



# COMPUTER GRAPHICS

---

## ЗАСОБИ ПРОГРАМУВАННЯ КОМП'ЮТЕРНОЇ ГРАФІКИ



# COMPUTER GRAPHICS

---

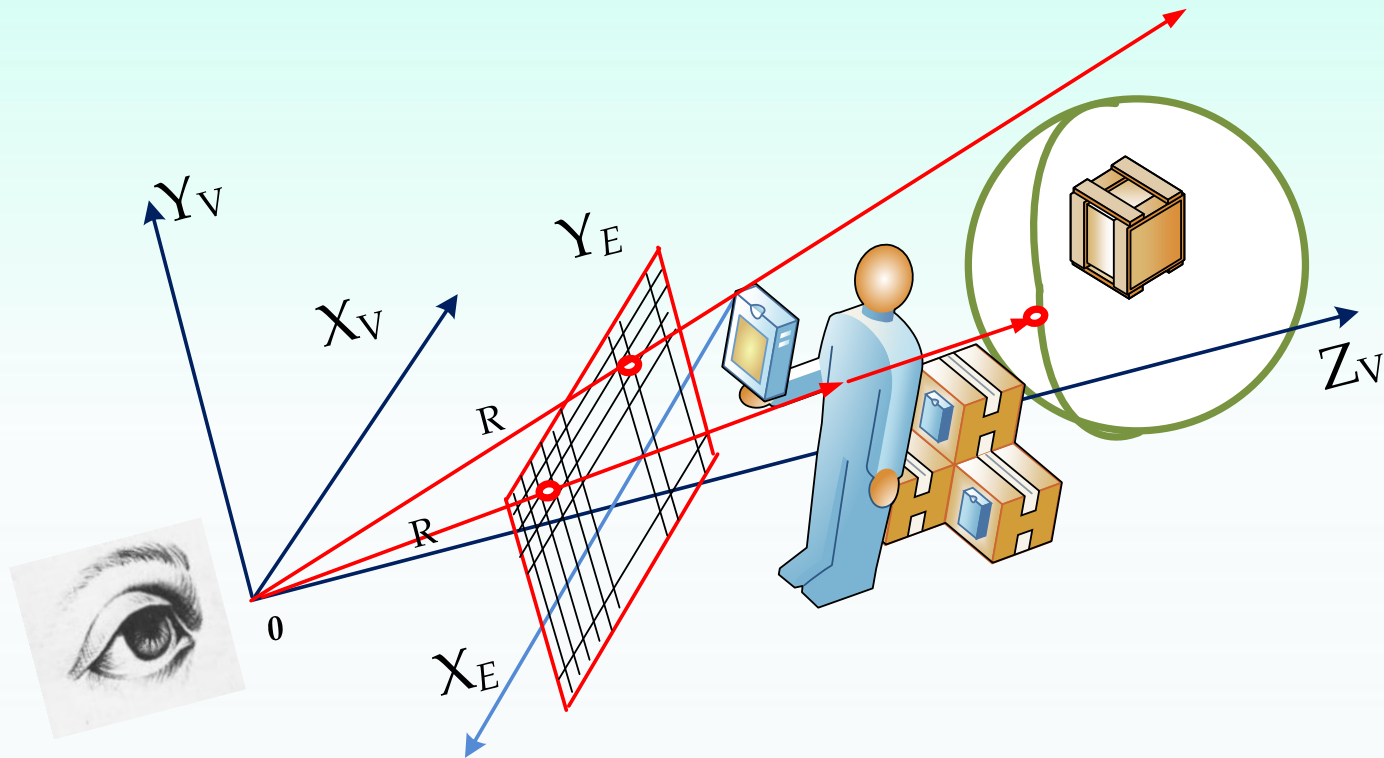
## Ray Tracing

# ТРАССИРОВКА ЛУЧЕЙ

- Ограничивающие объемы
- **ААВВ** тест (**A**xis-**A**ligned **B**ounding **B**ox).
- **Kd** – дерево.

