

# **Виртуальная реальность и мультимедиа**

***Математические и технические  
основы компьютерной графики***

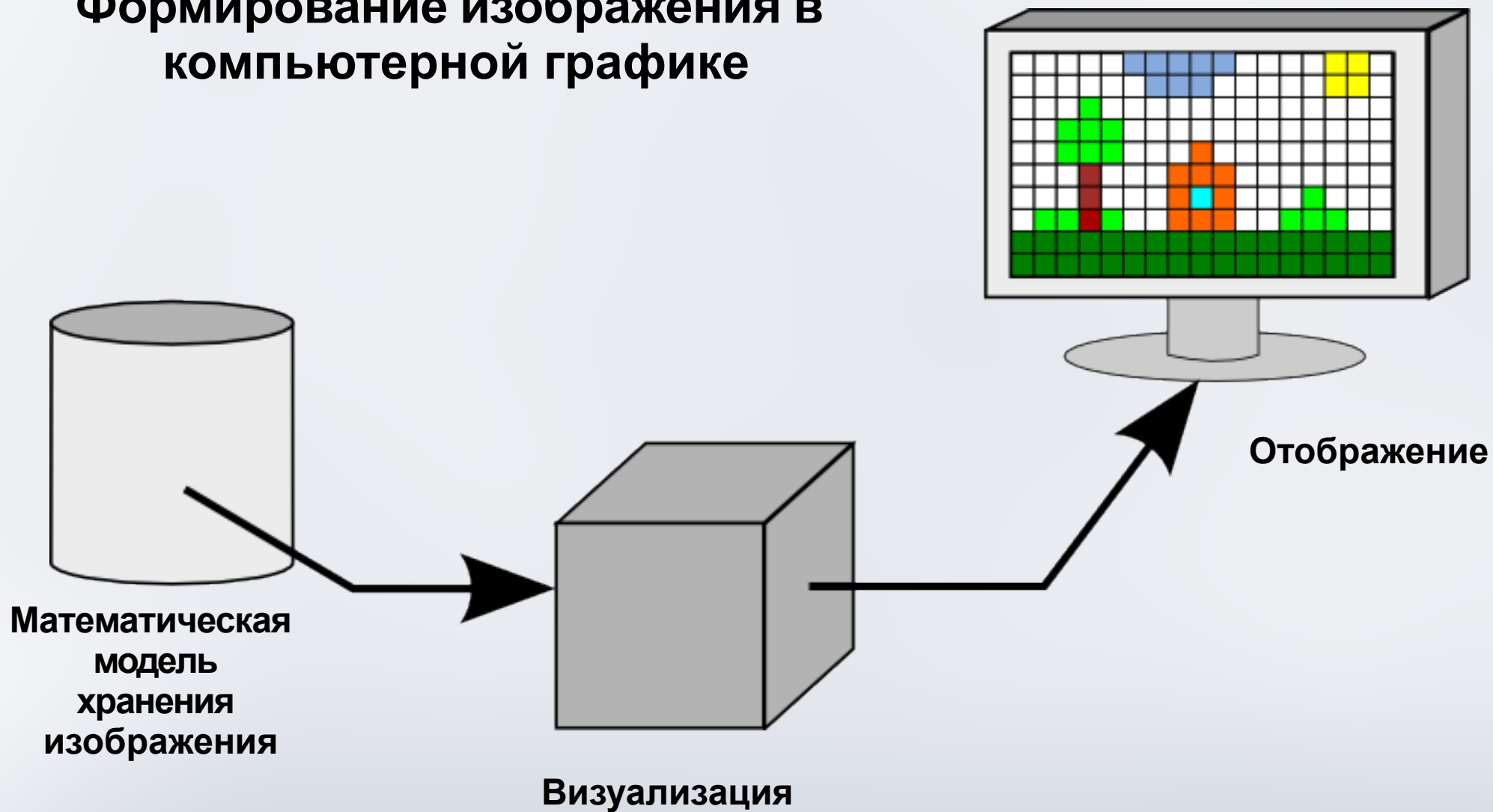
к.ф.-м.н.

