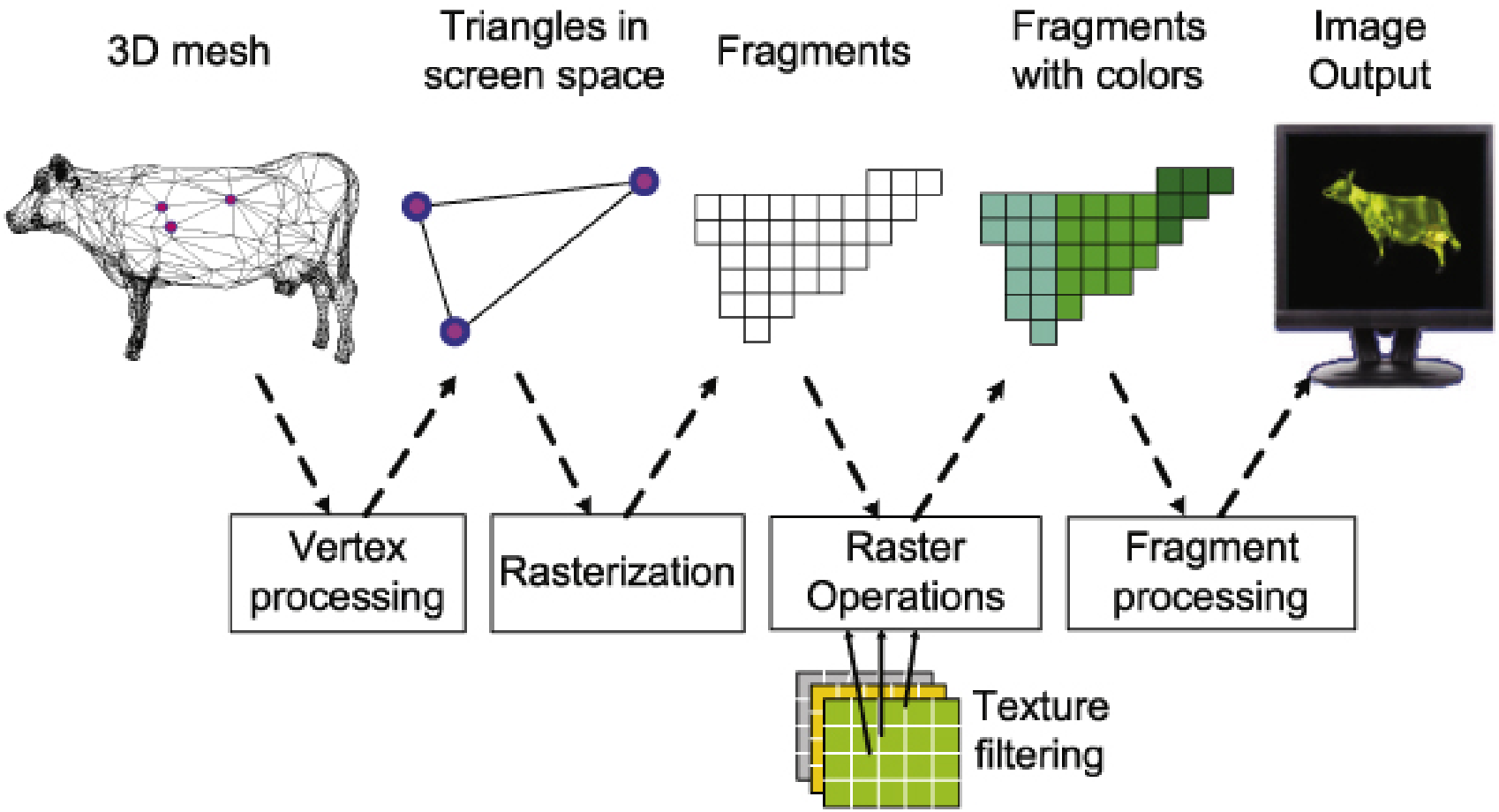
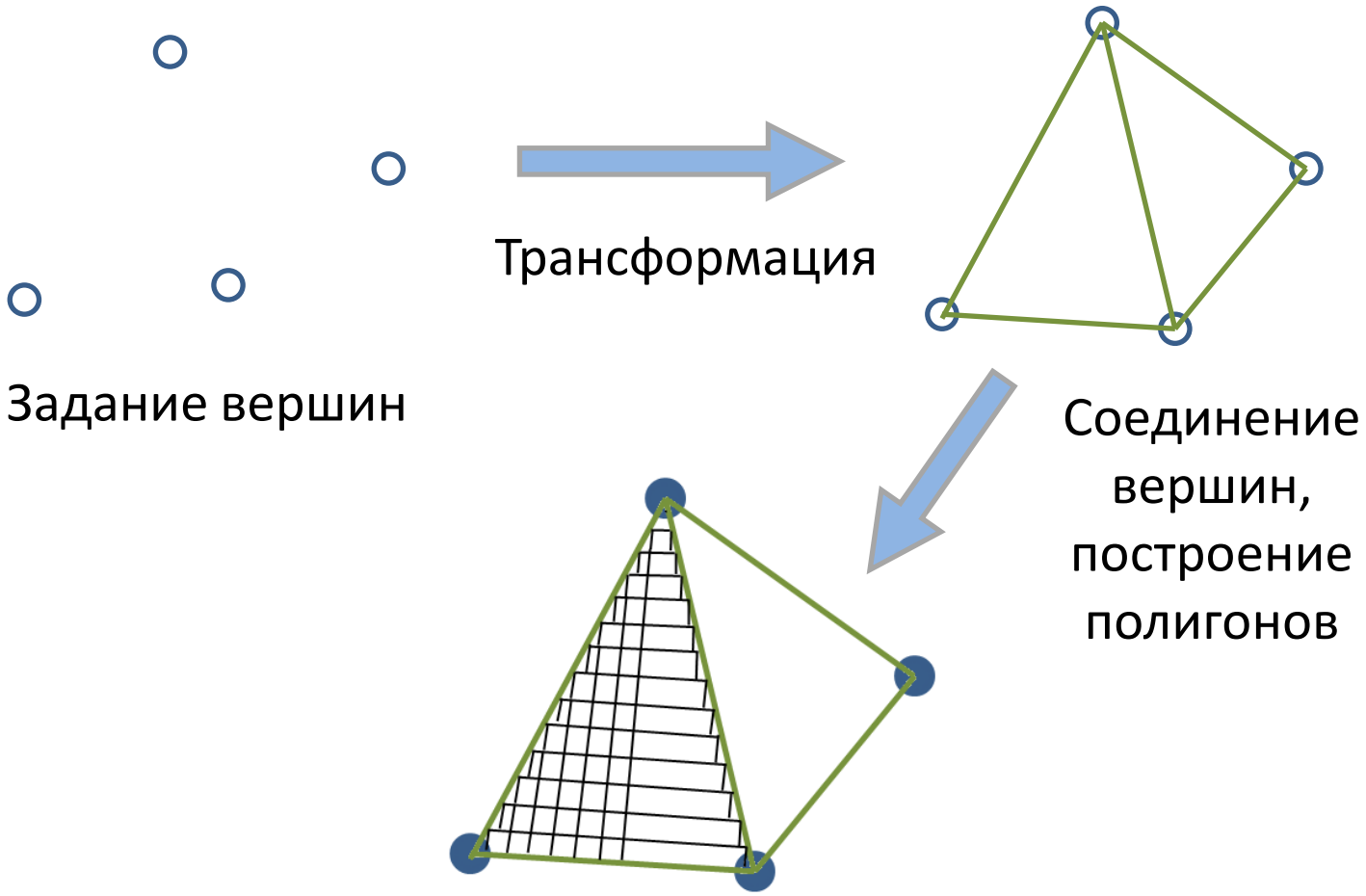
