

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

**Лабораторная работа №4
по курсу “Компьютерная графика”**

Основы 3D графики

Выполнил: А. Н. Гуляев
Группа: М8О-311Б-23
Преподаватель: В. Д. Бахарев

Москва, 2026

Условие

В этой лабораторной работе вам предстоит реализовать технику наложения теней. Предстоит работа с рендерингом вне кадра “от лица” направленного источника света в текстуру глубины с использованием расширения Vulkan 1.2 Dynamic Rendering, а также работа с использованием данных о глубине сцены, чтобы создать эффект тени на поверхностях моделей.

Метод решения

В этой лабораторной работе реализована полная система shadow mapping с двумя рендер проходами: depth pass для создания карты глубин из позиции источника света и shadow pass для основного рендеринга с проверкой теней. Использованы отдельные VkImage/VkImageView для depth texture, shadow vertex shader для depth pass и основной fragment shader с расчётом теней.

Shadow Map (карта глубин)

Shadow map — это текстура глубины размером 8192×8192 пикселей в формате 32-битные значения глубины:

$$\text{ShadowMap} = \text{Texture}(8192 \times 8192, \text{DEPTH_32_BIT})$$

Каждый пиксель хранит расстояние от источника света до ближайшего объекта в сцене. Это позволяет позже проверить, находится ли рендерируемый пиксель в тени. Shadow map создаётся через отдельный рендер-проход, где вся сцена рендерится из точки зрения источника света (обычно точечный свет или солнце).

Двухпроходный рендеринг

Алгоритм состоит из двух этапов:

Этап 1: Depth Pass (создание карты глубин)

На первом проходе рендерится вся сцена из позиции источника света. Используется специальный vertex shader, который преобразует вершины в пространство источника света:

$$\vec{p}_{\text{light space}} = P_{\text{light}} \cdot V_{\text{light}} \cdot M \cdot \vec{p}_{\text{world}}$$

где:

- P_{light} — матрица проекции источника света (ортографическая для направленного света)
- V_{light} — матрица вида из источника света
- M — матрица модели объекта
- \vec{p}_{world} — исходная позиция вершины в мировом пространстве

В итоге только значения глубины z записываются в shadow map

Этап 2: Shadow Pass (основной рендеринг с тенями)

На втором проходе рендерится сцена в обычный framebuffer. Для каждого пикселя проверяется, находится ли он в тени:

1. Трансформируем пиксель в пространство света:

$$\vec{p}_{light} = P_{light} \cdot V_{light} \cdot M \cdot \vec{p}_{world}$$

2. Получаем UV координаты shadow map:

$$uv = \frac{\vec{p}_{light} \cdot xy + 1.0}{2.0}$$

3. Читаем глубину из shadow map:

$$z_{shadow} = \text{sample}(\text{shadowMap}, uv)$$

4. Сравниваем с текущей глубиной пикселя:

$$\text{inShadow} = (z_{light} > z_{shadow} + \text{bias})$$

где $\text{bias} \approx 0.0005$ — небольшое смещение для борьбы с артефактами shadow acne.

Вычисление освещения с учётом теней

Если пиксель в тени:

$$I_{final} = I_{ambient} \cdot C_{albedo}$$

Видна только рассеянное освещение без влияния направленного света и точечных источников.

Если пиксель на свету:

$$I_{final} = I_{ambient} + I_{directional} + I_{pointlights}$$

Применяется полная модель Blinn-Phong с диффузной и зеркальной составляющими.

Текстура глубины (Depth Texture)

Shadow map хранится как VkImage с форматом D32_SFLOAT:

- Каждый пиксель содержит одно 32-битное значение — глубину
- VkImageView интерпретирует это хранилище как texture для чтения в shader
- VkSampler настраивается специально для работы с depth texture

Матрицы преобразования света

Для направленного света (например, солнца):

$$P_{light} = \text{orthographic}(-50, 50, -50, 50, -100, 100)$$

Используется ортографическая проекция, так как лучи параллельны.

Матрица вида из света:

$$V_{light} = \text{lookAt}(\overrightarrow{light_pos}, \overrightarrow{light_target}, \overrightarrow{up})$$

Обычно источник света смотрит на центр сцены.