# ОБРАТНАЯ ТРАССИРОВКА ЛУЧЕЙ

Ограничивающие объемы –  
Object Bounding Volume



# ОБРАТНАЯ ТРАССИРОВКА ЛУЧЕЙ

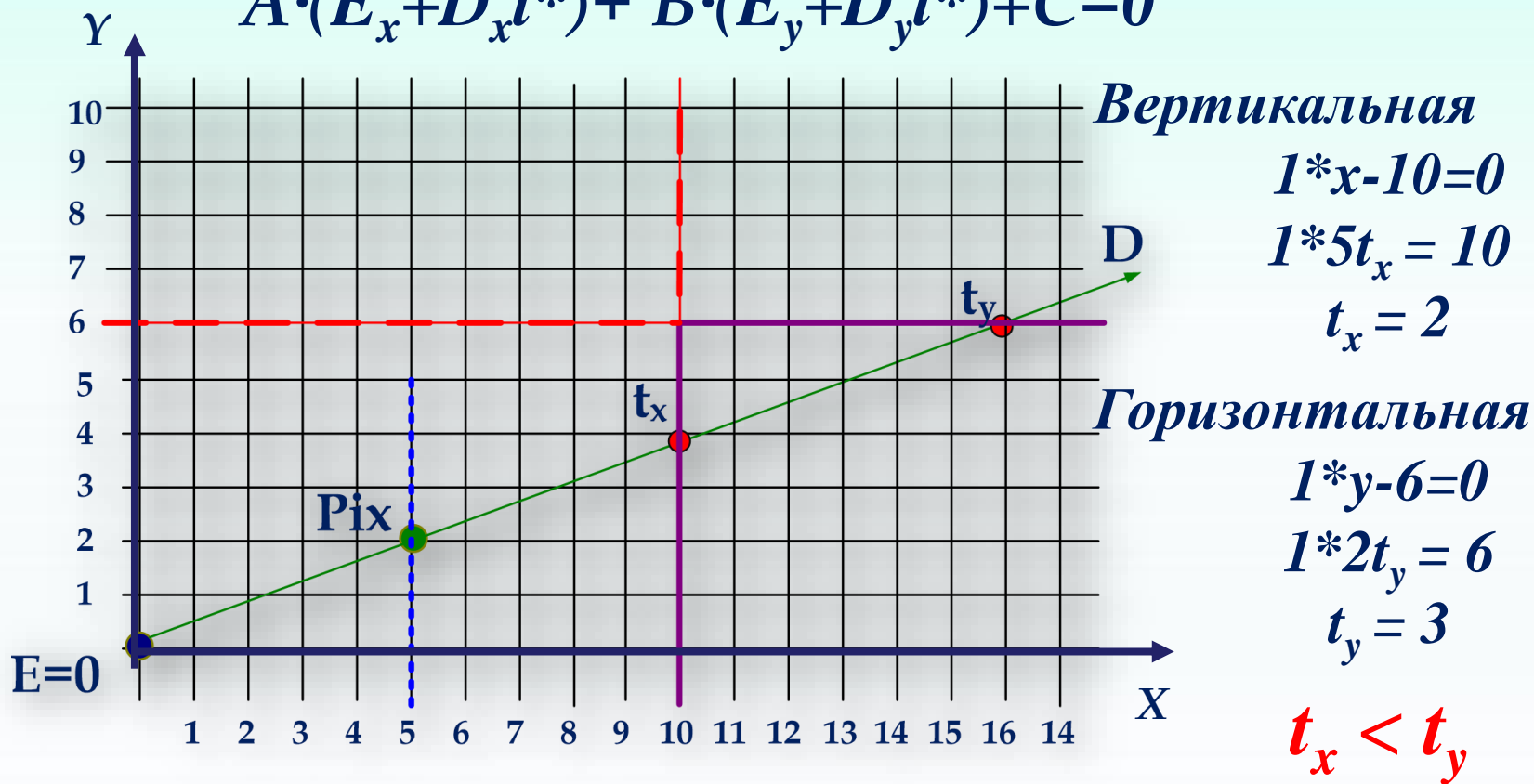
Ограничивающий объем. AABV тест. Идея

Уравнение луча  $P = E + Dt = E + (Pix - E)t$ .

