

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: В. Н. Шишин  
Группа: М8О-310Б-23  
Преподаватель: В. Д. Бахарев

Москва, 2025

## 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.

- Подготовить матрицу проекции для направленного источника света
- Подготовить текстуру глубины, куда будет производится рендеринг сцены “от лица” направленного источника освещения
- Использовать расширение Vulkan Dynamic Rendering (`vkCmdBeginRendering`, `vkCmdEndRendering`) для рисования сцены в текстуру глубины
- Приготовить объект сэмплера с поддержкой сравнения значений (или реализовать PCF фильтрацию самому в фрагментном шейдере без поддержки сэмплером сравнения референсных значений)
- Использование текстуры и специального сэмплера в фрагментном шейдере для получения информации о глубине для последующего затемнения пикселей в модели Блинна-Фонга
- Используйте код предыдущей лабораторной работы, чтобы получилось показать в работе все изученные техники: освещение по модели Блинна-Фонга, наложение текстур и тени!

## Метод решения

Реализуется shadow mapping для направленного источника света: рендер сцены «от лица» источника в depth-texture (shadow map) с использованием Dynamic Rendering, затем в основном проходе используется полученная depth-текстура для сравнения глубин и затемнения фрагментов в расчёте Blinn-Phong.

Математика (ключевые формулы):

- Световое пространство (light-space): для каждой модели вычисляется light-space позиция вершин:

$\text{light\_pos} = L\_proj * L\_view * model * \text{vec4}(\text{position}, 1.0)$

где  $L\_view$  — видовая матрица «от лица» направленного света,  
 $L\_proj$  — проекция света (обычно ортографическая для *directional light*).

- Проекция в текстурные координаты shadow map:

$\text{ndc} = \text{light\_pos}.xyz / \text{light\_pos}.w$

$\text{tex} = 0.5 * \text{ndc} + 0.5$  // перевод из NDC в [0,1] для выборки текстуры

- Сравнение глубин (основная логика):

$\text{sampledDepth} = \text{texture}(\text{shadowSampler}, \text{tex}.xy).r$

$\text{currentDepthInLightSpace} = \text{tex}.z$

$\text{shadow} = (\text{currentDepthInLightSpace} - \text{bias} > \text{sampledDepth}) ? 1.0 : 0.0$

итоговый цвет =  $\text{ambient} + (1.0 - \text{shadow}) * (\text{diffuse} + \text{specular})$

- Depth bias и коррекция:  $\text{bias} = c0 + c1 * \max(0, 1 - (N \cdot L))$  —  
смещение глубины для уменьшения self-shadow acne.
- PCF (Percentage-Closer Filtering) — если нет аппаратной поддержки  
*compare sampler*, выполнять множественные выборки вокруг  $\text{tex}.xy$  и  
усреднять результаты:  
 $\text{shadow} = \text{average}(\text{texture}(\text{shadowMap}, \text{tex}.xy + \text{offset}_i).r < \text{tex}.z - \text{bias} ? 1 : 0)$

#### Проекция для направленного источника:

- Для *directional light* обычно используется ортографическая проекция:  
 $L\_proj = \text{orthographic}(\text{left}, \text{right}, \text{bottom}, \text{top}, \text{near}, \text{far})$ . Параметры  
подбираются так, чтобы покрыть область сцены, освещаемую  
солнцем (или можно вычислять AABV в пространстве света).
- Итоговая матрица для шейдера:  $\text{shadow\_matrix} = L\_proj * L\_view$