Анимация куба и динамические тени

Куб вращается вокруг Y-оси и движется по круговой траектории:

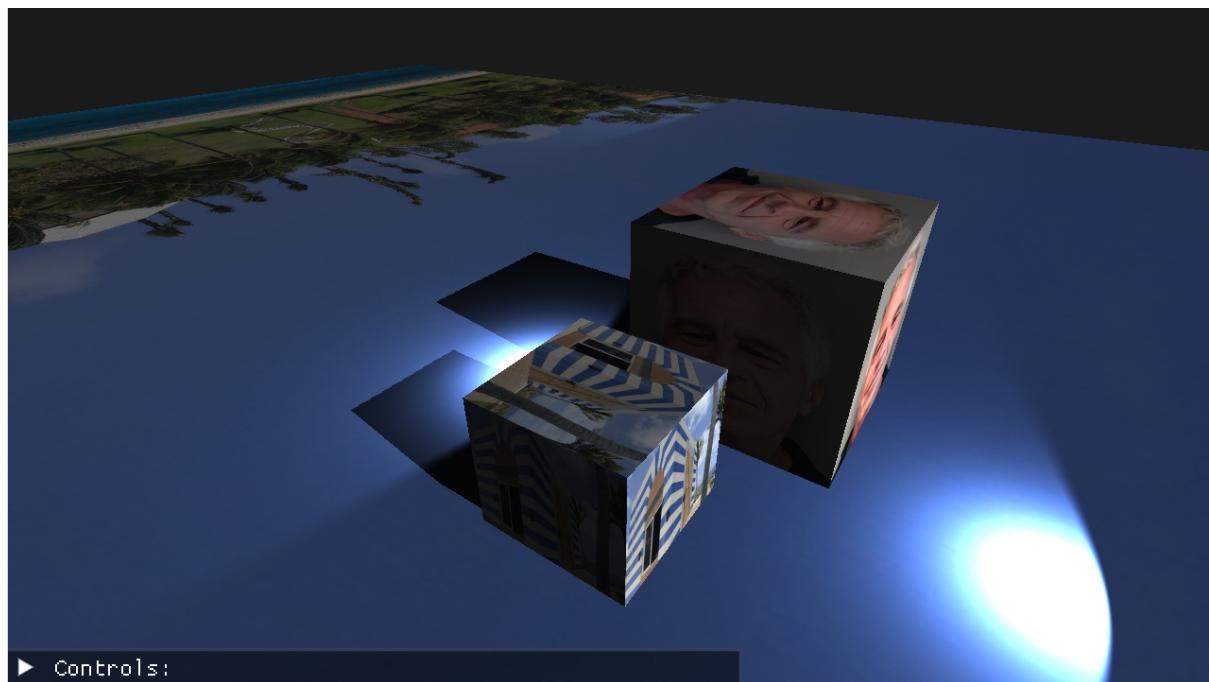
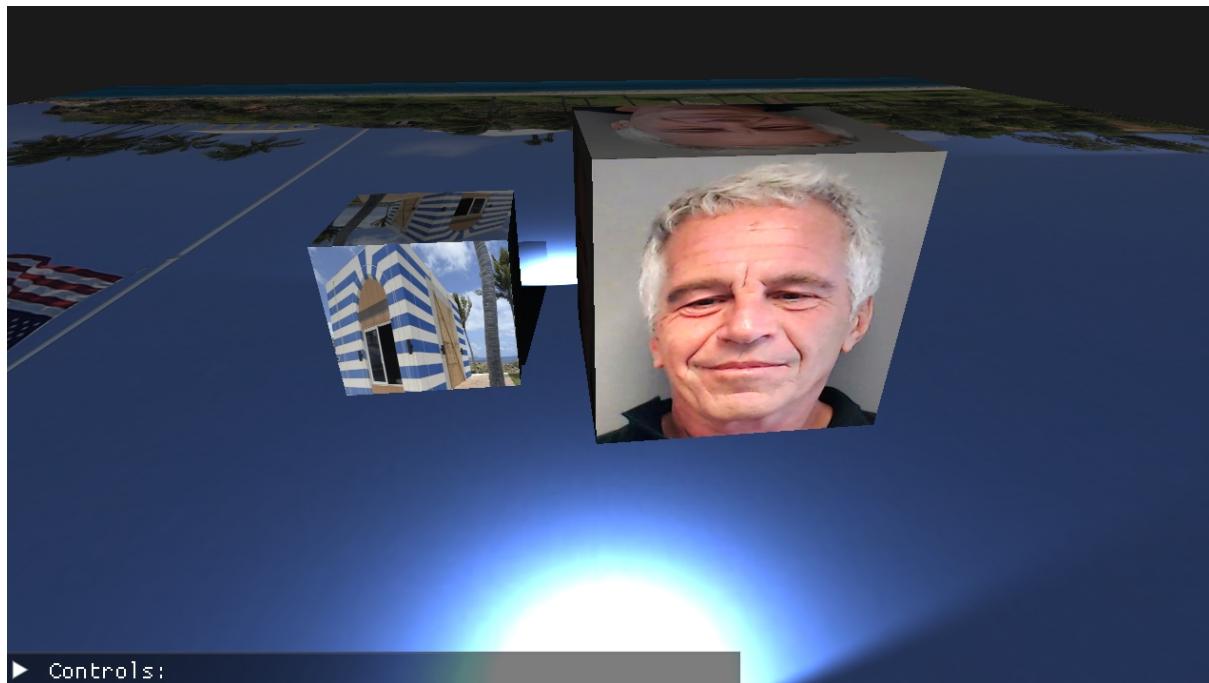
$$x(t) = r \cos(\omega t), z(t) = r \sin(\omega t), y(t) = \text{const}$$