Рябинин Константин Валентинович

e-mail: [kostya.ryabinin@gmail.com](mailto:kostya.ryabinin@gmail.com)

Пермь, 2015

## Формирование изображения в компьютерной графике



## Классификация графики по способу хранения (от способа хранения зависит визуализация)

	2D	3D
Растровая	Матрица точек	Воксели
Векторная	Описание контуров и заливок	Описание многоугольников

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



- Многоугольники (примитивы) описываются как множество вершин и связей между ними
- Атомарная управляемая единица геометрии – **вершина**
- Вершина имеет набор атрибутов (координаты в пространстве, координаты нормали, координаты текстуры, ...)
- Чтобы разместить объекты на сцене, над вершинами этих объектов производятся преобразования координат
- Вводятся различные системы координат (локальные координаты объекта, глобальные координаты сцены, NDC, ...)
- Канонически растеризации подвергаются лишь те примитивы (части примитивов), которые попали в  $[-1; 1] \times [-1; 1] \times [-1; 1]$  NDC

- Задача преобразования координат вершины из одной системы в другую удобно решается матрицами и векторами:

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

ось X новой системы координат

ось Y новой системы координат

ось Z новой системы координат

начало новой системы координат

Матрица переноса

Матрица масштаба

Матрица поворота вокруг оси

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

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

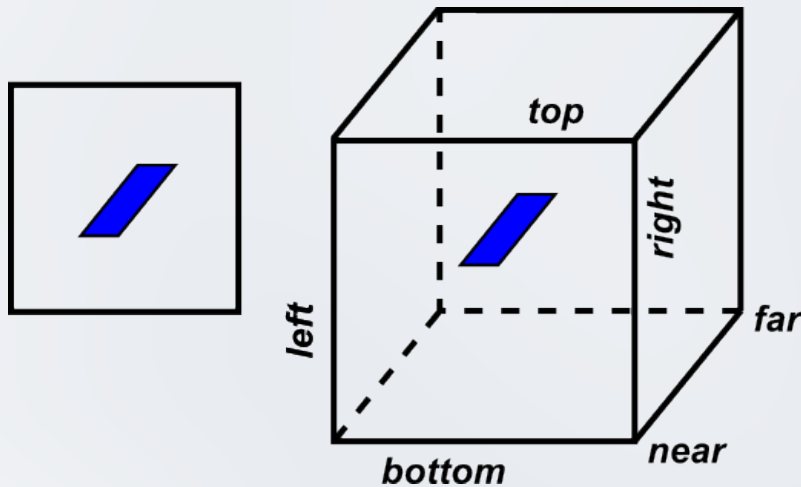
$$\begin{pmatrix} x^2(1-c)+c & xy(1-c)-zs & xz(1-c)+ys & 0 \\ yx(1-c)+zs & y^2(1-c)+c & yz(1-c)-xs & 0 \\ xz(1-c)-ys & yz(1-c)+xs & z^2(1-c)+c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

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

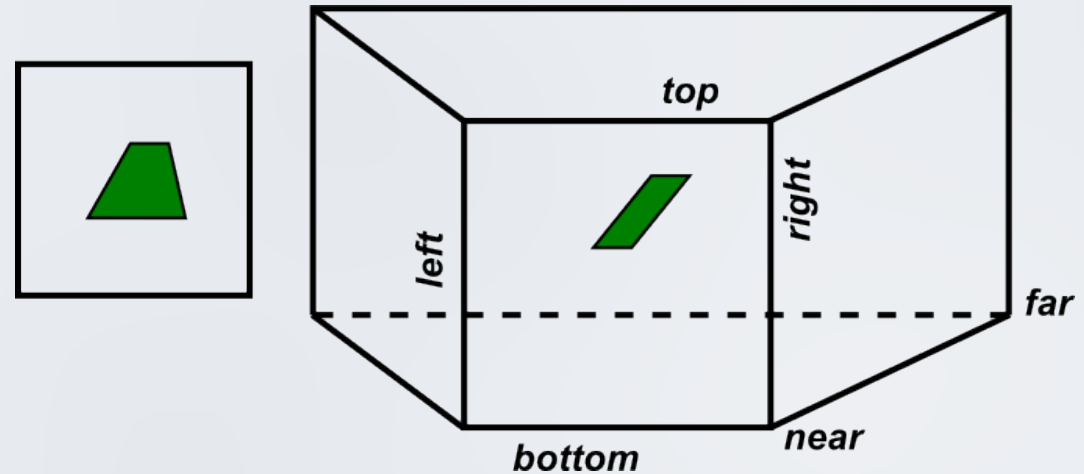
- Комбинирование преобразований осуществляется перемножением матриц

- Проецирование объектов на экран также сводится к преобразованию координат и может быть единообразно осуществлено матрицами:

## Параллельная проекция



## Перспективная проекция



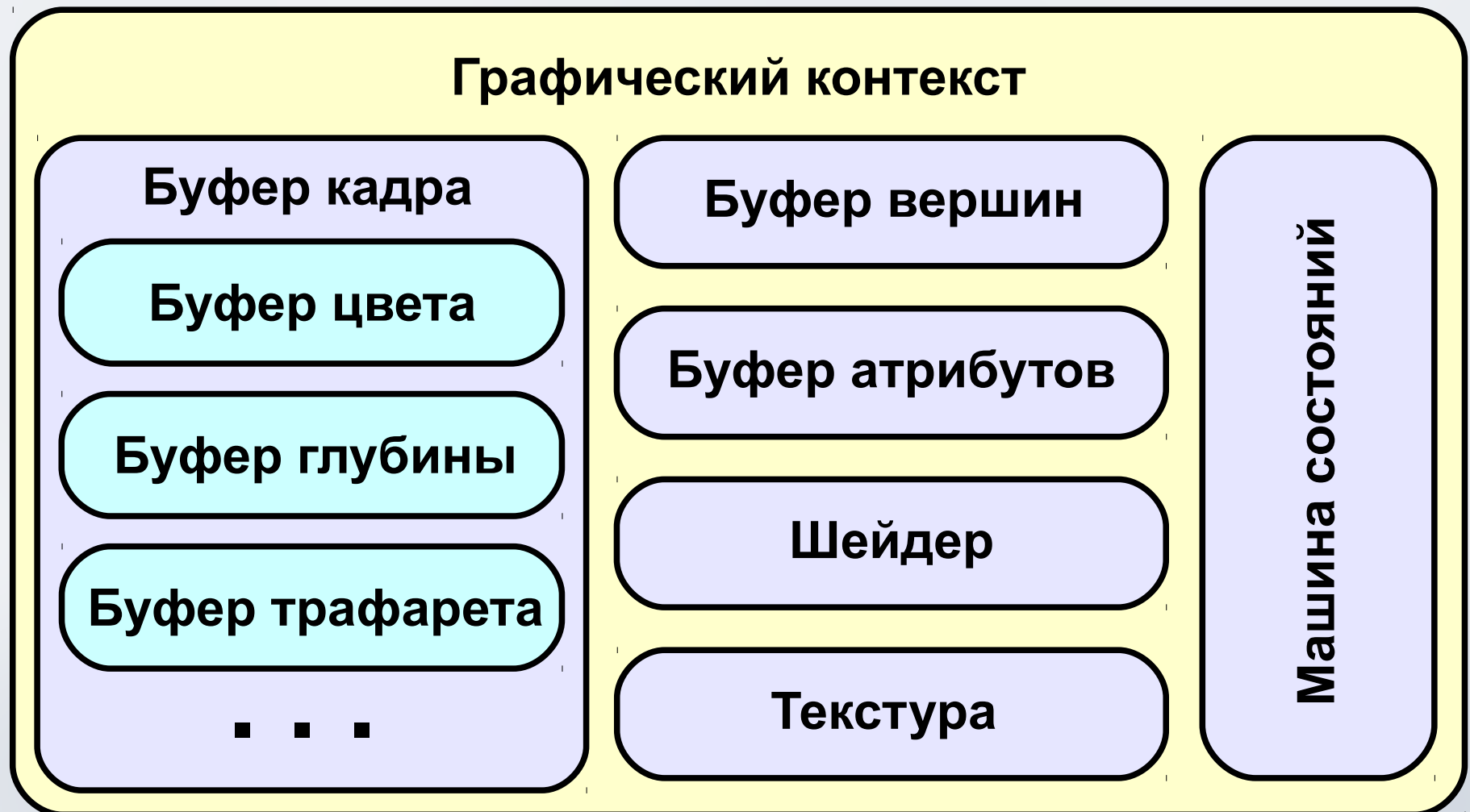
$$\begin{pmatrix} \frac{2}{\text{right} - \text{left}} & 0 & 0 & -\frac{\text{right} + \text{left}}{\text{right} - \text{left}} \\ 0 & \frac{2}{\text{top} - \text{bottom}} & 0 & -\frac{\text{top} + \text{bottom}}{\text{top} - \text{bottom}} \\ 0 & 0 & \frac{-2}{\text{far} - \text{near}} & -\frac{\text{far} + \text{near}}{\text{far} - \text{near}} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