Уравнение прямой  $Ax + By + C = 0$ ,

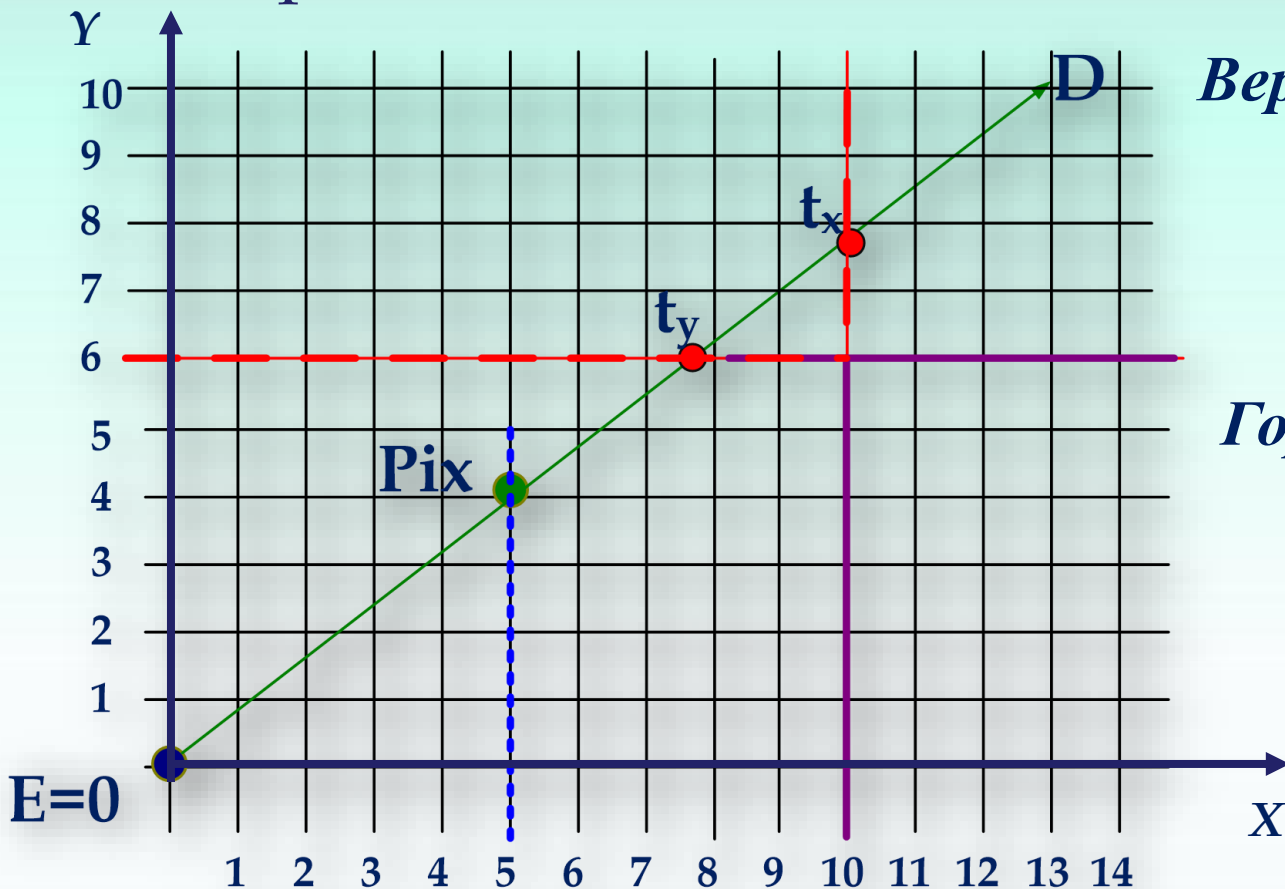
Параметр точки пересечения из уравнения

$$A \cdot (E_x + D_x t^*) + B \cdot (E_y + D_y t^*) + C = 0$$



# ОБРАТНАЯ ТРАССИРОВКА ЛУЧЕЙ

Ограничивающий объем. AABV тест. Идея



*Вертикальная*

$$1 * x - 10 = 0$$

$$1 * 5 t_x = 10$$

$$t_x = 2$$

*Горизонтальная*

$$1 * y - 6 = 0$$

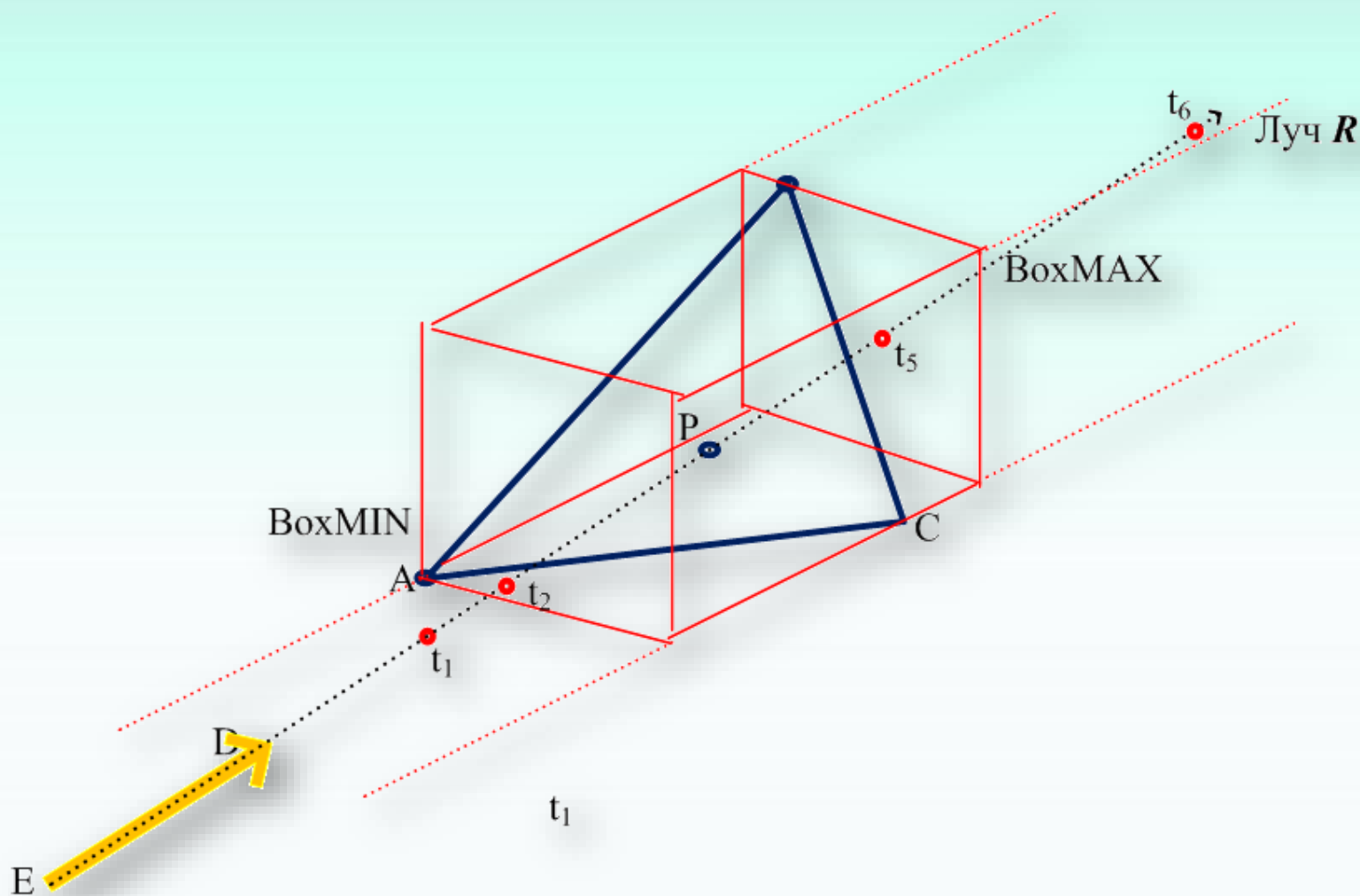
$$1 * 4 t_y = 6$$

$$t_y = 1.5$$

$$t_x > t_y$$

# ОБРАТНАЯ ТРАССИРОВКА ЛУЧЕЙ

## Ограничивающий объем. AABV тест



# ТРАССИРОВКА ЛУЧЕЙ

## Поиск пересечения. AABV тест

$$l_x = (1/D.x) * (BoxMIN.x - E.x)$$

$$h_x = (1/D.x) * (BoxMAX.x - E.x)$$

$$l_y = (1/D.y) * (BoxMIN.y - E.y)$$

$$h_y = (1/D.y) * (BoxMAX.y - E.y)$$

$$l_z = (1/D.z) * (BoxMIN.z - E.z)$$

$$h_z = (1/D.z) * (BoxMAX.z - E.z)$$

$$m_1 = \min(l_x, h_x); M_1 = \max(l_x, h_x)$$

$$m_2 = \min(l_y, h_y); M_2 = \max(l_y, h_y)$$

$$m_3 = \min(l_z, h_z); M_3 = \max(l_z, h_z)$$

$$t_{min} = \max(m_1, m_2, m_3)$$

$$t_{max} = \min(M_1, M_2, M_3)$$

Здесь:

$l_x, h_x, l_y, h_y, l_z, h_z$

параметры  $(l, h)$

пересечения луча с  
соответствующей  
плоскостью.

