

Компьютерная графика

Практика 9: Shadow mapping 2

2021

Задание 1

Подбираем shadow bias

- ▶ Нужно прибавить константу (bias) к значению, прочитанному из shadow map

Задание 1

Подбираем shadow bias

- ▶ Нужно прибавить константу (bias) к значению, прочитанному из shadow map
- ▶ Слишком большое значение приведёт к заметному peter panning (тень будет съезжать в сторону от модели)

Задание 1

Подбираем shadow bias

- ▶ Нужно прибавить константу (bias) к значению, прочитанному из shadow map
- ▶ Слишком большое значение приведёт к заметному peter panning (тень будет съезжать в сторону от модели)
- ▶ Слишком маленького значения будет недостаточно, чтобы убрать shadow acne

Задание 2

Вычисляем хорошую матрицу проекции для shadow map

- ▶ Нужно найти центр видимой области и оси X, Y, Z

Задание 2

Вычисляем хорошую матрицу проекции для shadow map

- ▶ Нужно найти центр видимой области и оси X, Y, Z
- ▶ Направления осей $\hat{X}, \hat{Y}, \hat{Z}$ уже посчитаны, нужно только вычислить их длину

Задание 2

Вычисляем хорошую матрицу проекции для shadow map

- ▶ Нужно найти центр видимой области и оси X, Y, Z
- ▶ Направления осей $\hat{X}, \hat{Y}, \hat{Z}$ уже посчитаны, нужно только вычислить их длину
- ▶ С помощью функции `bbox` можно посчитать bounding box сцены, её центр C - центр видимой области

Задание 2

Вычисляем хорошую матрицу проекции для shadow map

- ▶ Нужно найти центр видимой области и оси X, Y, Z
- ▶ Направления осей $\hat{X}, \hat{Y}, \hat{Z}$ уже посчитаны, нужно только вычислить их длину
- ▶ С помощью функции `bbox` можно посчитать bounding box сцены, её центр C - центр видимой области
- ▶ Пройдясь по всем 8 вершинам V bounding box'a сцены, можно посчитать скалярное произведение $(V - C) \cdot \hat{X}$, максимум модуля таких произведений - длина вектора X

Задание 2

Вычисляем хорошую матрицу проекции для shadow map

- ▶ Нужно найти центр видимой области и оси X, Y, Z
- ▶ Направления осей $\hat{X}, \hat{Y}, \hat{Z}$ уже посчитаны, нужно только вычислить их длину
- ▶ С помощью функции `bbox` можно посчитать bounding box сцены, её центр C - центр видимой области
- ▶ Пройдясь по всем 8 вершинам V bounding box'a сцены, можно посчитать скалярное произведение $(V - C) \cdot \hat{X}$, максимум модуля таких произведений - длина вектора X
- ▶ Аналогично для Y и Z

Задание 2

Вычисляем хорошую матрицу проекции для shadow map

- ▶ Нужно найти центр видимой области и оси X, Y, Z
- ▶ Направления осей $\hat{X}, \hat{Y}, \hat{Z}$ уже посчитаны, нужно только вычислить их длину
- ▶ С помощью функции `bbox` можно посчитать bounding box сцены, её центр C - центр видимой области
- ▶ Пройдясь по всем 8 вершинам V bounding box'a сцены, можно посчитать скалярное произведение $(V - C) \cdot \hat{X}$, максимум модуля таких произведений - длина вектора X
- ▶ Аналогично для Y и Z
- ▶ Используя X, Y, Z, C можно построить матрицу ортографической проекции

Задание 3

Shadow sampler + PCF

- ▶ Меняем min/mag фильтры shadow map на GL_LINEAR

Задание 3

Shadow sampler + PCF

- ▶ Меняем min/mag фильтры shadow map на GL_LINEAR
- ▶ Устанавливаем текстуре shadow map свойства
GL_TEXTURE_COMPARE_MODE = GL_COMPARE_REF_TO_TEXTURE
и GL_TEXTURE_COMPARE_FUNC, GL_LESS