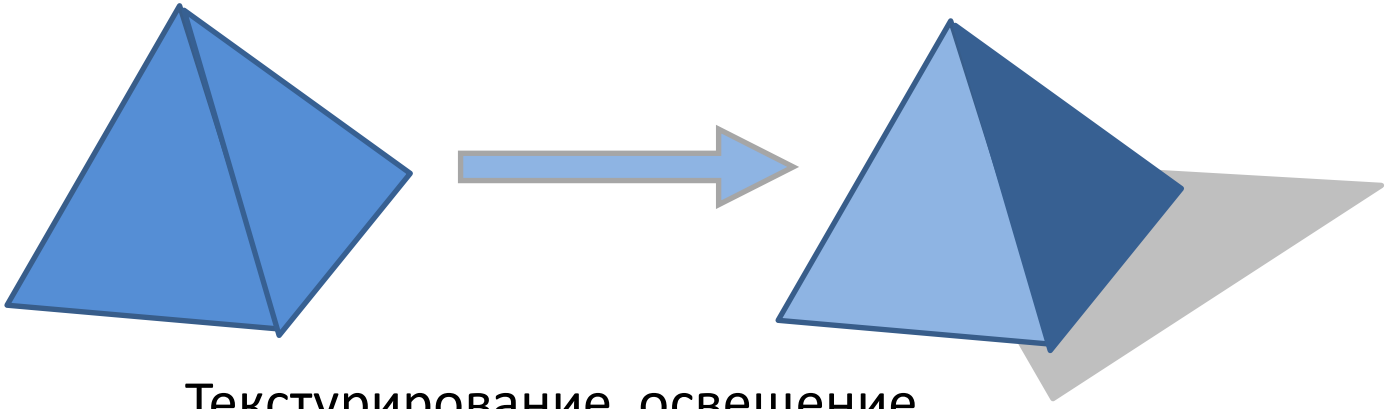
Графический конвейер