Если  $t_{min} < t_{max}$  то НІТ с AABV есть



# ТРАССИРОВКА ЛУЧЕЙ

Поиск пересечения. AABV тест. Пример

$E$  - наблюдатель  $[0, 0, 0]$

$P_{ix}$  - пиксель  $[0.4, 0.1, 1]$

$A$  вершина  $= [-2, 2, 4]$

$B$  вершина  $= [2, 2, 4]$

$C$  вершина  $= [2, -2, 9]$

$BoxMIN = [-2, -2, 4];$

$BoxMAX = [2, 2, 9]$

$$l_x = (1/P_{ix}.x) * (BoxMIN.x) = -5;$$

$$h_x = (1/P_{ix}.x) * (BoxMAX.x) = 5;$$

$$l_y = (1/P_{ix}.y) * (BoxMIN.y) = -20;$$

$$h_y = (1/P_{ix}.y) * (BoxMAX.y) = 20;$$

$$l_z = (1/P_{ix}.z) * (BoxMIN.z) = 4;$$

$$h_z = (1/P_{ix}.z) * (BoxMAX.z) = 9;$$

# ТРАССИРОВКА ЛУЧЕЙ

Поиск пересечения. AABV тест. Пример

$$m_1 = \min(l_x, h_x) = -5; M_1 = \max(l_x, h_x) = +5;$$

$$m_2 = \min(l_y, h_y) = -20; M_2 = \max(l_y, h_y) = +20;$$

$$m_3 = \min(l_z, h_z) = +4; M_3 = \max(l_z, h_z) = +9;$$

$$t_{\min} = \max(m_1, m_2, m_3) = +4$$

$$t_{\max} = \min(M_1, M_2, M_3) = +5$$

$$t_{\min} = 4 < t_{\max} = 5 \rightarrow \text{HIT с AABV}$$

# ТРАССИРОВКА ЛУЧЕЙ

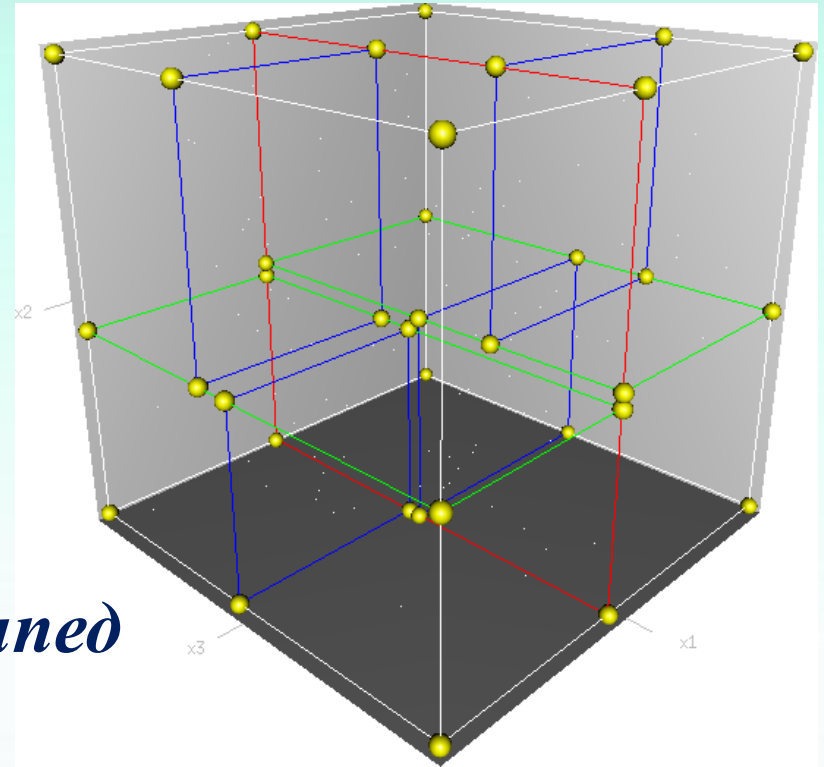
## Ускорение. kd-деревья

*Kd-дерево (K мерное дерево, Kd tree) – специальная «геометрическая» структура, выполняющее бинарное пространственное разбиение, посредством сечения K-мерного пространства (K-1) - мерными гиперплоскостями.*

*K=3: бинарное дерево, ограничивающих параллелепипедов (bounding box), вложенных друг в друга.*

# ТРАССИРОВКА ЛУЧЕЙ

## Ускорение. kd-дерева



***K=3:*** Каждый параллелепипед (bounding box) разбивается плоскостью, перпендикулярной одной из осей на два «дочерних» параллелепипеда.

