исходную

$$T_{\Delta}^{-1} = \begin{bmatrix} A_x - C_x & B_x - C_x & N_x - C_x \\ A_y - C_y & B_y - C_y & N_y - C_y \\ A_z - C_z & B_z - C_z & N_z - C_z \end{bmatrix} * X + \begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix}$$

Зная $T_Δ^{-1}$ можно найти $T_Δ$ и затем находит точку пересечения и оценивать "попадание"

!! Преобразование $T_Δ$ необходимо найти для каждого треугольника

Вопросы для экзамена

ТЕМА: ТРАССИРОВКА ЛУЧЕЙ.

1. Опишите идею прямой и обратной трассировки лучей.

2. Общий алгоритм рекурсивной трассировки

3. Поиск пересечения. Барицентрический тест

Литература:

http://www.ray-tracing.ru/

END 15