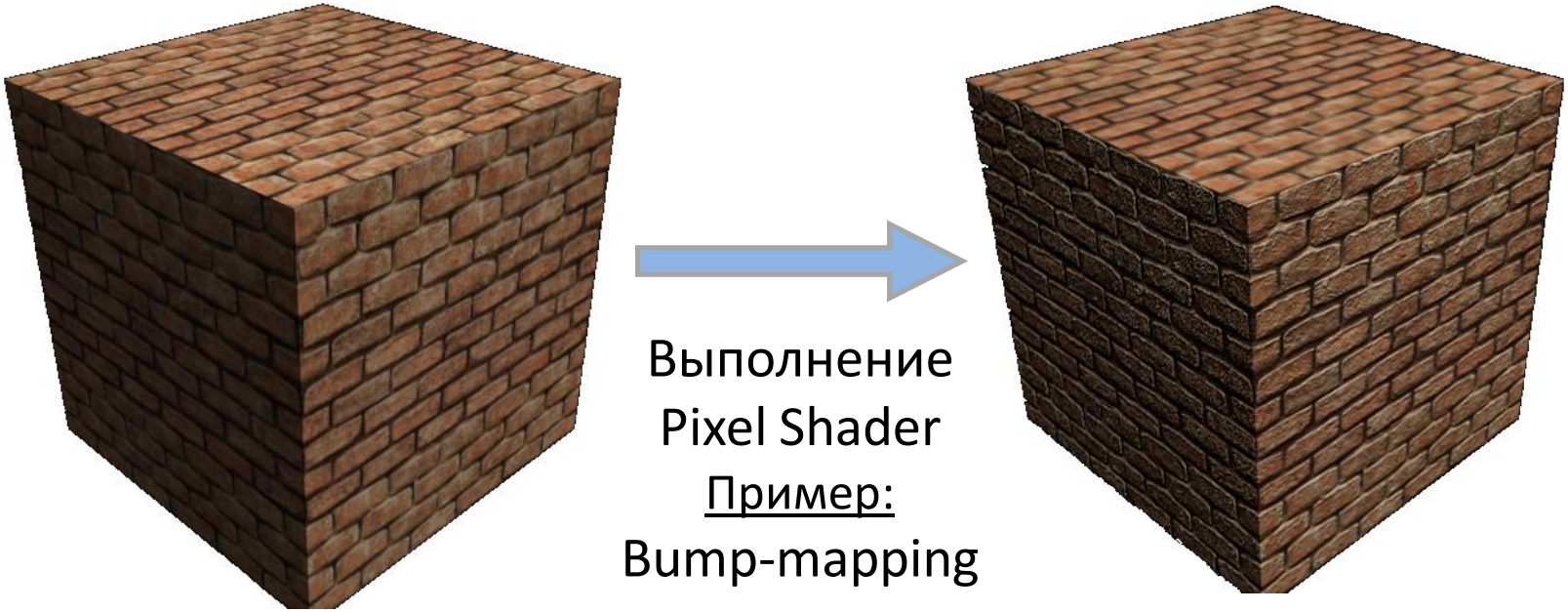
Создание трехмерного объекта



Создание трехмерного объекта



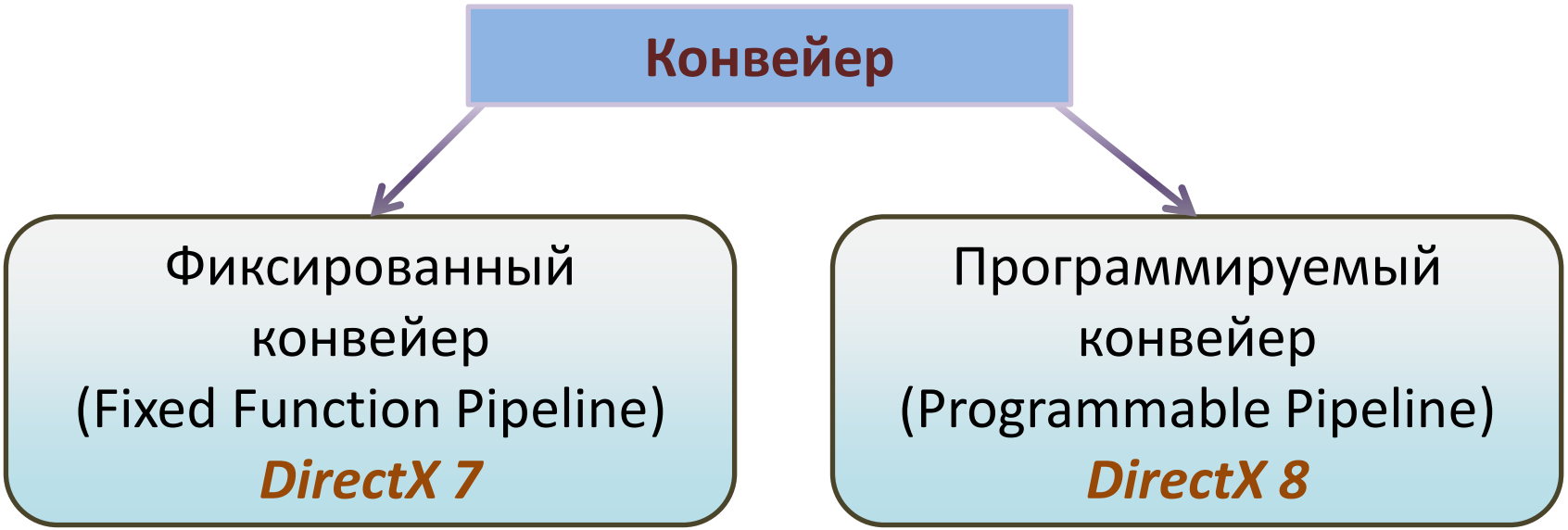
Текстурирование, освещение



Выполнение
Pixel Shader
<u>Пример:</u>
Bump-mapping

Понятие конвейера

- Изначально под конвейером понимался пиксельный процессор, который был подключён к своему блоку наложения текстур (TMU).
- После появления вершинных процессоров в DX 8 конвейеры стали разделять на пиксельные и вершинные.



Пиксельный шейдер

➢ Пиксельный шейдер — это программа, выполняемая пиксельным процессором для каждого фрагмента растеризованной геометрии.

➢ Фрагменты [пиксел**ы**] – пиксели изображения, которым поставлен в соответствие некоторый набор атрибутов, таких как цвет, глубина, текстурные координаты.