# ТРАССИРОВКА ЛУЧЕЙ

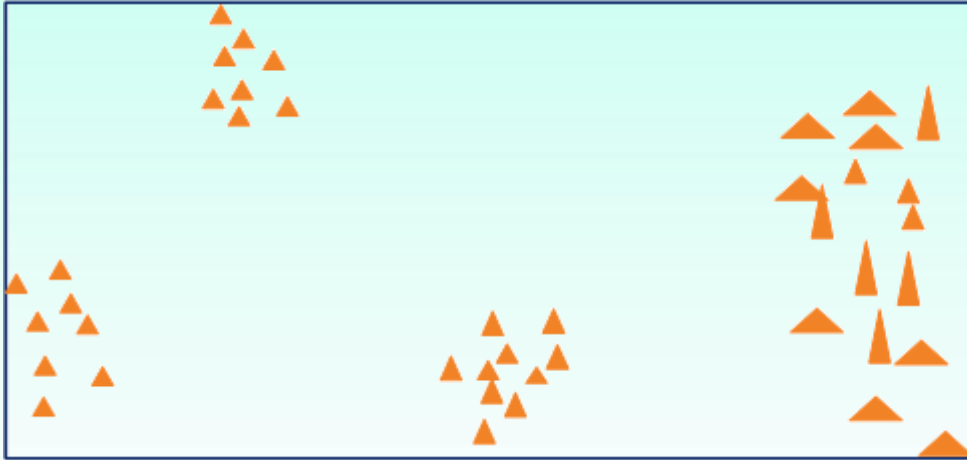
## Ускорение. kd-деревья

### *Алгоритм построения*

- 1. "Добавить" все примитивы в ограничивающий бокс.*
- 2. Если примитивов в узле **мало** или достигнут **предел глубины** дерева, завершить построение.*
- 3. **Выбрать** (?) плоскость разбиения, которая делит данный узел на два дочерних – **L** (левый узел) и **R** (правый узел).*
- 4. Добавить примитивы, пересекающиеся с боксом левого узла в **L** узел, примитивы, пересекающиеся с боксом правого узла в **R** .*
- 5. Для каждого из узлов рекурсивно выполнить данный алгоритм начиная с шага 2.*

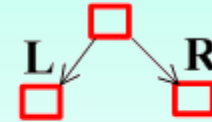
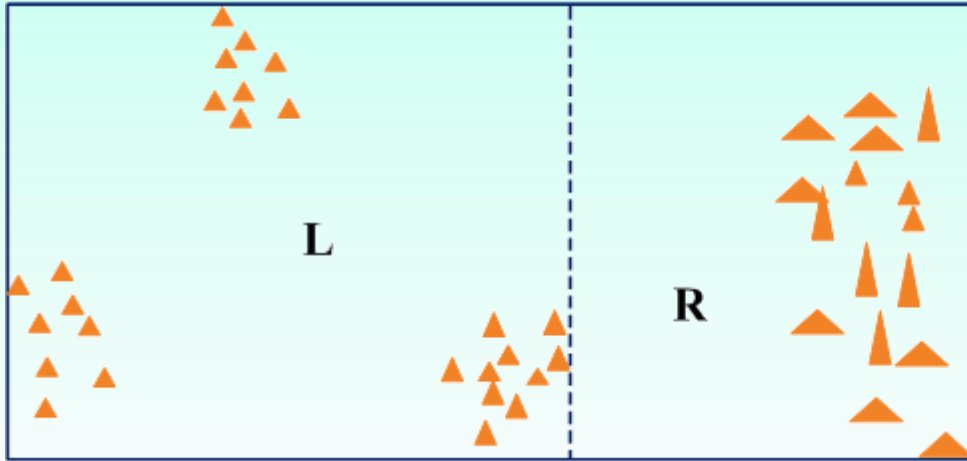
# ТРАССИРОВКА ЛУЧЕЙ