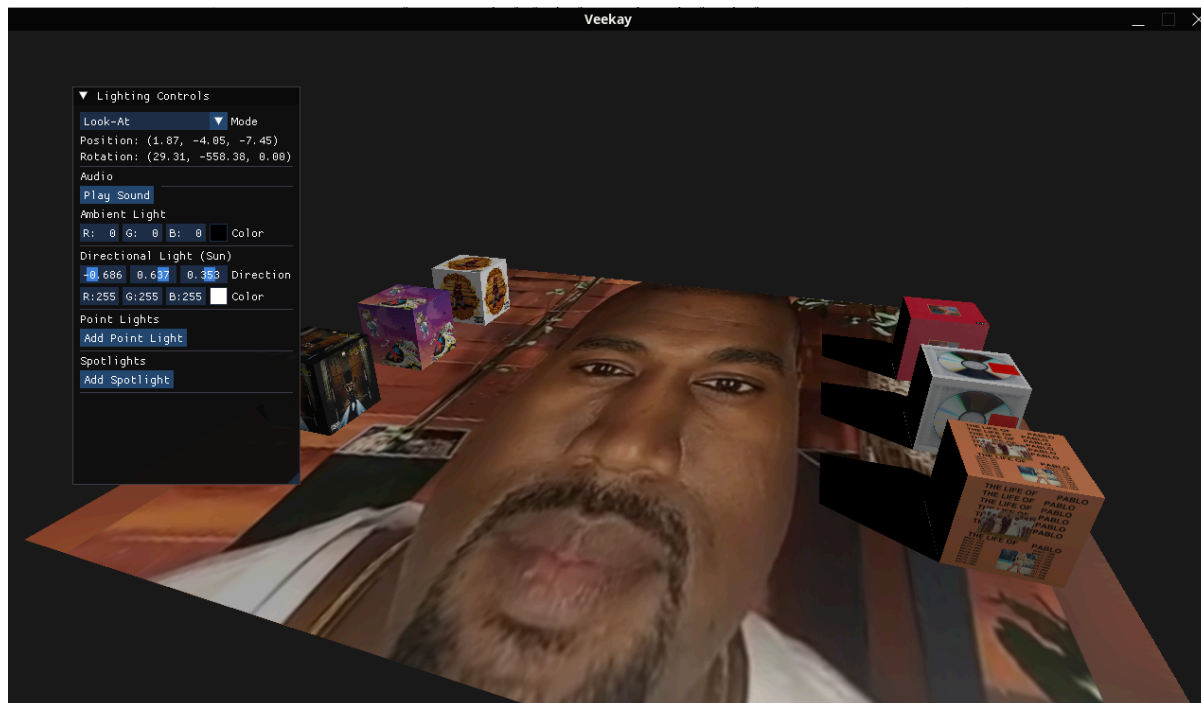
#### Основные шаги реализации:

1. Подготовить depth image (shadow map):
  - Создать *VkImage* с форматом глубины (например, *VK\_FORMAT\_D32\_SFLOAT*), выделить память, создать *VkImageView*.
  - Размер shadow map (например, 2048 или 4096) выбирается в зависимости от качества.
2. Рендер в depth texture (shadow pass) с Dynamic Rendering:
  - Создать depth-only pipeline (вершинный шейдер для записи *gl\_Position* по модели; фрагментный не нужен).

