

Задача Wireframe – Построение проволочной модели, В-сплайны, управление камерой

ПОСТАНОВКА ЗАДАЧИ

Требуется разработать приложение для отображения в виде проволочной модели тел вращения с образующей в виде В-сплайна. Научится управлять камерой и освоить матричные преобразования координат.

Цель: Усвоить практическое применение преобразований в однородных координатах, преобразование модели в мировую систему координат (размещение на сцене), затем в систему координат камеры, проецирующие преобразования (перспективное), клиппирование по полукубу.

Итак, пусть поверхность вращения задана параметрически как:

$$r = r(u, v) = (x(u, v), y(u, v), z(u, v)), \text{ где } (u, v) \in D = [a; b] \times [c; d].$$

Шаг 1. Построить проволочную модель поверхности S в модельном пространстве. Для этого строим сетку на области D размером $n \times m$ **клеток** (не узлов) – $u_0 = a, \dots, u_n = b$ и $v_0 = c, \dots, v_m = d$. При этом значения a, b, c, d имеют следующие ограничения: $0 \leq a \leq b \leq 1$ и $0 \leq c \leq d \leq 2\pi$.

Далее строим пространственные ребра – 2 семейства отрезков:

- а) для $i = 1..n, j = 0..m$: $[r(u_{i-1}, v_j) - r(u_i, v_j)]$;
- б) для $i = 1..n, j = 0..m$: $[r(u_i, v_{j-1}) - r(u_i, v_j)]$.

Другими словами, мы построили два семейства линий для поверхности, которые соответствуют параметрическим линиям $u = const, v = const$.

Поверхность S задана как поверхность вращения. Образующую G задаёт пользователь в специальном диалоге (или режиме работы программы). Пользователем задаются опорные точки $\{G_i = (x_i, y_i)\}$, где $i = 1..N$, по которым строится В-сплайн. Как построить образующую $G: \{G_i\} \rightarrow \{x, y\}$ описано в лекциях.

В данной задаче для построения поверхности вращения используется не весь В-сплайн целиком, а только часть, которая определяется интервалом $[a; b]$, где $0 \leq a \leq b \leq 1$. Часть определяется по длине кривой, то есть если общая длина В-сплайна, построенного по точкам $\{G_i: i = 1..N\}$ равна l , то для построения поверхности вращения необходимо использовать участок $[l \times a; l \times b]$ отложенный вдоль кривой (по длине). Для изменения общей длины В-сплайна необходимо разбить каждую часть В-сплайна, на достаточное число частей, например, на 100. И просуммировать длины полученных отрезков. Фактически требуется реализовать функцию вида: $f(u): u \in [0; 1] \rightarrow \{k, t\}$, где u – положение точки на В-сплайне, то есть расстояние вдоль В-сплайна от точки $G(u)$ до начала В-сплайна равно $l \times u$, а k – номер опорной точки, такой что точка $G(u)$ принадлежит куску В-сплайна между точками P_k и P_{k+1} (для его построения используются точки P_{k-1}, P_k, P_{k+1} и P_{k+2}), а значение t – это значение параметра кривой на данном отрезке, соответствующей точке $G(u)$. Значение функции должно вычисляться приблизительно.

Узлы сетки также определяются с помощью функции f , при этом $u = a(1 - \delta) + b\delta$, где $\delta = i/n$, а $i = 0..n$.

Тогда поверхность вращения задана относительно оси OZ следующим образом:

$$\begin{aligned}x(u, v) &= G(u)_y \times \cos(v) \\y(u, v) &= G(u)_y \times \sin(v) \\z(u, v) &= G(u)_x\end{aligned}$$

Для однозначного расположения тела вращения в пространстве определим точку $C = (C_x, C_y, C_z)$, соответствующую опорной точке $(0,0)$. Ориентацию тела зададим матрицей поворота R относительно точки C (не забывайте, что надо переместить систему координат). Повороты нужно применить к отрезкам в порядке поворот относительно оси OX , затем поворот относительно оси OY , затем поворот относительно оси OZ . После необходимо добавить перемещение в тела в точку C .