Функции пиксельного шейдера

- Задание модели расчета освещения отдельно взятой точки изображения (попиксельное освещение);
- Выборка из текстур;
- Преобразования цвета и значения глубины;
- Автоматическая генерация текстур;
- Преобразование текстур, мультитекстурирование;
- Постобработка кадра.

Вершинный шейдер

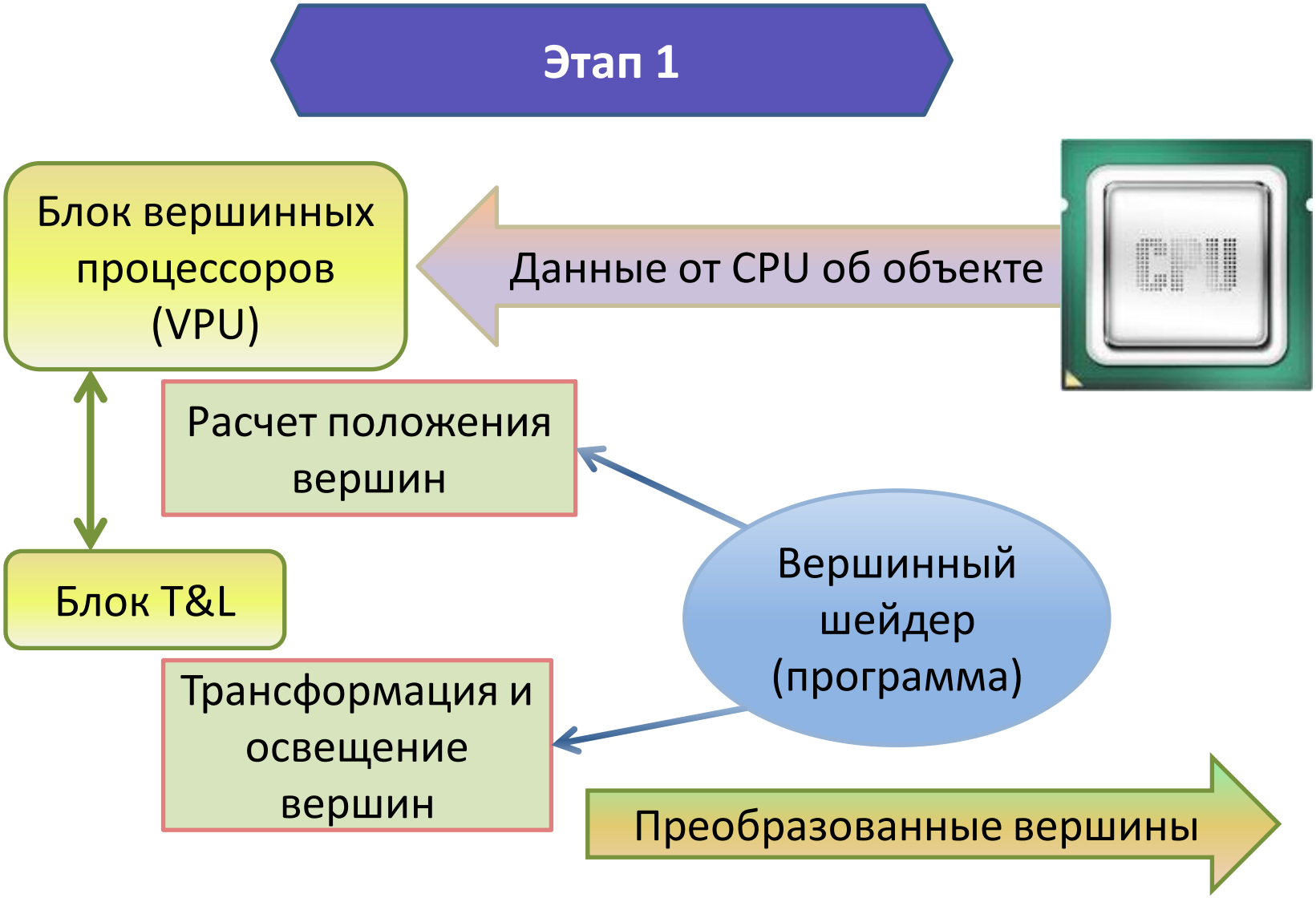
➢ Вершинный шейдер — это программа, выполняемая процессором видеокарты и заменяющая этапы преобразования и освещения в фиксированном конвейере.

- Стал стандартным элементом блока T&L в версии API DirectX 8.
- Предоставляет возможность выполнять программируемые алгоритмы по изменению параметров вершин и их освещению.
- Может эмулироваться программно.
- Пример использования: скиннинг (skinning) скелетной анимации персонажей; деформация и анимация объектов; имитация ткани и прочих гибких поверхностей.

Различия версий пиксельных и вершинных шейдеров

	DX 8 SM1.x	DX 9.0 SM2	DX 9.0c SM3	DX 10 SM4
Вершинные инструкции	128	256	512	64k
Пиксельные инструкции	4 ариф. + 8 текст.	32 ариф. + 64 текст.	512	
Вершинные константы	96	256	256	16x4096
Пиксельные константы	8	32	224	
Вершинные шаблоны	16	16	16	4096
Пиксельные шаблоны	2	12	32	
Вершинные входные данные	16	16	16	16
Пиксельные входные данные	4+2	8+2	10	32
Объекты визуализации (RTT)	1	4	4	8
Вершинные текстуры	нет	нет	4	128
Пиксельные текстуры	8	16	16	
Размер 2D текстур			2k x 2k	8k x 8k
Целочисленные и побитовые операции	-	-	-	+
Особый произвольный доступ	-	-	-	+
Производные	-	-	+	+
Вершинные условные переходы	нет	Статические	Статич/Динамич	Динамические
Пиксельные условные	нет	нет	Статич/Динамич	

