

Компьютерная графика

Домашнее задание 3: Визуализатор сцен v2

2021

Задание

- Сделать визуализатор сцен (в любом формате) с текстурами и освещением по Фонгу, normal mapping'ом, material mapping'ом, tone mapping'ом, гамма-коррекцией, отражениями, туманом и объёмными тенями

Задание

- Сделать визуализатор сцен (в любом формате) с текстурами и освещением по Фонгу, normal mapping'ом, material mapping'ом, tone mapping'ом, гамма-коррекцией, отражениями, туманом и объёмными тенями
- Удобнее иметь два файла: один – с большой сценой (напр. Sponza), второй – с маленьким отражающим объектом (напр. Stanford bunny)

Задание

- Сделать визуализатор сцен (в любом формате) с текстурами и освещением по Фонгу, normal mapping'ом, material mapping'ом, tone mapping'ом, гамма-коррекцией, отражениями, туманом и объёмными тенями
- Удобнее иметь два файла: один – с большой сценой (напр. Sponza), второй – с маленьkim отражающим объектом (напр. Stanford bunny)
- Камера должна управляться пользователем (любым способом, главное – чтобы можно было всё разглядеть)

Задание

- Сделать визуализатор сцен (в любом формате) с текстурами и освещением по Фонгу, normal mapping'ом, material mapping'ом, tone mapping'ом, гамма-коррекцией, отражениями, туманом и объёмными тенями
- Удобнее иметь два файла: один – с большой сценой (напр. Sponza), второй – с маленьким отражающим объектом (напр. Stanford bunny)
- Камера должна управляться пользователем (любым способом, главное – чтобы можно было всё разглядеть)
- Тени нужны только от направленного источника (солнца)

Задание

- Сделать визуализатор сцен (в любом формате) с текстурами и освещением по Фонгу, normal mapping'ом, material mapping'ом, tone mapping'ом, гамма-коррекцией, отражениями, туманом и объёмными тенями
- Удобнее иметь два файла: один – с большой сценой (напр. Sponza), второй – с маленьким отражающим объектом (напр. Stanford bunny)
- Камера должна управляться пользователем (любым способом, главное – чтобы можно было всё разглядеть)
- Тени нужны только от направленного источника (солнца)
- Источники света – какие хотите, главное, чтобы были видны все эффекты; один из источников должен иметь яркость **больше единицы**

Задание

- Сделать визуализатор сцен (в любом формате) с текстурами и освещением по Фонгу, normal mapping'ом, material mapping'ом, tone mapping'ом, гамма-коррекцией, отражениями, туманом и объёмными тенями
- Удобнее иметь два файла: один – с большой сценой (напр. Sponza), второй – с маленьким отражающим объектом (напр. Stanford bunny)
- Камера должна управляться пользователем (любым способом, главное – чтобы можно было всё разглядеть)
- Тени нужны только от направленного источника (солнца)
- Источники света – какие хотите, главное, чтобы были видны все эффекты; один из источников должен иметь яркость **больше единицы**
- Можно переиспользовать код из 2ого домашнего задания

Задание



Normal mapping

- Если в вашей модели есть normal maps, вам нужны tangent/bitangent векторы, которые не поддерживаются форматом Wavefront OBJ (но поддерживаются в glTF)