Должна быть реализована возможность добавлять, удалять и двигать опорные точки. При редактировании пользователь может видеть опорные точки в виде небольших кружков с центрами соответствующими, контрольным точкам. Точки можно перемещать, двигая тем самым опорные точки. По горизонтали отложены значения x , по вертикали значения y . По центру значения $(0,0)$. Оси должны быть видны как на рисунке ниже.

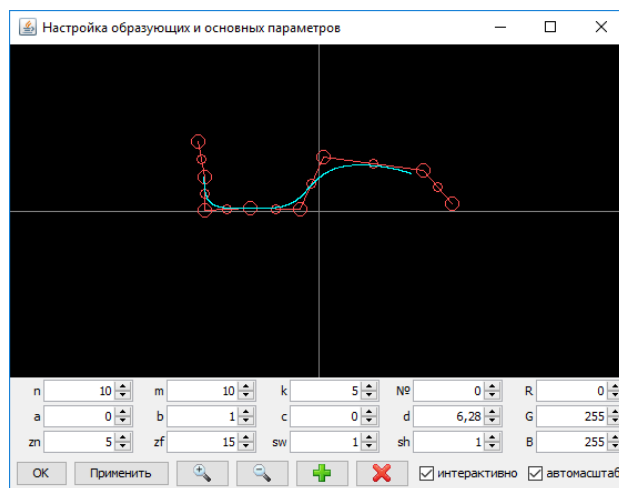


Рис. 1. Пример диалога для редактирования B-сплайнов и параметров поверхности вращения.

Для удобства редактирования необходимо масштабировать множество опорных точек на область редактирования и отобразить B-сплайн, соответствующий текущему расположению опорных точек. Необходимо также добавить риски с шагом 1 на каждой оси. Коэффициент сжатия/растяжения по обеим осям должен быть одинаковым.

На сцене может присутствовать несколько моделей, заданных пользователем. Должна быть возможность добавить новое тело (открывается диалог для задания образующей) и удалить какое-либо из существующих.

Необходимо реализовать возможность редактирования любого тела. Для этих целей в диалоге редактирования параметров тел можно реализовать возможность переключения тел или какое-то тело должно быть помечено как редактируемое (это должно быть видно). После добавления или удаления фигуры пересчитывается габаритный бокс, и сцена приводится к начальному положению (см. шаг 3). Для всех тел вращения параметры a, b, c, d одинаковые. При изменении a, b – часть кривой, которая лежит за пределами этих значений должна отображаться другим

цветом (например, серым), то есть её должно быть видно, чтобы можно было использовать управляющие точки, но при этом было понятно, что она не будет частью образующей фигуры вращения. Нужно так же показать на кривой разбиение на n частей, например, мелкими рисками, можно добавить возможность включения и выключения отображения этих отметок, если они визуально мешают редактированию.

Внимание: Сразу после запуска программы какая-то фигура должна визуализироваться (предопределите набор точек в программе).

Шаг 2. Выбор n и m может оказаться таким, что некоторые особенности поведения поверхности окажутся упущенными. Для более точного приближения проволочной модели к поверхности зададим сечения более подробно, т.е. выберем ещё несколько дополнительных точек на каждом ребре – разобьём его на k более мелких отрезков. Можно, конечно, взять более мелкую сетку (т.е. большие значения n и m , но тогда может получиться совсем неразборчивый чертёж. Таким образом, мы увеличиваем число пространственных отрезков, представляющих нашу проволочную (wireframe) геометрическую модель поверхности, в k раз. В таком случае изображение выглядит более гладким. На приведённом ниже рисунке $k = 5$.

