Вычислительная геометрия и алгоритмы компьютерной графики

Практика №2: Размещение объектов на сцене и проекция на экран

к.ф.-м.н.
Рябинин Константин Валентинович

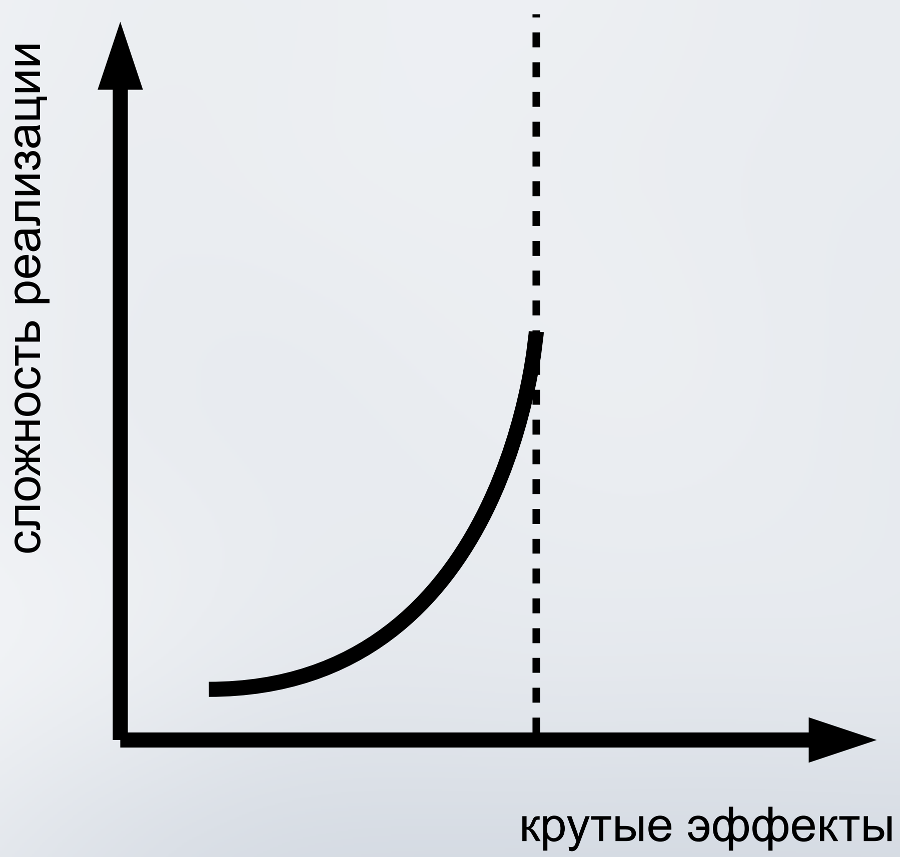
e-mail: kostya.ryabinin@gmail.com

Пермь, 2016

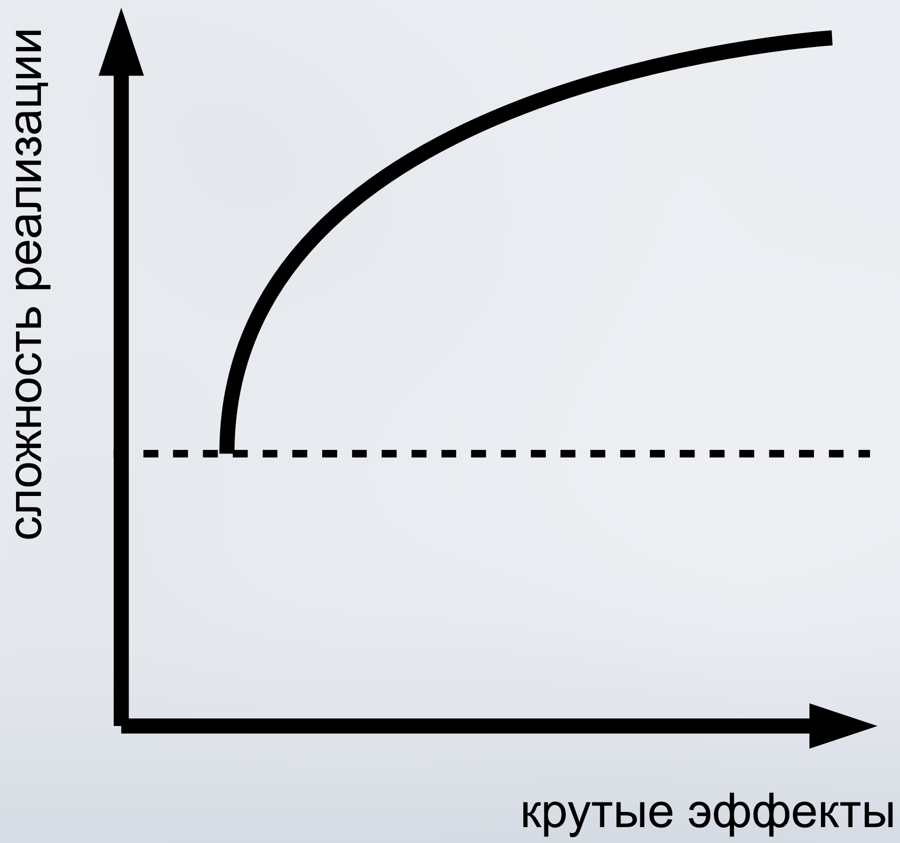
Рубим хвосты :)

- Грубо говоря, есть два класса версий OpenGL

OpenGL 1.x – «Old School»



OpenGL 3.3+ – «Brand New»



Структура объектов

- **Объекты в полигональной 3D- графике представляют собой поверхности, аппроксимированные множеством многоугольников**

Структура объектов

- Атомарная управляемая единица геометрии – вершина
- Вершина имеет набор атрибутов (типа float), интерпретация которых, вообще говоря, лежит на программисте:
  - Координаты в пространстве
  - Координаты нормали
  - Координаты текстуры
  - Цвет
  - ...
- Вершины объединяются в примитивы, чаще всего – треугольники

Построение картинки

- Сцена состоит из объектов
- Объекты состоят из примитивов
- Примитивы состоят из вершин
- Вершины имеют 3D-координаты
- Чтобы получить картинку, нужно
  - Разместить объекты в пространстве
  - Спроецировать их примитивы на плоскость экрана
  - Подобрать для проекций пиксели (растеризация)
  - Раскрасить пиксели

Размещение объектов

- Каждый объект имеет локальную систему координат
- Все типовые задачи «размещения» объектов решаются при помощи аффинных преобразований вершин этих объектов:
  - Параллельного переноса
  - Масштабирования
  - Поворота
  - Наклона