Задание 3

Shadow sampler + PCF

- ▶ Меняем min/mag фильтры shadow map на `GL_LINEAR`
- ▶ Устанавливаем текстуре shadow map свойства
`GL_TEXTURE_COMPARE_MODE = GL_COMPARE_REF_TO_TEXTURE`
и `GL_TEXTURE_COMPARE_FUNC, GL_LESS`
- ▶ Меняем тип sampler'а в шейдере на `sampler2DShadow`

Задание 3

Shadow sampler + PCF

- ▶ Меняем min/mag фильтры shadow map на `GL_LINEAR`
- ▶ Устанавливаем текстуре shadow map свойства `GL_TEXTURE_COMPARE_MODE = GL_COMPARE_REF_TO_TEXTURE` и `GL_TEXTURE_COMPARE_FUNC, GL_LESS`
- ▶ Меняем тип sampler'а в шейдере на `sampler2DShadow`
- ▶ Меняем обращение к текстуре:
`texture(..., shadow_pos.xyz + bias)` (возвращает float, а не vec4!)

Задание 4

Variance shadow maps

- ▶ Убираем свойства `GL_TEXTURE_COMPARE_MODE` и `GL_TEXTURE_COMPARE_FUNC`

Задание 4

Variance shadow maps

- ▶ Убираем свойства `GL_TEXTURE_COMPARE_MODE` и `GL_TEXTURE_COMPARE_FUNC`
- ▶ Устанавливаем текстуру shadow map тип данных `GL_RG32F` (format и type не принципиальны, можно `GL_RGBA` и `GL_FLOAT`)

Задание 4

Variance shadow maps

- ▶ Убираем свойства `GL_TEXTURE_COMPARE_MODE` и `GL_TEXTURE_COMPARE_FUNC`
- ▶ Устанавливаем текстуре shadow map тип данных `GL_RG32F` (format и type не принципиальны, можно `GL_RGBA` и `GL_FLOAT`)
- ▶ Добавляем shadow map к фреймбуферу как `GL_COLOR_ATTACHMENT0` (вместо `GL_DEPTH_ATTACHMENT`)

Задание 4

Variance shadow maps

- ▶ Убираем свойства `GL_TEXTURE_COMPARE_MODE` и `GL_TEXTURE_COMPARE_FUNC`
- ▶ Устанавливаем текстуре shadow map тип данных `GL_RG32F` (`format` и `type` не принципиальны, можно `GL_RGBA` и `GL_FLOAT`)
- ▶ Добавляем shadow map к фреймбуферу как `GL_COLOR_ATTACHMENT0` (вместо `GL_DEPTH_ATTACHMENT`)
- ▶ Создаём текстуру или `renderbuffer`, которые будут использоваться для глубины, и добавляем его как `GL_DEPTH_ATTACHMENT` фреймбуфера

Задание 4

Variance shadow maps

- ▶ Убираем свойства `GL_TEXTURE_COMPARE_MODE` и `GL_TEXTURE_COMPARE_FUNC`
- ▶ Устанавливаем текстуре shadow map тип данных `GL_RG32F` (format и type не принципиальны, можно `GL_RGBA` и `GL_FLOAT`)
- ▶ Добавляем shadow map к фреймбуферу как `GL_COLOR_ATTACHMENT0` (вместо `GL_DEPTH_ATTACHMENT`)
- ▶ Создаём текстуру или renderbuffer, которые будут использоваться для глубины, и добавляем его как `GL_DEPTH_ATTACHMENT` фреймбуфера
- ▶ Во фрагментном шейдере, рисуящем shadow map, добавляем out-переменную и пишем в неё `vec4(z, z * z, 0.0, 0.0)` (z можно достать из `gl_FragCoord`)