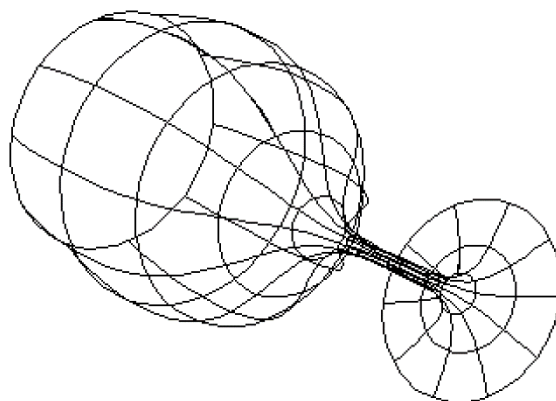


Рис. 2. Пример построенной проволочной модели, соответствующей контрольным точкам, приведённым на рис. 1.

Шаг 3. Мы построили в модельных координатах нашу сцену в виде набора отрезков – множество отрезков. Подсчитываем габаритный объем – габаритный бокс (ориентированный параллельно осям координат), т.е. минимальные и максимальные значения координат. Далее, сдвигаем и масштабируем с сохранением пропорций так, чтобы габаритный бокс был вписан в $[-1,1] \times [-1,1] \times [-1,1]$. Затем поворачиваем отрезки матрицей поворота E . Комбинация сдвигов и поворотов на данный момент описывается промежуточной матрицей однородного преобразования M_1 .

Шаг 4. Задаём положение камеры тремя тройками в мировой системе координат:

$P_{cam} = (x_{cam}, y_{cam}, z_{cam})$ – точка положения верхушки камеры – точка $(-10,0,0)$

$P_{view} = (x_{view}, y_{view}, z_{view})$ – точка, на которую смотрит камера (view point) – точка $(10,0,0)$

$V_{up} = (x_{up}, y_{up}, z_{up})$ – вектор направления вверх (up-vector) – $(0, 1, 0)$

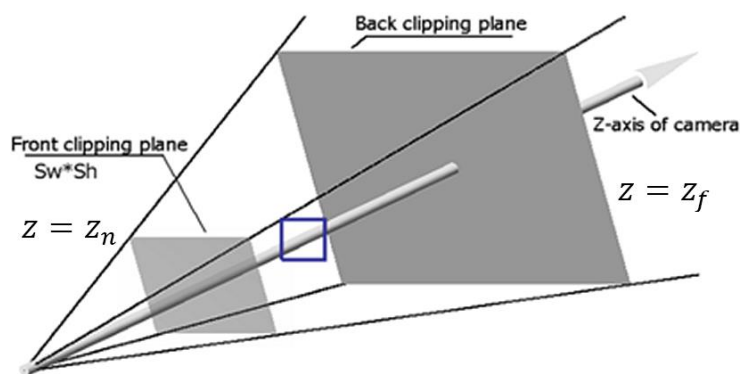


Рис. 3. Камера.

Далее зададим параметры пирамиды видимости (pyramid of view) и объёма визуализации (viewing frustum):

- а) z_n – расстояние до ближней клиппирующей плоскости;
- б) z_f – расстояние до дальней клиппирующей плоскости;
- в) s_w, s_h – размеры грани объёма визуализации на ближней плоскости.

На клиентской области окна приложения выбираем прямоугольник P в экранных координатах (пикселях) (view port). Между прямоугольником и граница области окна, с небольшими отступами, например, 10 пикселей. Соотношение длины к высоте прямоугольника должно быть равно соотношению s_w к s_h , это необходимо, чтобы отсутствовали искажения изображения. Прямоугольник должен быть максимального размера с учётом отступа и сохранения пропорций, т.е. быть вписан. Прямоугольник должен быть обведён.

Шаг 5. Для перевода всех отрезков из мировой системы координат в систему координат камеры служит матрица M_{cam} (построить).

Пересчитываем координаты всех отрезков в систему координат камеры, используя полное преобразование, которое определяется матрицей $M_{to\,cam}$ равной $M_{cam} \times M_1$. Это отрезки, представляющие проволочную модель поверхности.

Затем, применяем полное преобразование проецирования – матрица M_{proj} , т.е. переводим точки из модельной системы координат в точки видимого объёма – точки полукуба $[-1,1] \times [-1,1] \times [0,1]$.

Шаг 6. Клиппируем все отрезки границей полукуба и изображаем в прямоугольник P те их части, которые попали внутрь его.