Графический конвейер

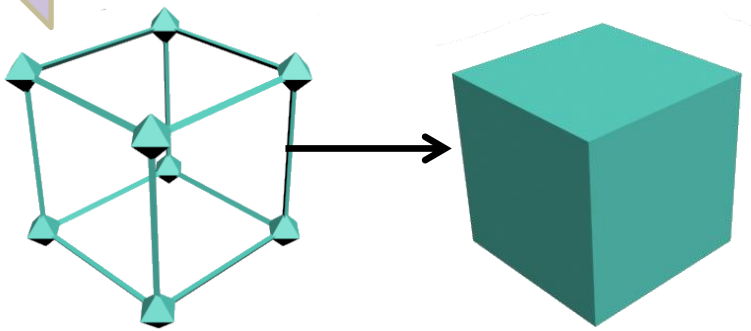


Графический конвейер

Этап 2

Сборка треугольников и растеризация (Primitive Assembly & Rasterization)

Трансформированные вершины

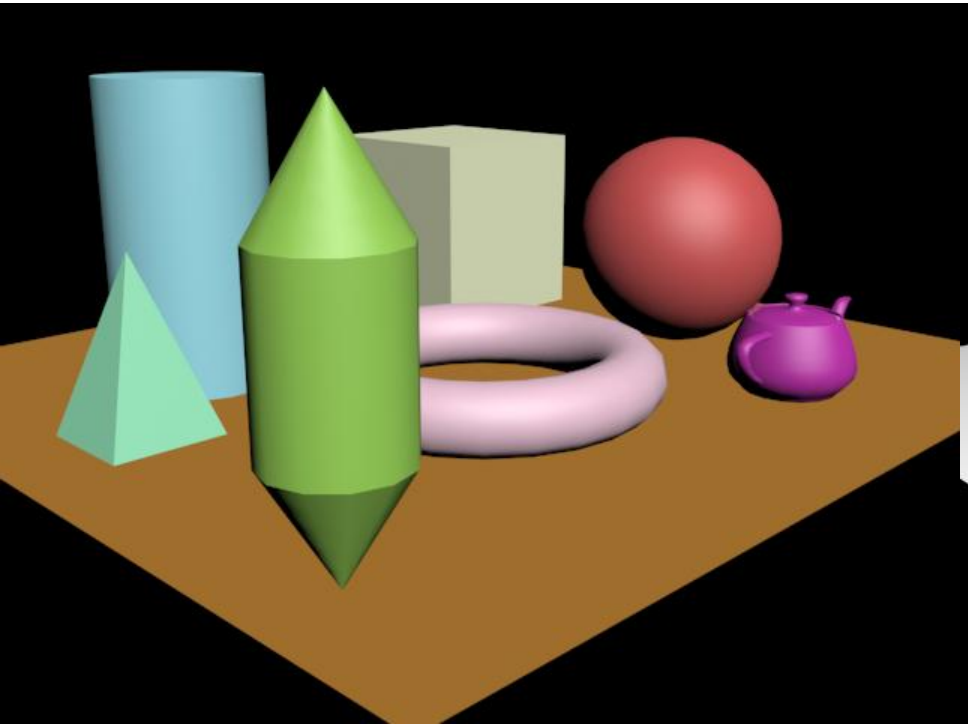


- Сборка 3D модели в полигоны
- Отсечение полигонов
- Определение фрагментов и положения пикселей примитива

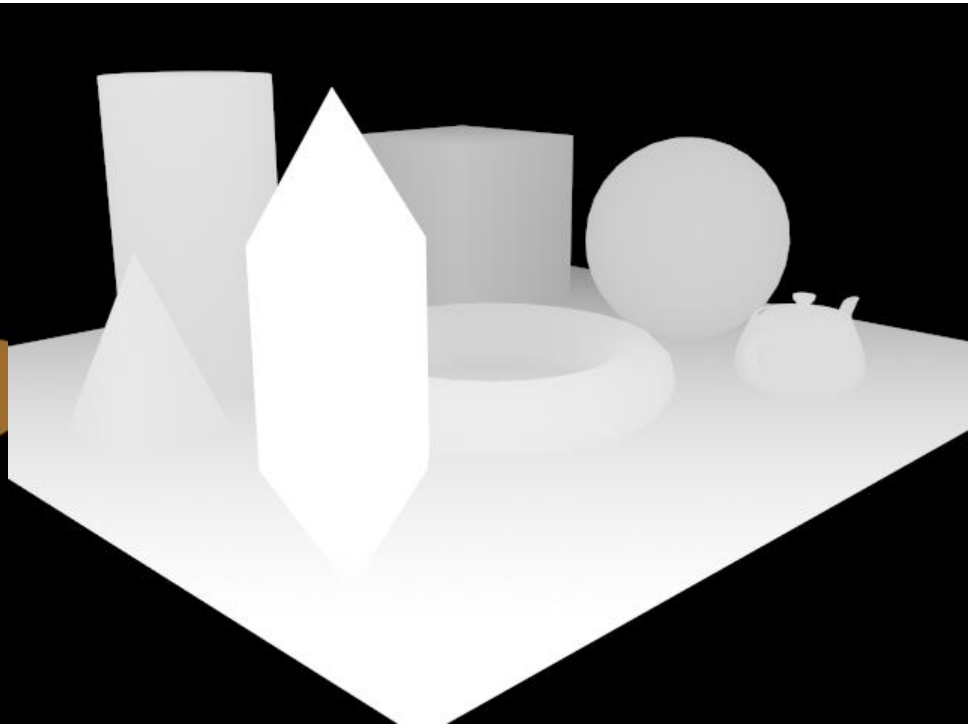
Позиция фрагмента в буфере

Z-буфер

- z-буфер - это отдельный буфер глубины, используемый для запоминания координаты Z (глубины) каждого видимого фрагмента (пиксела) в пространстве изображения.



