**Описание программы**

Программа визуализирует объёмные данные, находящиеся в файле Hand16.raw, в котором записана трёхмерная матрица двубайтовых беззнаковых чисел. Программа выводит изоповерхность для этих данных. Можно менять изозначение, координаты концов ограничивающей коробки, положение и ориентацию камеры. Также можно просматривать данные в стерео-режиме. В консольном окне кратко описан интерфейс и происходит вывод результатов компиляции шейдеров.

[S] – сменить режим стерео

При движении мыши и нажатых ниже клавишах:

[1] – изменить изозначение для выводимой изоповерхности

[3], [4] – редактирование углов ограничивающей коробки

Данные операции можно найти в функциях KeyButton и MouseMove.

Ниже представлено описание содержания файла Hand16.raw.txt, который описывает структуру файла Hand16.raw с исходными данными КТ-томограммы руки.

*size: 244 124 257 //размеры данных в вокселях*

*spacing: 1.000000 1.000000 1.000000 //размеры каджого вокселя (элемента 3д массива)*

*value\_format: USHORT //формат данных, здесь беззнаковое целое*

**Исходный код шейдеров**

В вершинном шейдере происходит обычное преобразование вершин (как правило, вершинный шейдер не используют для вычислений общего назначения). Во фрагментном шейдере находится алгоритм трассировки луча. Для каждого искомого изображения будет запускаться этот алгоритм. На входе алгоритма – координаты фрагмента в пространстве и нормаль к нему. Есть и глобальные параметры, такие как координаты и направление камеры, ограничивающая коробка и т.д., но они одинаковы для всех лучей (потому и хранятся не в varying, а uniform-переменных). На выходе алгоритм трассировки луча выдаёт цвет, степень прозрачности и глубину (для заполнения буфера глубины). Таким образом, при “закрашивании” выводимой геометрии (у нас - ограничивающей коробки) по алгоритму трассировки луча, мы получим изображение объёмных данных.

В алгоритме луч стартует с поверхности ограничивающей коробки и начинает двигаться по направлению от наблюдателя. Движение прекращается либо при выходе луча за пределы коробки, либо при более раннем завершении алгоритма (например, при столкновении с изоповерхностью).

**Исходный код программы**

В функции initIVP(); происходит основная инициализация – создание шейдерной программы, загрузка объёмных данных. В папке MyGraph – различные вспомогательные функции, структуры, классы. В основном используется структура vec3 (трёхкомпонентный вещественный вектор). Класс ShaderProgram используется для работы с шейдерами – в конструкторе происходит загрузка и компилляция шейдеров, с помощью методов Use и UnUse можно использовать шейдер (выводить геометрию с его использованием), а методом SetVar можно менять uniform-переменные шейдера (предварительно вызвав Use).



Ожидаемый результат запуска исходной программы (CT\_tutorial.exe).

Задания:

1. Заменить чёрно-белую раскраску в режиме объёмного рендеринга на красно-белую (модифицировать wlDVR.fs), т.е. в качестве transfer function будет использоваться не градиент от чёрного к белому, а градиент от красного к белому.
2. Загрузить уменьшенную версию (в 2 раза по каждому измерению) томограммы руки (файл Hand16.raw), т.е. найти место в программе, где эти данные загружаются с диска, сформировать уменьшенный массив и загрузить его в GPU (вместо исходного массива).
3. Добавить uniform-переменную level2 в DVR.fs, чтобы пользователь мог её менять, аналогично переменной level1. level2 уже объявлена в функции main и используется, как граница диапазона раскрашиваемых значений данных, но пользователь не может её менять отдельно от level1 (т.к. level2 всегда равна level1+0.01). Переменная level2 уже используется в цикле трассировки, там её менять не нужно. Таким образом, нужно:

А) Удалить строку с объявлением локальной переменной level2

Б) Добавить uniform-переменную level2

В) В основной программе добавить контроль пользователя над переменной level2 по клавише ‘2’(нужно посмотреть, как этот контроль сделан для переменной level1)

1. Реализовать технику рендеринга MIP (Maximum Intensity Projection). Предлагается модифицировать DVR.fs, заменив цикле трассировки накопление цвета и непрозрачности на поиск максимального значения.



Ожидаемый результат рендеринга техникой MIP.