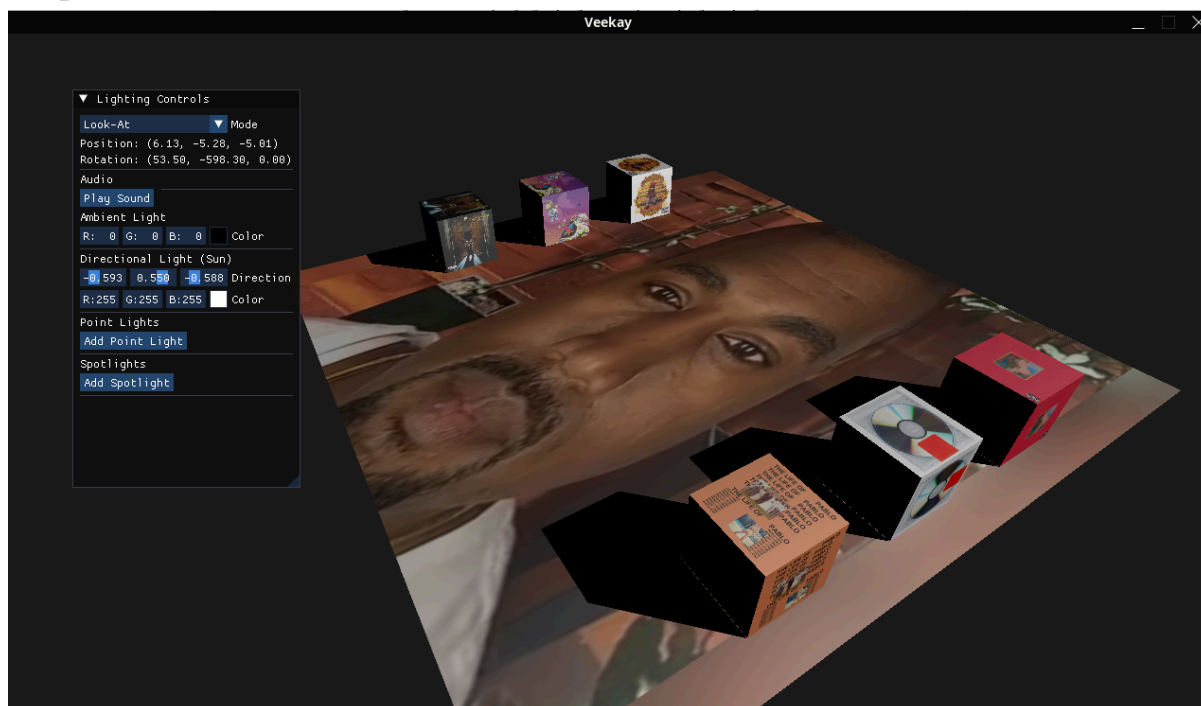
- Вызовы: `vkCmdBeginRenderingKHR(...)` с соответствующим `VkRenderingAttachmentInfo` для depth attachment → отрисовать сцены → `vkCmdEndRenderingKHR(...)`.
  - При записи глубины включить depth bias (`vkCmdSetDepthBias`) и, при необходимости, изменить cull mode (например, `VK_CULL_MODE_FRONT_BIT`) чтобы уменьшить артефакты.
3. Создать и настроить sampler для shadow map:
- Либо создать comparison sampler (`VkSampler` с `compareEnable = VK_TRUE` и `compareOp = VK_COMPARE_OP_LESS_OR_EQUAL`) и в шейдере использовать `sampler2DShadow` (hardware PCF доступен на некоторых устройствах);
  - Либо обычный sampler и реализовать PCF вручную в фрагментном шейдере (несколько выборов и усреднение).
4. Передать `shadow_matrix` и shadow map в основной проход:
- Добавить `mat4 shadow_matrix` или `mat4 light_space` в UBO/scene uniforms (или в отдельный uniform).
  - Обновить `VkDescriptorSet` (биндинг `VK_DESCRIPTOR_TYPE_COMBINED_IMAGE_SAMPLER`) с `VkDescriptorImageInfo` для shadow map + sampler.
5. В основном фрагментном шейдере:
- Преобразовать мировую позицию фрагмента в light-space:  $\text{proj} = \text{shadow\_matrix} * \text{vec4}(\text{worldPos}, 1)$
  - Получить  $\text{tex} = \text{proj.xy} / \text{proj.w}$  и  $\text{depth} = \text{proj.z} / \text{proj.w} \rightarrow \text{tex} = 0.5 * \text{tex} + 0.5$
  - Считать `sampledDepth` и выполнить сравнение с bias/PCF → получить коэффициент  $\text{shadow} \in [0, 1]$
  - Умножить diffuse/specular компоненты Blinn-Phong на  $(1.0 - \text{shadow})$
6. Синхронизация и переходы layout:
- Перед рендером в shadow map обеспечить layout `VK_IMAGE_LAYOUT_DEPTH_STENCIL_ATTACHMENT_OPTIMAL` для записи; перед чтением в основном проходе перейти в `VK_IMAGE_LAYOUT_SHADER_READ_ONLY_OPTIMAL`.
  - Используются image memory barriers (`vkCmdPipelineBarrier`) для корректной синхронизации между проходами.