Аффинные преобразования

- **Аффинные преобразования – отображение пространства в себя, при котором параллельные прямые переходят в параллельные прямые:**

  $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$

  $f(x) = M \cdot x + v$,

  где M – матрица аффинного преобразования размерности n,

  $v \in \mathbb{R}^n$

- **Иначе говоря, преобразование называется аффинным, если его можно получить следующим образом:**
  - **Выбрать «новый» базис пространства с «новым» началом координат *v***
  - **Каждой точке *x* пространства поставить в соответствие точку *f*(*x*), имеющую те же координаты относительно «новой» системы координат, что и *x* в «старой».**

Аффинные преобразования

- В трёхмерной компьютерной графике преобразования производятся в трёхмерном пространстве (ваш К.О.)
- Помимо аффинных преобразований, используются ещё преобразования проекции
- С целью упрощения формулы и унификации аффинных и проективных преобразований, переходят к однородным координатам векторов и матрицам размерности 4x4:

  $$f\left(x\right) = M \cdot x$$

Аффинные преобразования

- **При использовании матриц все преобразования сводятся к умножению вектора однородных координат вершины на матрицу преобразования, в результате чего получается вектор «новых» координат данной вершины:**

$$\begin{pmatrix} m_0 & m_4 & m_8 & m_{12} \\ m_1 & m_5 & m_9 & m_{13} \\ m_2 & m_6 & m_{10} & m_{14} \\ m_3 & m_7 & m_{11} & m_{15} \end{pmatrix} \bullet \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix}$$