Normal mapping

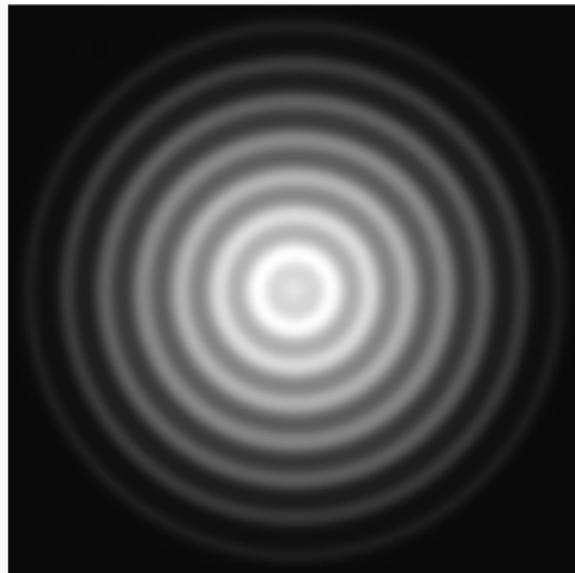
- Если в вашей модели есть normal maps, вам нужны tangent/bitangent векторы, которые не поддерживаются форматом Wavefront OBJ (но поддерживаются в glTF)
- Если в вашей модели есть bump maps, то вам не нужны дополнительные данные в вершинах, но сложнее алгоритм вычисления итоговой нормали

Normal mapping

- Если в вашей модели есть normal maps, вам нужны tangent/bitangent векторы, которые не поддерживаются форматом Wavefront OBJ (но поддерживаются в glTF)
- Если в вашей модели есть bump maps, то вам не нужны дополнительные данные в вершинах, но сложнее алгоритм вычисления итоговой нормали
- Второй способ описан, например, в статье [Morten S. Mikkelsen - Bump Mapping Unparametrized Surfaces on the GPU](#), за него будет больше баллов

Bump map vs normal map

Bump map



Normal map



Эффект normal mapping



Material mapping

- Скорее всего, у вас будут обычные gloss maps (`map_Ks` в OBJ)

Material mapping

- Скорее всего, у вас будут обычные gloss maps (`map_Ks` в OBJ)
- Их нужно применять как множитель для specular составляющей освещения

Tone mapping

- Возьмите любую функцию, которая вам понравится

Tone mapping

- Возьмите любую функцию, которая вам понравится
- Лучше ACES или Uncharted

Tone mapping

- Возьмите любую функцию, которая вам понравится
- Лучше ACES или Uncharted
- Применять *перед* гамма-коррекцией

Tone mapping

- Возьмите любую функцию, которая вам понравится
- Лучше ACES или Uncharted
- Применять *перед* гамма-коррекцией
- Текстуры альбено из входных данных нужно загружать как **sRGB-текстуры**, иначе они будут выглядеть слишком ярко из-за гамма-коррекции

Отражения

- Возьмите один небольшой движущийся по сцене объект

Отражения

- Возьмите один небольшой движущийся по сцене объект
- Страйте cube-map-отражения из центра этого объекта

Отражения

- Возьмите один небольшой движущийся по сцене объект
- Страйте cube-map-отражения из центра этого объекта
- Применяйте отражения только к этому объекту

Отражения

- Возьмите один небольшой движущийся по сцене объект
- Страйте cube-map-отражения из центра этого объекта
- Применяйте отражения только к этому объекту
- Не страшно, если в отражениях не будет виден туман, но тени должны быть видны

Отражения



Отражения

- Текстуру отражений можно сделать довольно маленькой, например 256x256

Отражения

- Текстуру отражений можно сделать довольно маленькой, например 256x256
- В текстуру отражений нужно записывать цвет **без tone-mapping'a и гамма-коррекции**, т.е. сырое количество света

Отражения

- Текстуру отражений можно сделать довольно маленькой, например 256x256
- В текстуру отражений нужно записывать цвет **без tone-mapping'a и гамма-коррекции**, т.е. сырое количество света
- Это значение может быть больше единицы \Rightarrow для отражений нужно или использовать floating-point текстуру, или записывать в неё цвет делённый на некую максимальную яркость (и не забыть домножить обратно при чтении из этой текстуры)

Отражения

- Текстуру отражений можно сделать довольно маленькой, например 256x256
- В текстуру отражений нужно записывать цвет **без tone-mapping'a и гамма-коррекции**, т.е. сырое количество света
- Это значение может быть больше единицы \Rightarrow для отражений нужно или использовать floating-point текстуру, или записывать в неё цвет делённый на некую максимальную яркость (и не забыть домножить обратно при чтении из этой текстуры)
- Если использовать второй способ, может возникнуть banding из-за ограниченной точности текстуры – тогда лучше сделать её sRGB, или вручную конвертировать при записи и чтении