Рендер трехмерной сцены



Рендер Z-буфера сцены отдельно

Алгоритм удаления невидимых граней

➢ Алгоритм z-буфера – один из простейших. Алгоритм решает задачу об удалении невидимых поверхностей, позволяя провести визуализацию области пересечения сложных геометрических объектов.

Преимущества алгоритма z-буфера

- Простота выполнения;
- Высокая эффективность при аппаратной реализации;
- Не требует предварительной сортировки по приоритету глубины – элементы заносятся в буфер в произвольном порядке;

Недостатки алгоритма z-буфера

- Высокое потребление памяти;
- Трудоемкость в реализации эффектов прозрачности (альфа-канал);
- Скорость работы Z-буфера серьёзно зависит от сортировки объектов.

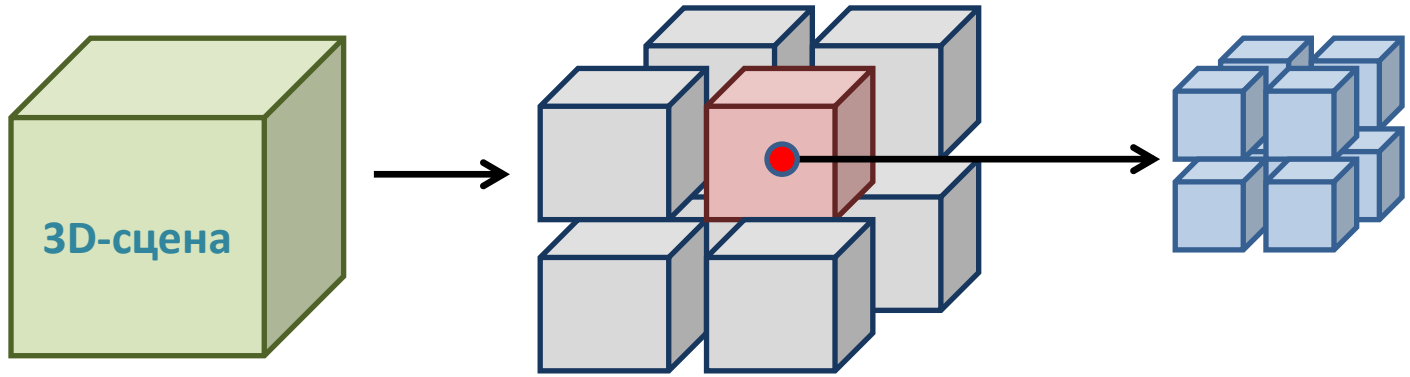
Алгоритм удаления невидимых граней

Принцип работы алгоритма Z-буфера:

1. Буфер кадра заполняется фоновым цветом;
2. В z-буфер помещается минимальное значение глубины;
3. Производится растеризация многоугольников (разбиение полигонов на фрагменты (пикселы));
4. Для каждого фрагмента (пиксела с координатами (x, y)) вычисляется глубина z(x, y);
5. Глубина z(x, y) сравнивается со значением, хранящимся в Z-буфере на этой же позиции;
6. Если z(x, y) > значения в Z-буфере, то в буфер кадра записываются атрибуты этого многоугольника (интенсивность, цвет и т. п.). z(x, y) записывается в Z-буфер;
7. Если z(x, y) < значения в Z-буфере, никаких действий не производится.