ось X новой системы координат
ось Y новой системы координат
ось Z новой системы координат
начало новой системы координат

Матрица переноса

$$\begin{pmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Матрица масштаба

$$\begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Матрица поворота вокруг оси

$$\begin{pmatrix} x^2(1-c)+c & xy(1-c)-zs & xz(1-c)+ys & 0 \\ yx(1-c)+zs & y^2(1-c)+c & yz(1-c)-xs & 0 \\ xz(1-c)-ys & yz(1-c)+xs & z^2(1-c)+c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$c = \cos\theta, \quad s = \sin\theta, \quad |(x, y, z)| = 1$$

Проекция на экран

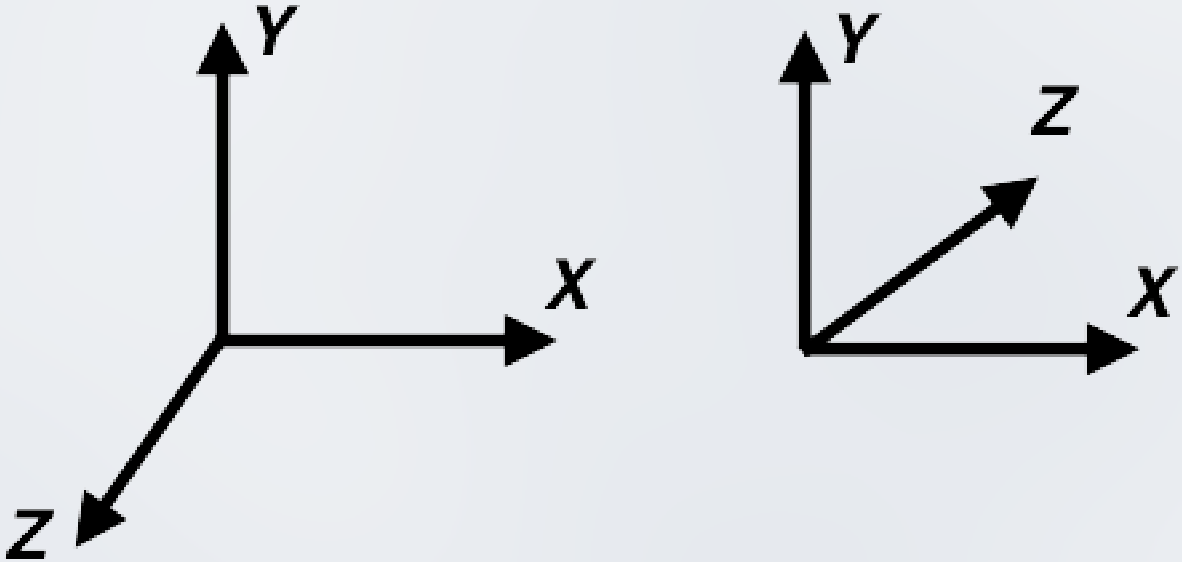
- **Выделяется фиксированная область пространства NDC в виде параллелепипеда (нормированные координаты устройства)**
- **На сцене выделяется «область интереса» S, которая должна быть видима пользователю**
- **Ко всем вершинам объектов применяется преобразование**

$$f(v) = \begin{cases} v' \in NDC, & v \in S \\ v'' \notin NDC, & v \in P, P \cap S = \emptyset \end{cases}$$

- **На экране выделяется прямоугольная область для вывода картинки (порт просмотра)**
- **Осуществляется проекция NDC в порт просмотра путём отбрасывания координаты Z и линейного масштабирования итогового множества точек до совпадения с размером порта просмотра**