Banding



Туман и объёмные тени

- Обычный туман можно сделать, вычислив оптическую глубину и подставив в формулу вида

```
mix(ambient_light, color, exp(-optical_depth/smth))
```

Туман и объёмные тени

- Обычный туман можно сделать, вычислив оптическую глубину и подставив в формулу вида
`mix(ambient_light, color, exp(-optical_depth/smth))`
- Для объёмных теней нужно делать интегрирование вдоль луча от камеры до рисуемого пикселя (или наоборот)

Туман и объёмные тени

- Обычный туман можно сделать, вычислив оптическую глубину и подставив в формулу вида
`mix(ambient_light, color, exp(-optical_depth/smth))`
- Для объёмных теней нужно делать интегрирование вдоль луча от камеры до рисуемого пикселя (или наоборот)
- Считаем, что в сцене повсюду однородная ‘пыль’, которая частично поглощает свет, частично излучает ambient свет, и рассеивает солнечный свет

Туман и объёмные тени

- Обычный туман можно сделать, вычислив оптическую глубину и подставив в формулу вида
`mix(ambient_light, color, exp(-optical_depth/smth))`
- Для объёмных теней нужно делать интегрирование вдоль луча от камеры до рисуемого пикселя (или наоборот)
- Считаем, что в сцене повсюду однородная 'пыль', которая частично поглощает свет, частично излучает ambient свет, и рассеивает солнечный свет
- В рассеивании нужно учитывать, находится ли точка луча в тени (так же, как при обычном shadow mapping'e)

Туман и объёмные тени

- Обычный туман можно сделать, вычислив оптическую глубину и подставив в формулу вида
`mix(ambient_light, color, exp(-optical_depth/smth))`
- Для объёмных теней нужно делать интегрирование вдоль луча от камеры до рисуемого пикселя (или наоборот)
- Считаем, что в сцене повсюду однородная 'пыль', которая частично поглощает свет, частично излучает ambient свет, и рассеивает солнечный свет
- В рассеивании нужно учитывать, находится ли точка луча в тени (так же, как при обычном shadow mapping'e)
- Функцию рассеивания можно взять независящей от углов, и просто подогнать коэффициенты чтобы выглядело красиво

Туман и объёмные тени

- Обычный туман можно сделать, вычислив оптическую глубину и подставив в формулу вида
`mix(ambient_light, color, exp(-optical_depth/smth))`
- Для объёмных теней нужно делать интегрирование вдоль луча от камеры до рисуемого пикселя (или наоборот)
- Считаем, что в сцене повсюду однородная 'пыль', которая частично поглощает свет, частично излучает ambient свет, и рассеивает солнечный свет
- В рассеивании нужно учитывать, находится ли точка луча в тени (так же, как при обычном shadow mapping'e)
- Функцию рассеивания можно взять независящей от углов, и просто подогнать коэффициенты чтобы выглядело красиво
- Если число шагов интегрирования будет маленьким, будут артефакты, – ничего страшного, лишь бы не 3 FPS :)

Туман



Объёмные тени



Баллы

- **2 балла:** геометрия сцены загружается и рисуется, движется камера
- **1 балл:** есть источники света, и один имеет яркость больше единицы
- **1 балл:** текстуры альбено (и только они!) загружаются как sRGB
- **2 балла:** есть normal mapping с текстурами нормалей
- либо **3 балла:** есть normal mapping с bump-текстурами
- **4 балла:** есть cubemap-отражения хотя бы на одном движущемся объекте
- **1 балла:** есть простой туман
- **3 балла:** есть объёмные тени

Всего: **15 баллов**

Захита заданий на практике **9 декабря**