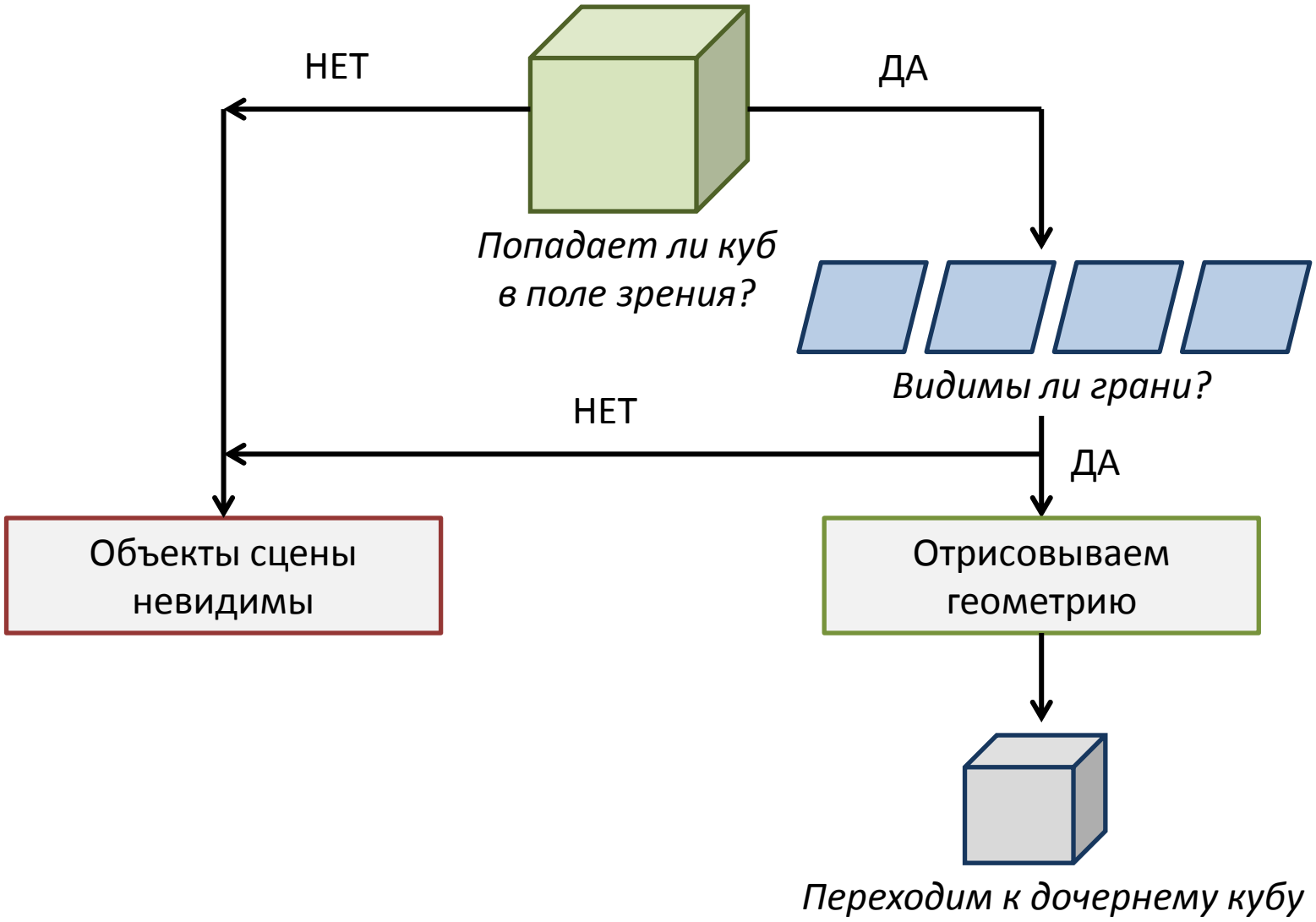
Иерархический Z-буфер

➢ Поскольку скорость работы Z-буфера зависит от сортировки объектов, для оптимизации расчетов используется алгоритм иерархического Z-буфера.