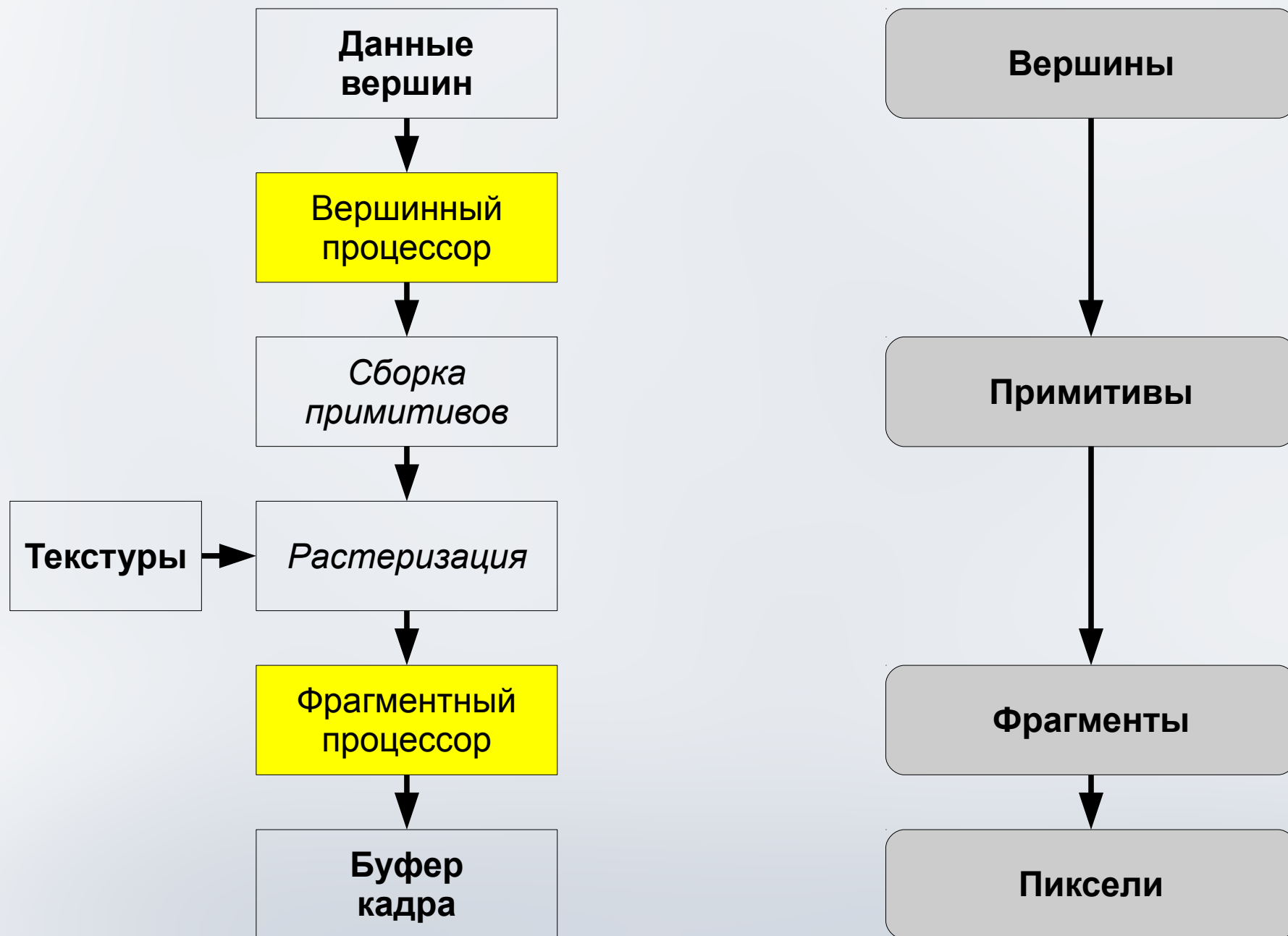
$$\begin{pmatrix} \frac{2 \text{ near}}{\text{right} - \text{left}} & 0 & \frac{\text{right} + \text{left}}{\text{right} - \text{left}} & 0 \\ 0 & \frac{2 \text{ near}}{\text{top} - \text{bottom}} & \frac{\text{top} + \text{bottom}}{\text{top} - \text{bottom}} & 0 \\ 0 & 0 & -\frac{\text{far} + \text{near}}{\text{far} - \text{near}} & -\frac{2 \text{ far near}}{\text{far} - \text{near}} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