NDC

- Конкретный вид NDC – конвенционализм



OpenGL
[-1; 1] x [-1; 1] x [-1; 1]

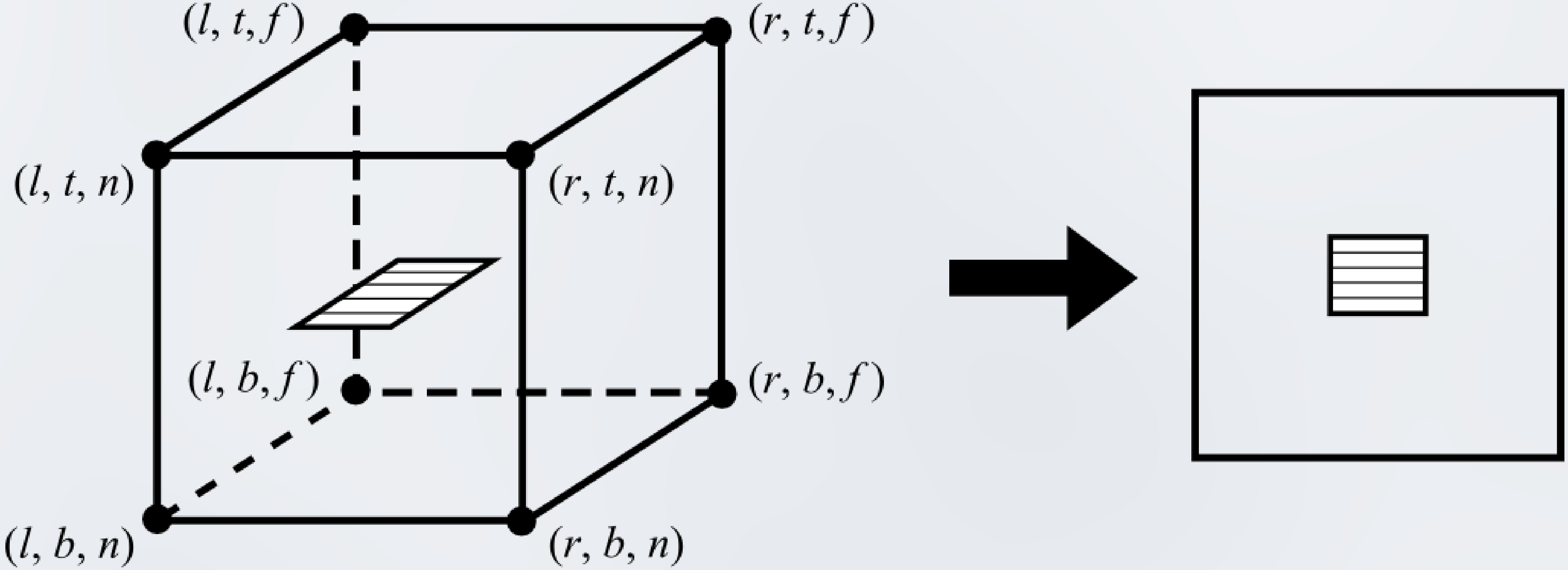
DirectX
[-1; 1] x [-1; 1] x [0; 1]

- На экран выводится только то, что в результате всех преобразований попало в NDC