ОСОБЕННОСТИ РЕАЛИЗАЦИИ

1. **Самое главное: должна применяться только матричная арифметика.** Все преобразования рассчитывать, как матрицы, формировать результирующую матрицу, а затем применять её к точкам. Использовать однородные координаты. Нарушение этого условия – максимум 2 балла.
2. Должна быть возможность изменять значения n, m, k, a, b, c, d . Эти значения задаются изначально в файле данных. Файл с данными должен быть в папке <DATA>.
3. Углы поворота сцены e_x, e_y, e_z меняются с помощью перемещения мыши с прижатой клавишей. Придумайте, как это сделать. При этом перемещение точек фигур должно происходить параллельно направления движения мыши. Точки ближе чем центр габаритного бокса к наблюдателю должны двигаться в одном направлении с мышью, а точки дальше центра в противоположном направлении, то есть вращения относительно центра бокса.

4. Должен быть режим поворота конкретного тела, т.е. изменения углов поворота r_x, r_y, r_z . Изменение значений должно быть реализовано с мышкой с помощью прижатой клавиши. В этом же режиме необходимо иметь возможность изменять координаты точки C для тела. Возможно просто в диалоге, можно клавишами, например, перемещение вдоль осей.
5. Параметры $n, m, k, a, b, c, d, z_n, z_f, s_w, s_h$ должны также задаваться в диалоге.
6. Значения, задающие ракурс камеры: $P_{cam}, P_{view}, V_{up}$ – фиксированные
7. Просто wireframe без удаления невидимых линий.
8. При создании нового тела начальные опорные точки, положение и поворот тела выбирать как-то разумно, чтобы тело уже можно было видеть внутри сцена, даже если ничего не менять (если другие тела тоже видны).
9. Цвета новых тел можно генерировать случайно.
10. z_n камеры можно изменять с помощью колёсика мыши. При скроле к себе z_n должен увеличиваться, то есть угол обзора камеры уменьшаться, а значит отображаемые объекты должны становиться больше. При скроле от себя происходит обратное. При этом можно изменять значение z_f пропорционально.
11. Необходимо отобразить оси для каждого тела просто добавьте три отрезка для каждого тела выходящих из точки C . Цвета отрезков красный для оси OX , зелёный для оси OY , синий для оси OZ . Это нужно сделать до применения матриц поворота тела.
12. Необходимо отобразить оси всей сцены, если тел больше, чем одно. Таким же образом, как и для тела только из центра габаритного бокса. Цвета те же.
13. Фоновый цвет и цвет тел задаётся в файле. Цвет каждого тела можно изменять.

ФОРМАТ ФАЙЛА СЦЕНЫ

Файл сцены содержит описание всех примитивов сцены. Расширение файла "txt". Обратите внимание, формат файла допускает комментарии и пустые строки (как и в предыдущих задачах).

```
// Через пробел или табуляцию
n m k a b c d // Общие параметры тел вращения
zn zf sw sh // Параметры пирамиды видимости
E11 E12 E13 // Композиция матриц вращения поворота всей сцены
E21 E22 E23 // 4-я строка и 4-й столбец отброшены
E31 E32 E33
BR BG BV // Фоновый цвет
K // Количество тел вращения
// Дальше описание отдельных тел
// Описание первого тела
R G B // Цвет тела в пространстве RGB
CX CY CZ // Точка в мировой системе координат, соответствующая опорной точке (0, 0)
R11 R12 R13 // Матрица вращения тела, относительно точки CX CY CZ
R21 R22 R23 // 4-я строка и 4-й столбец отброшены
R31 R32 R33
N // Количество опорных точек B-сплайна для образующей текущего тела
X1 Y1 // координаты первой точки
X2 Y2 // координаты второй точки
...
XN YN // координаты N-й точки
// потом второе тело и т.д.
```

Приложение должно иметь следующие кнопки:

1. Загрузка сцены из файла. **[Open]**
2. Сохранение сцены в файл. **[Save]**
3. Сбросить углы поворота сцены **[Init]**
4. Показать диалог редактирования параметров тел **[Settings]**