## Ускорение. kd-деревья



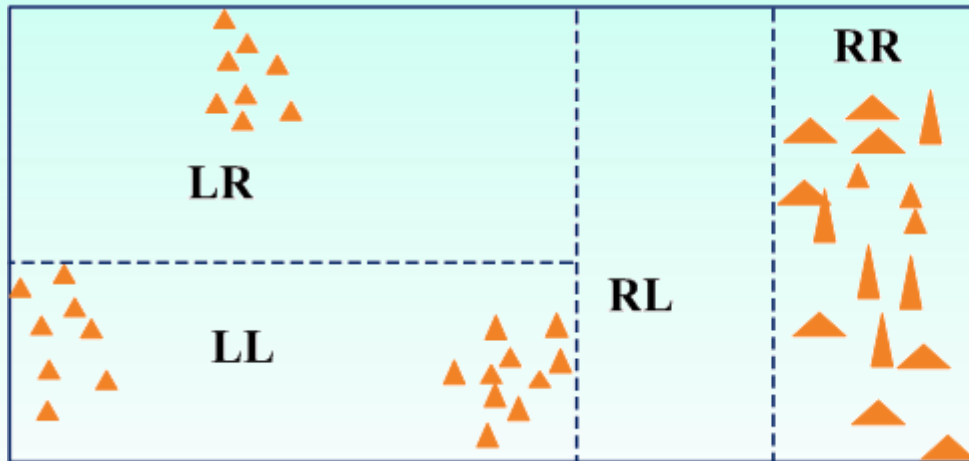
# ТРАССИРОВКА ЛУЧЕЙ

## Ускорение. kd-деревья



# ТРАССИРОВКА ЛУЧЕЙ

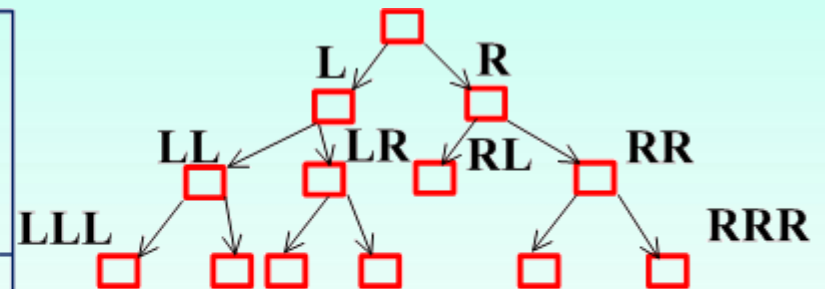
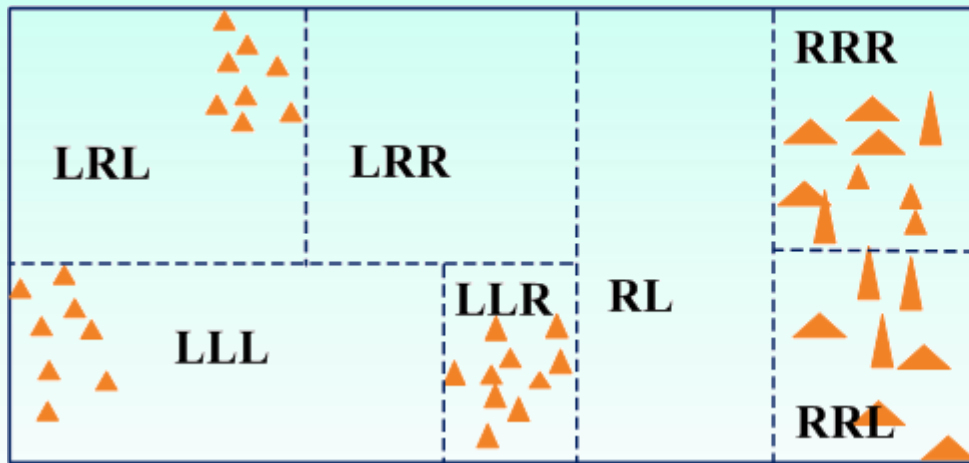
## Ускорение. kd-деревья





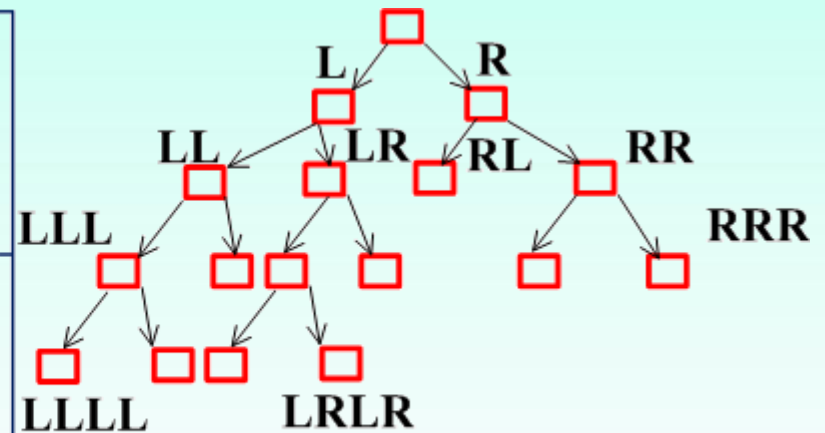
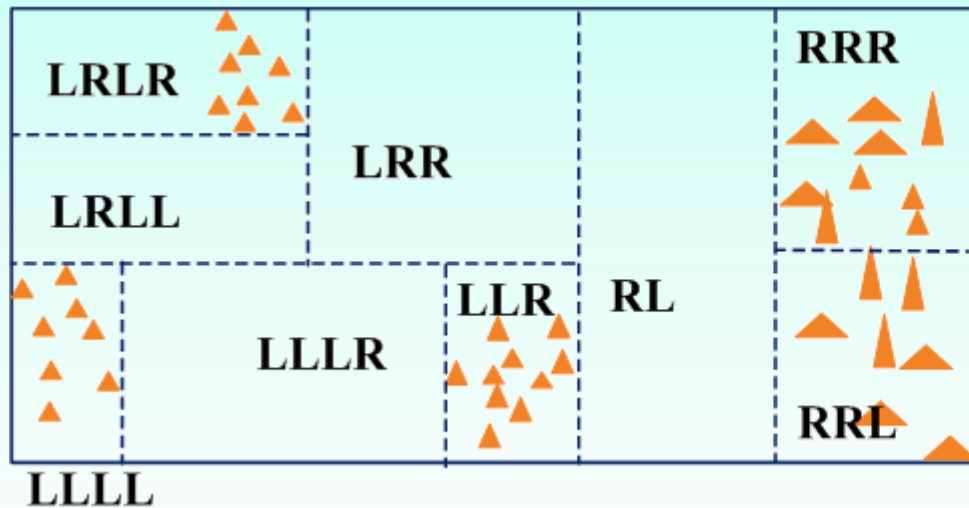
# ТРАССИРОВКА ЛУЧЕЙ

## Ускорение. kd-деревья



# ТРАССИРОВКА ЛУЧЕЙ

## Ускорение. kd-деревья



# ТРАССИРОВКА ЛУЧЕЙ

Ускорение. kd-деревья

3. **Выбрать (?)** плоскость разбиения .... **Как?**

А) По центру максимального размера бокса

Б) По медиане (слева и справа одинаковое количество треугольников)

В) Чередование сторон ( $X \rightarrow Y \rightarrow Z \rightarrow X \dots$  .)