Матрица проекции

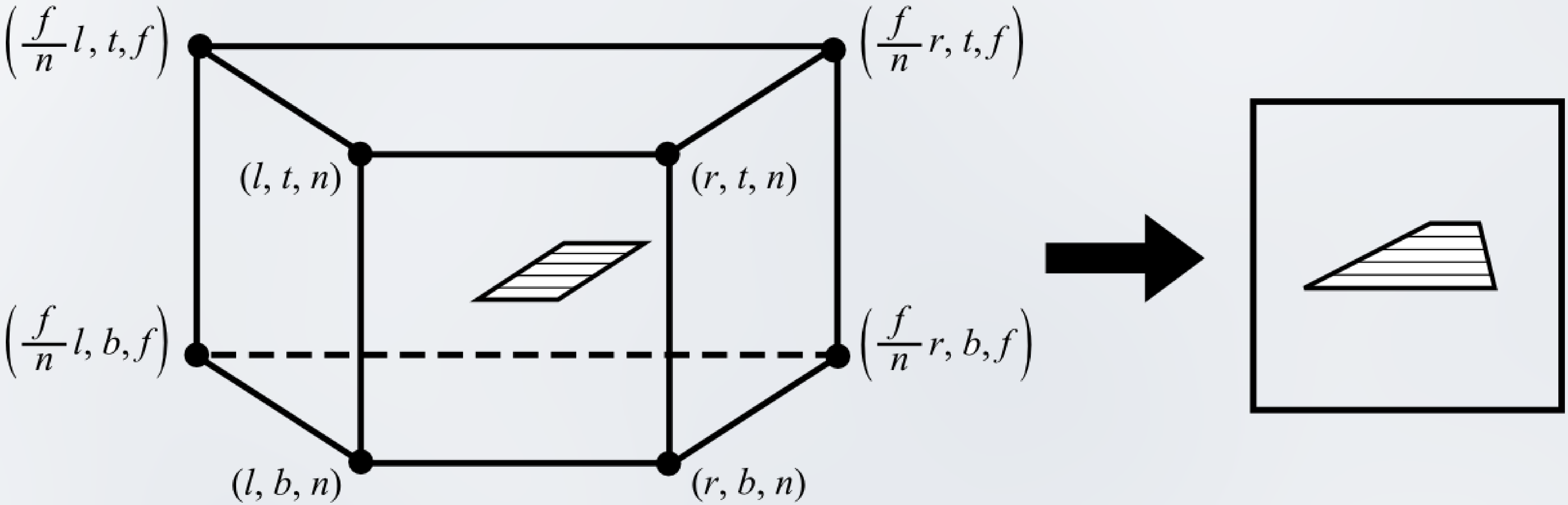
- Преобразование области интереса в NDC для единообразия также осуществляется при помощи матриц
- Матрица преобразования в данном случае носит название матрицы проекции, хотя само умножение и не изменяет размерность вектора
- Используются параллельные и перспективные проекции

Параллельная проекция



$$\begin{pmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & -\frac{2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