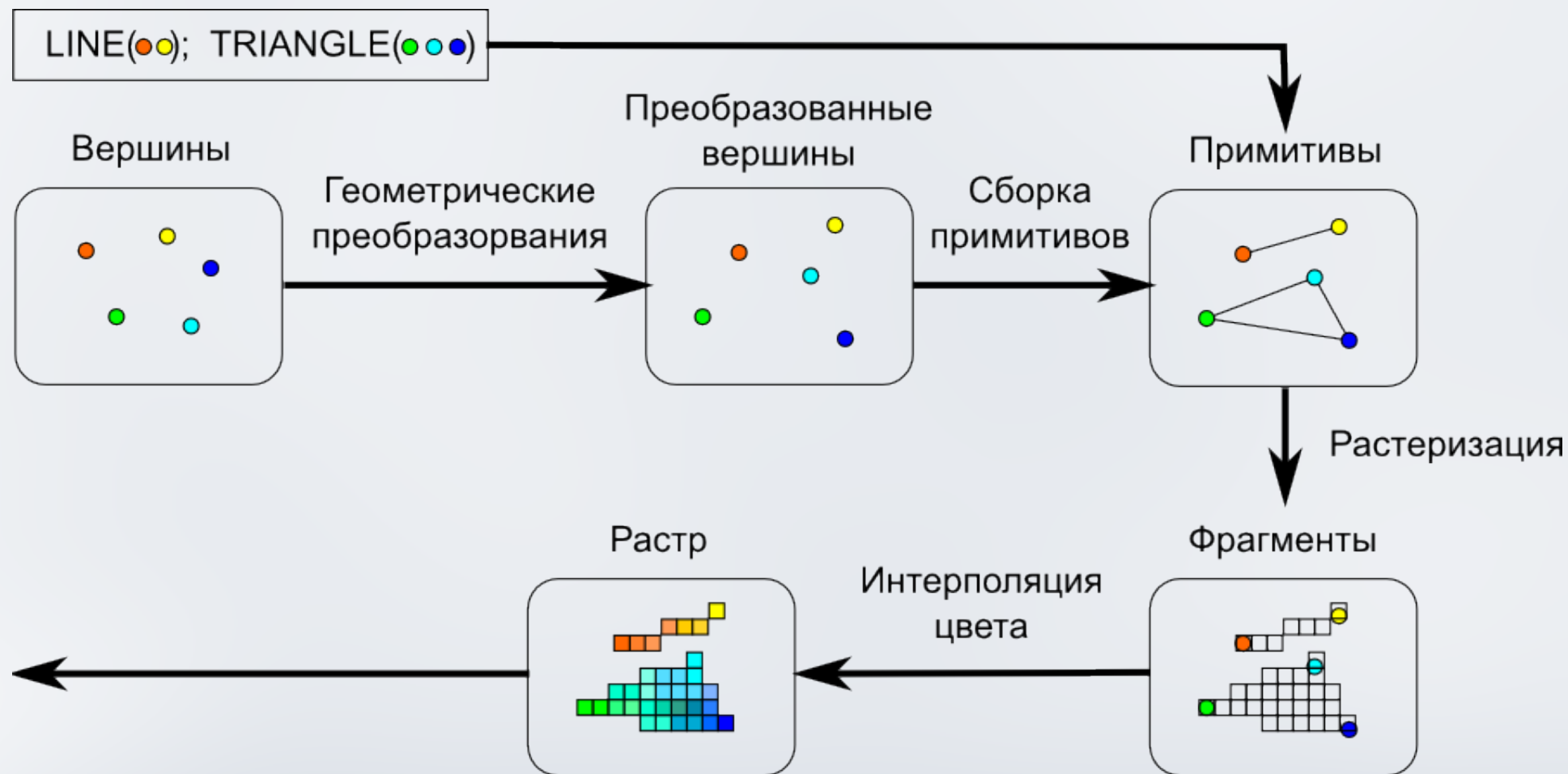
- Камера – это псевдообъект в трёхмерном пространстве, характеризующий положение наблюдателя
- Камера – лишь полезная метафора, на низком уровне она выражена матричным преобразованием, математически ничем не отличающимся от всех остальных
- Часто преобразование камеры является лишь аффинным
- Для удобства выражения соответствующих преобразований вводят паттерн хранения матриц Model-View-Projection



- Большая часть графических API имеют следующую концептуальную организацию данных для визуализации:







- Шейдер – это микропрограмма, написанная на специальном ЯП для одной из ступеней графического конвейера
- Шейдеры служат для единообразного преобразования атрибутов объектов и создания различных визуальных эффектов
- Виды шейдеров:
  - Вершинные
  - Геометрические
  - Фрагментные
- Шейдеры более ранних этапов конвейера готовят данные для шейдеров более поздних этапов
- Шейдерная программа – группа шейдеров, полностью определяющих все этапы конвейера (по шейдеру на этап)

Совпадение имён  
Интерполяция значений

```
/* VERTEX SHADER */
uniform mat4 u_mvp;
attribute vec3 a_coord;
attribute vec4 a_color;
varying vec4 v_color;
void main()
{
    v_color = a_color;
    gl_Position = u_mvp * vec4(a_coord, 1.0);
}

/* FRAGMENT SHADER */
varying vec4 v_color;
void main()
{
    gl_FragColor = v_color;
}
```