В) Функция стоимости **SAH** (Surface Area Heuristic) – отбрасываем как можно больше пустого пространства

$$SAH(x) = CostEmpty + SurfaceArea(Left) * N(Left) + SurfaceArea(Right) * N(Right)$$

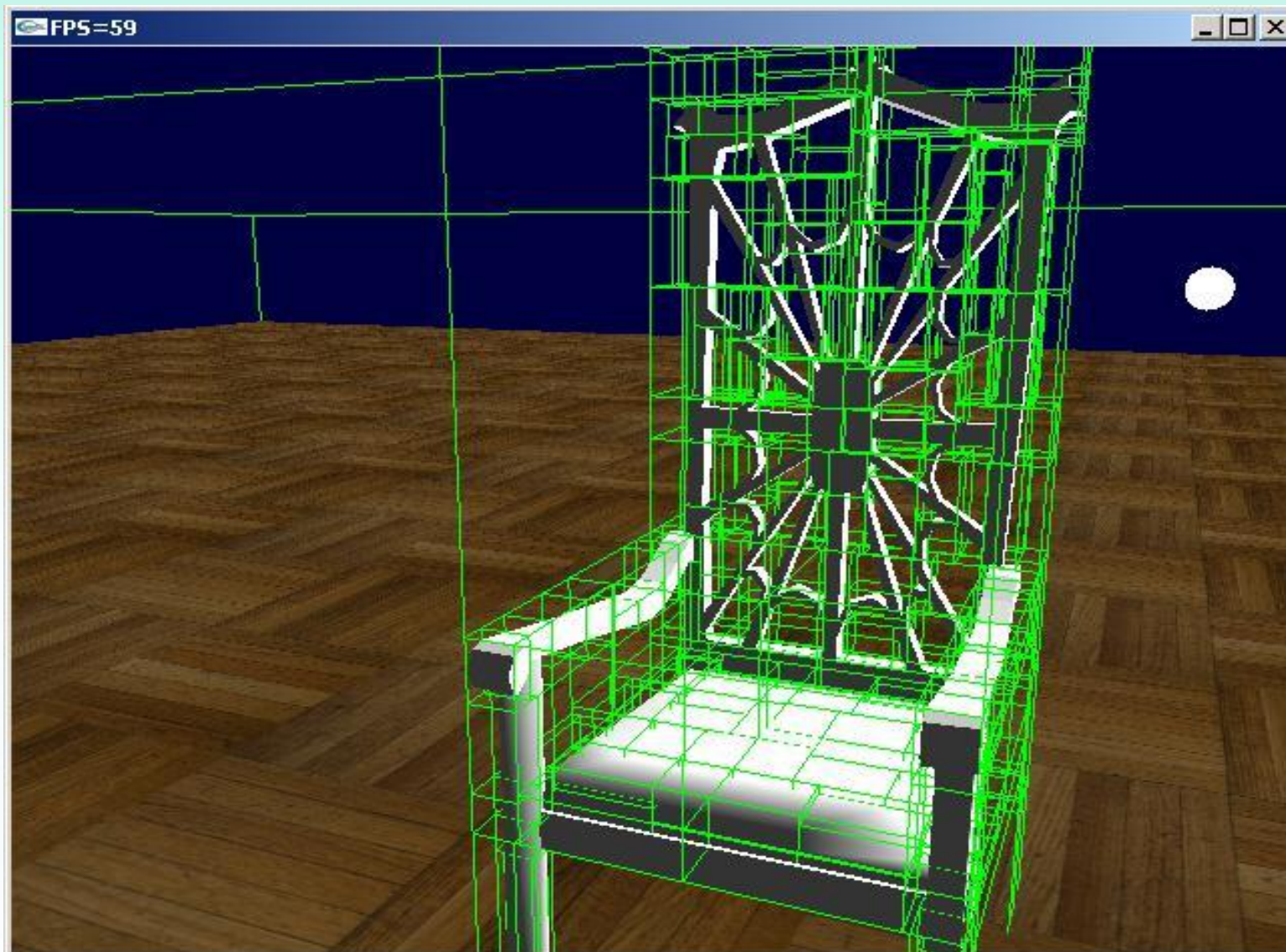
$SurfaceArea( )$  - площадь поверхности узла,