## Результаты

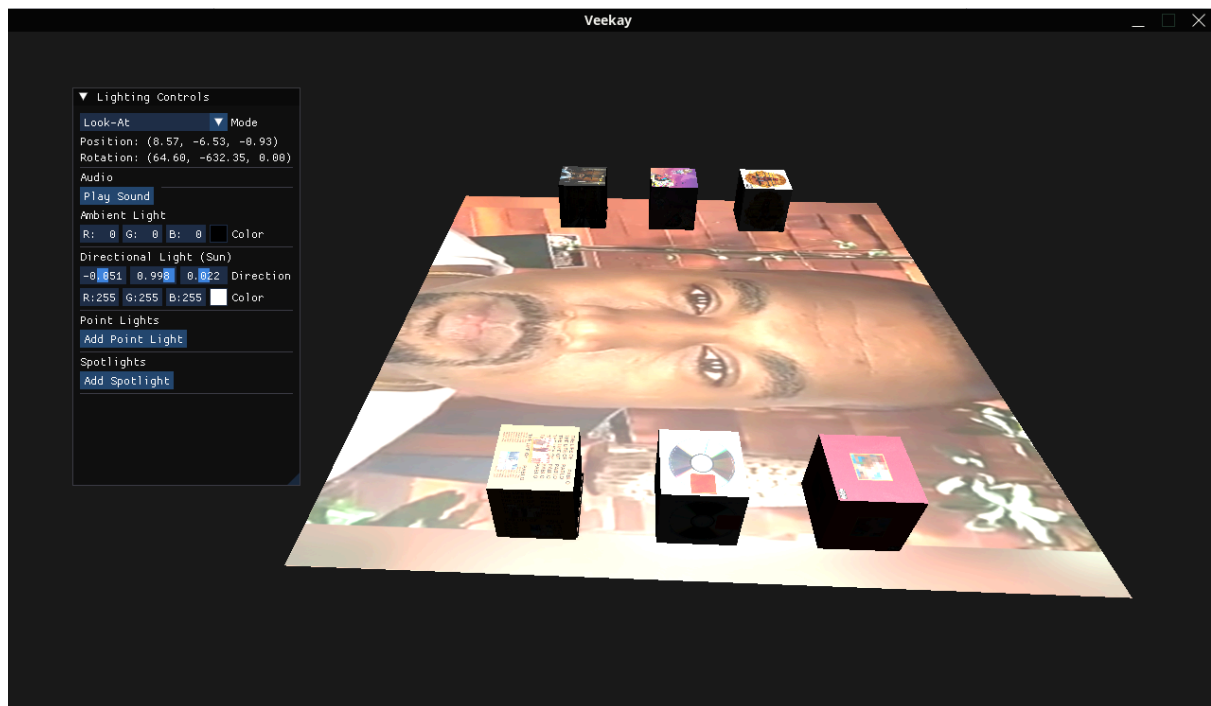
Тени от направленного света (солнца)



Тени от направленного света (солнца) под другим углом и положением направленного света



## Тени при направленном свете под прямым углом



## Выводы

Я научился(-ась) формировать матрицу проекции и вид из точки зрения направленного источника, рендерить сцену в depth-текстуру с использованием Dynamic Rendering, настраивать depth-image и comparison-sampler (или реализовывать PCF вручную), корректно сравнивать глубины в шейдере с учётом bias и применять результат для затемнения фрагментов в модели Блинн-Фонга; а также управлять этими ресурсами через UBO/descriptor sets и интегрировать тени с остальными техниками (освещение, текстурирование) в одном приложении.