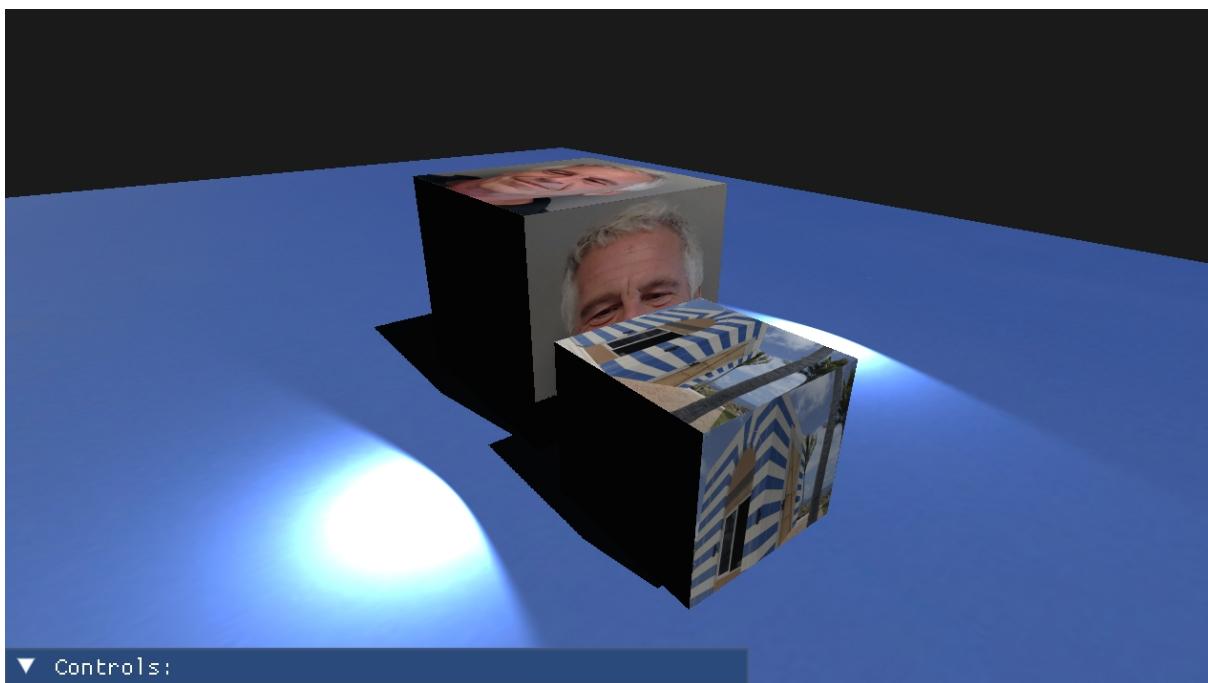
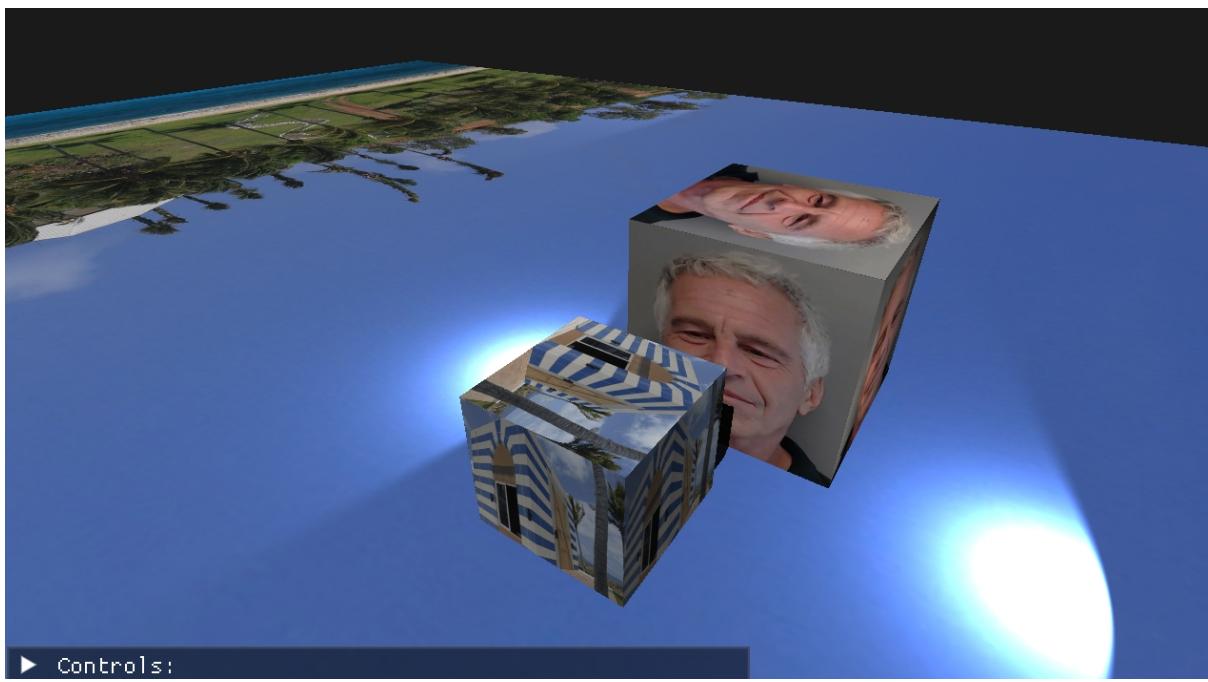
Тень куба вычисляется динамически в каждом пикселе:

1. Текущая позиция куба используется в матрице модели M
2. Тень рендерится с текущей ориентацией куба в depth pass
3. На shadow pass проверяется наличие тени в зависимости от текущей позиции
4. Результат: тень движется вместе с кубом по плоскости

Результаты

Сцена с статическими и динамической тенью:





Выводы

Я научился реализовывать shadow mapping — алгоритм создания динамических теней в Vulkan. Система состоит из двух проходов: в первом проходе сцена рендерится из позиции источника света в texture глубины (shadow map), во втором проходе при основном рендеринге для каждого пикселя проверяется его глубина в light space и сравнивается с

shadow map для определения, находится ли он в тени. Я освоил работу с depth texture (VkImage D32_SFLOAT), матрицами преобразования в пространство света, сравнением глубин с bias для борьбы с артефактами, и реализацию динамических теней, которые обновляются при движении объектов. Вращающийся куб корректно отбрасывает тень на плоскость в реальном времени.