Задание 4

Variance shadow maps

- ▶ Убираем свойства `GL_TEXTURE_COMPARE_MODE` и `GL_TEXTURE_COMPARE_FUNC`
- ▶ Устанавливаем текстуру shadow map тип данных `GL_RG32F` (format и type не принципиальны, можно `GL_RGBA` и `GL_FLOAT`)
- ▶ Добавляем shadow map к фреймбуферу как `GL_COLOR_ATTACHMENT0` (вместо `GL_DEPTH_ATTACHMENT`)
- ▶ Создаём текстуру или renderbuffer, которые будут использоваться для глубины, и добавляем его как `GL_DEPTH_ATTACHMENT` фреймбуфера
- ▶ Во фрагментном шейдере, рисуя shadow map, добавляем out-переменную и пишем в неё `vec4(z, z * z, 0.0, 0.0)` (z можно достать из `gl_FragCoord`)
- ▶ В основном фрагментном шейдере меняем тип sampler'a обратно на `sampler2D`, читаем из него и используем неравенство Чебышёва для вычисления освещённости:

```
vec2 data = texture(shadow_map, shadow_pos.xy).rg;
float mu = data.r;
float sigma = data.g - mu * mu;
float z = shadow_pos.z;
float factor = (z < mu) ? 1.0
    : sigma / (sigma + (z - mu) * (z - mu));
```

Задание 5

Исправляем артефакты

- ▶ Добавляем наклон поверхности к среднему значению квадрата глубины:
 - ▶ Через $dFdx$ и $dFdy$ можно получить градиент глубины по X и Y
 - ▶ К квадрату глубины добавляем $\frac{1}{4} \left[\left(\frac{\partial Z}{\partial X} \right)^2 + \left(\frac{\partial Z}{\partial Y} \right)^2 \right]$

Задание 5

Исправляем артефакты

- ▶ Добавляем наклон поверхности к среднему значению квадрата глубины:
 - ▶ Через $dFdx$ и $dFdy$ можно получить градиент глубины по X и Y
 - ▶ К квадрату глубины добавляем $\frac{1}{4} \left[\left(\frac{\partial Z}{\partial X} \right)^2 + \left(\frac{\partial Z}{\partial Y} \right)^2 \right]$
- ▶ Добавляем shadow bias (вычитаем константу из значения Z , использующегося для вычисления освещённости)

Задание 5

Исправляем артефакты

- ▶ Добавляем наклон поверхности к среднему значению квадрата глубины:
 - ▶ Через $dFdx$ и $dFdy$ можно получить градиент глубины по X и Y
 - ▶ К квадрату глубины добавляем $\frac{1}{4} \left[\left(\frac{\partial Z}{\partial X} \right)^2 + \left(\frac{\partial Z}{\partial Y} \right)^2 \right]$
- ▶ Добавляем shadow bias (вычитаем константу из значения Z , использующегося для вычисления освещённости)
- ▶ Значение, получающееся из формулы неравенства Чебышёва, преобразуем: диапазон $[0, \delta]$ переходит в 0 , а диапазон $[\delta, 1]$ переходит в $[0, 1]$ (δ - некое фиксированное значение, например, 0.125)

Задание 6

Размываем shadow map

- ▶ Вместо чтения одного пикселя из shadow map, читаем набор значений из соседних пикселей (аналогично тому, как делалось размытие в задании №5 практики №7) и усредняем по Гауссу

Задание 6

Размываем shadow map

- ▶ Вместо чтения одного пикселя из shadow map, читаем набор значений из соседних пикселей (аналогично тому, как делалось размытие в задании №5 практики №7) и усредняем по Гауссу
- ▶ Полученный двумерный вектор используем для вычисления освещённости через неравенство Чебышёва

Задание 6

Размываем shadow map

- ▶ Вместо чтения одного пикселя из shadow map, читаем набор значений из соседних пикселей (аналогично тому, как делалось размытие в задании №5 практики №7) и усредняем по Гауссу
- ▶ Полученный двумерный вектор используем для вычисления освещённости через неравенство Чебышёва
- ▶ **N.B.:** по-хорошему это размытие нужно делать отдельными проходами с отдельными шейдерами и отдельным размытием по X и Y (см. *separable Gaussian blur*)