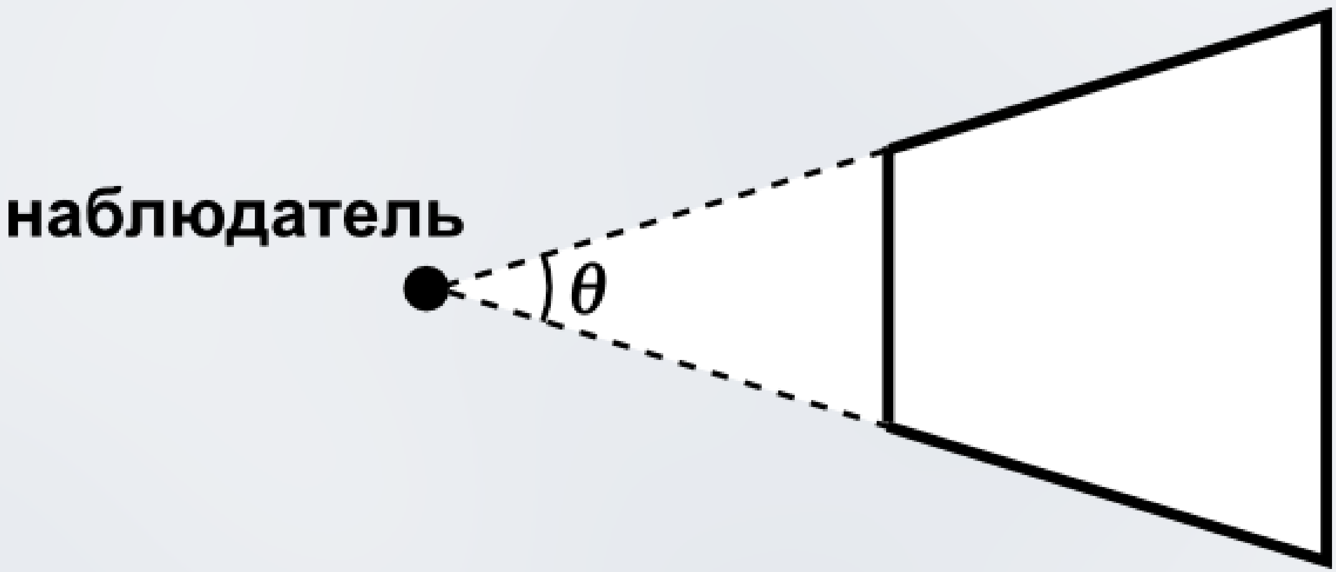
Перспективная проекция



$$\begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

После умножения матрицы на вектор,
x, y и z результата делятся на w результата

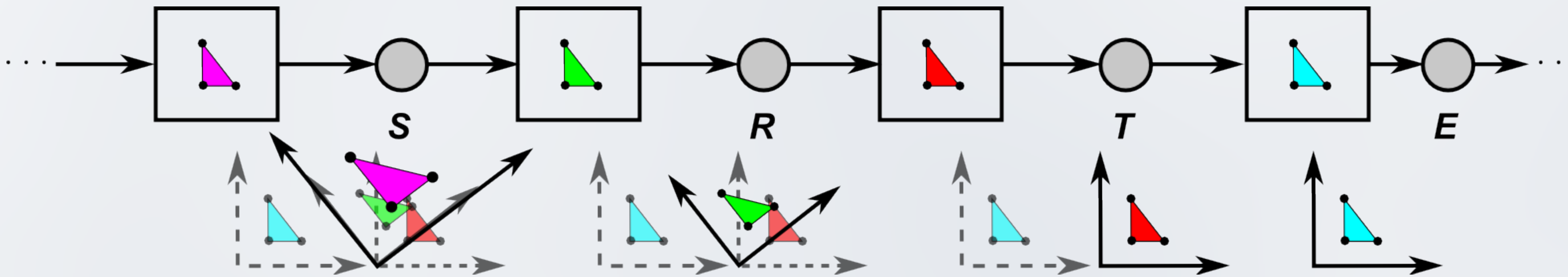
Перспективная проекция



$$-l = r = n\,tg\frac{\theta}{2}, \quad -b = t = n\frac{w}{h}tg\frac{\theta}{2}$$

Комбинируемость

- **Важным свойством матричных преобразований является их комбинируемость:**



- **В связи с этим, в компьютерной графике имеется паттерн хранения и применения преобразований Model-View-Projection:**

$v' = (P \cdot V \cdot M) \cdot v$

v' – вектор координат, передаваемый системе для произведения растеризации

v – вектор координат вершины

P – матрица проекции

V – матрица вида (преобразование камеры)

M – матрица модели (преобразование размещения объекта на сцене)

$M = M_{родителя} \cdot M_{объекта}$

Камера

- **Камера – это псевдообъект в трёхмерном пространстве, характеризующий положение наблюдателя**
- **Камера – лишь полезная метафора, на низком уровне она выражена матричным преобразованием, математически ничем не отличающимся от всех остальных**
- **Часто преобразование камеры является лишь аффинным**
- **В связи с этим, *иногда* преобразование камеры не хранят отдельно, а «смешивают» его с преобразованием резмещения, получая матрицу, которую принято называть ModelView (в «олдскульном» OpenGL было именно так)**

Преобразование координат

- **В итоге, преобразование координат, осуществляемое в графическом приложении, имеет вид:**

*** В новых версиях OpenGL, преобразования из первого ряда должен выполнять программист,а преобразования из второго ряда система выполняет автоматически**