$N( )$  - число примитивов,  $x$  - координата разбиения

$$x_{\text{разбиения}} = \text{доставляет } \min SAH(x)$$

# ТРАССИРОВКА ЛУЧЕЙ

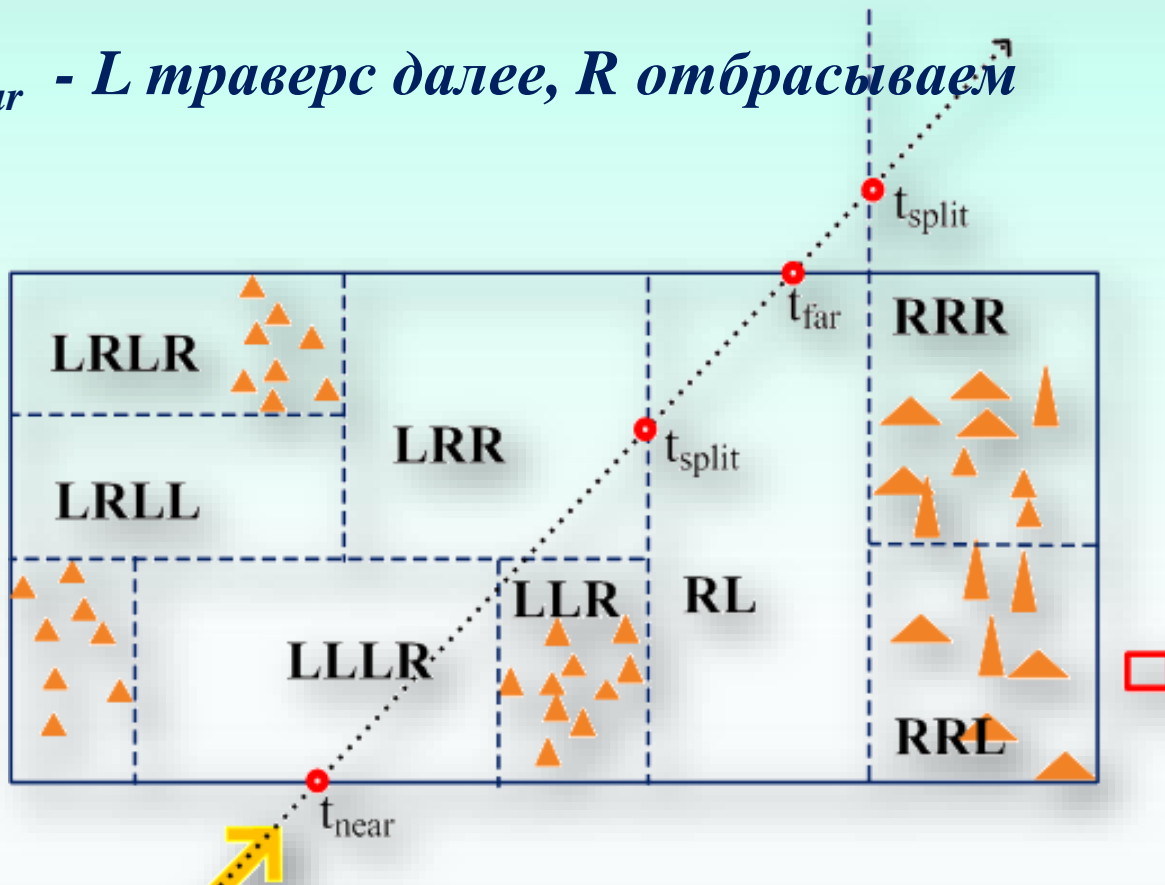
## Ускорение. kd-деревья



# ТРАССИРОВКА ЛУЧЕЙ

## Ускорение. Траверс (kd-tree traversal)

**A.  $t_{split} > t_{far}$  -  $L$  траверс далее,  $R$  отбрасываем**

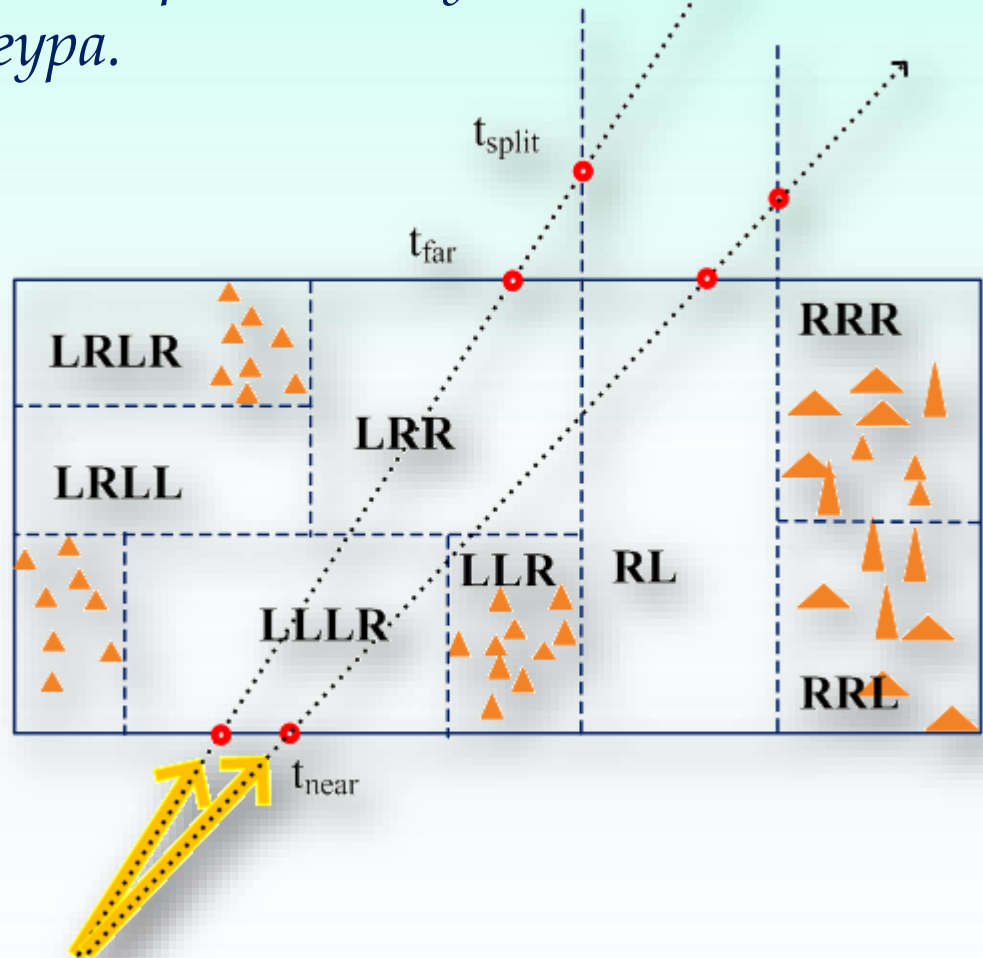


***B.  $t_{near} < t_{split} < t_{far}$  - L траверс далее, R в стек***

# ТРАССИРОВКА ЛУЧЕЙ

Ускорение. Траверс пакетов(**pocket traversal**)

**ИДЕЯ.** Группа лучей ограничивается некоторой фигурой - обычно *frustum* пирамидой и при поиске учитывается сначала эта ограничивающая фигура.



# Вопросы для экзамена

## ТЕМА: ТРАССИРОВКА ЛУЧЕЙ.

1. Ускорение трассировки. AABV тест
2. Kd деревья

## Литература:

[HTTPS://EN.WIKIPEDIA.ORG/WIKI/K-D\\_TREE](https://en.wikipedia.org/wiki/K-D_tree)

**END 16**