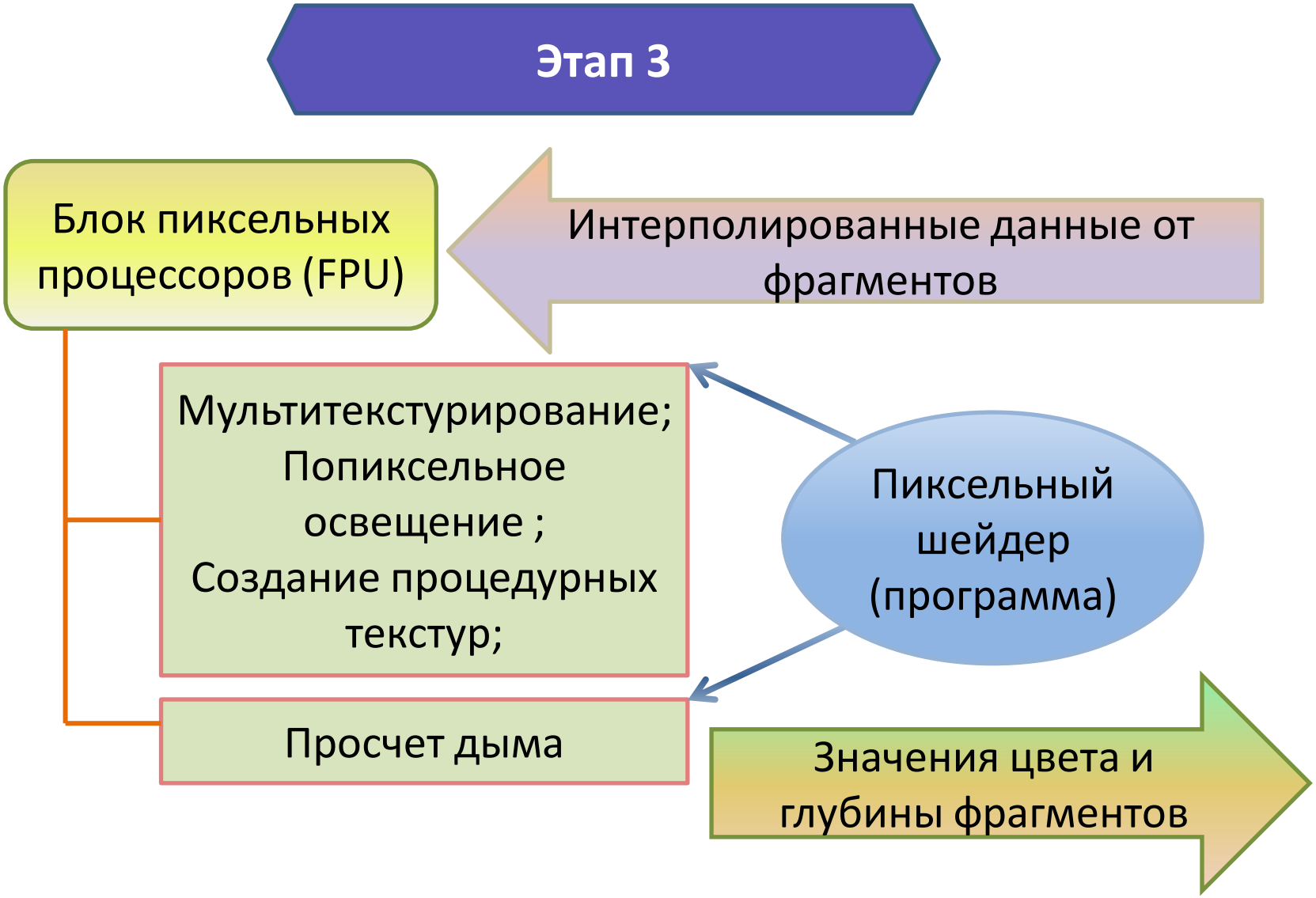


➢ Вокруг всей сцены описывается куб;
➢ Он иерархически разбивается на 8 меньших кубов;
➢ Каждый из 8 разбивается ещё на 8 частей, образуя древовидную структуру;
➢ Проверяем содержащиеся в данном кубе примитивы, и если их количество меньше определенного порогового значения, то разбиение прекращается.

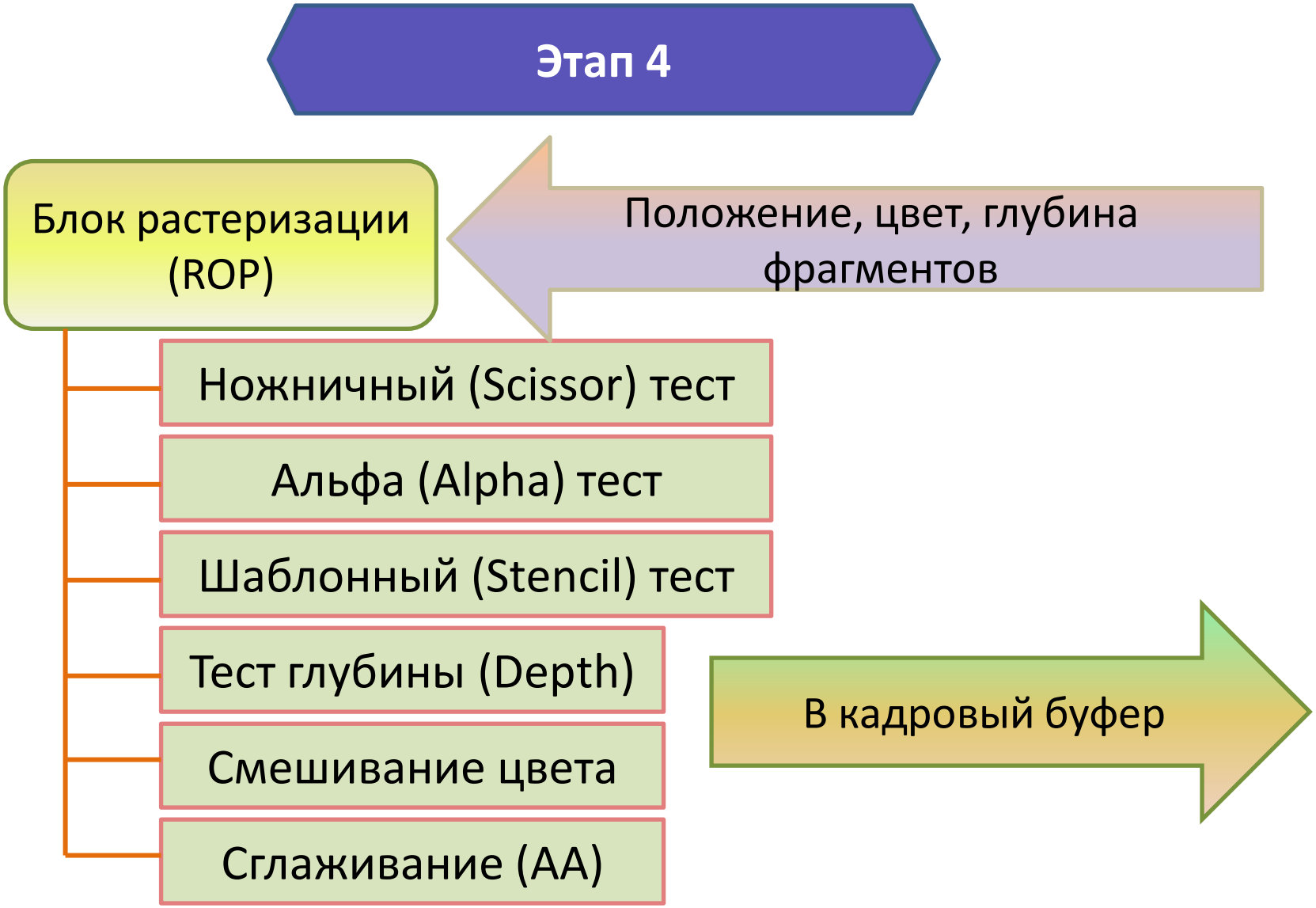
Иерархический Z-буфер



Графический конвейер



Графический конвейер



Графический конвейер

Этап 5

ЦАП ОЗУ(RAMDAC)

Кадровый буфер

Чтение кадрового буфера

Формирование сигнала

Вывод кадра на монитор