

Предварительные замечания:

1. Описание алгоритма состоит из описания двух процедур, где процедура SHOWLINE — вспомогательная.
2. Предполагается, что к началу исполнения алгоритма был произведен переход к экранной системе координат и проведена операция кадрирования, в ходе которой координата z каждой точки (координата в экранной системе координат) не отбрасывается, а остается неизменной. Т. е. после перехода к экранной системе координат выполняется преобразование:

$$\begin{bmatrix} \chi' \\ \gamma' \\ \zeta' \\ \alpha' \end{bmatrix} = \begin{bmatrix} W_x/2 & 0 & 0 & W_{cx} + W_x/2 \\ 0 & -W_y/2 & 0 & W_{cy} - W_y/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \chi \\ \gamma \\ \zeta \\ \alpha \end{bmatrix}$$

3. В описании алгоритма подразумевается, что область рисования имеет координаты по x от W_{cx} до $W_{cx} + W_x$, по y от W_{cy} до $W_{cy} + W_y$, где W_{cx}, W_x, W_{cy}, W_y — неотрицательны.
4. Список AEL — список семерок, первый элемент которых — идентификатор многоугольника, второй — цвет многоугольника, а оставшиеся пять — вещественные числа.
5. При сравнении значений координаты y на равенство следует округлять эти значения до целого числа.
6. При рассмотрении ребер многоугольника $[(x_1, y_1, z_1), (x_2, y_2, z_2)]$ считаем, что $y_1 \leq y_2$ (начальная точка ребра находится не ниже конечной), а при равенстве $y_1 = y_2$ выполняется $x_1 \leq x_2$ (начальная точка ребра находится не левее конечной).

Алгоритм 1: Алгоритм отсечения невидимых граней с использованием S-буфера

Вход: \mathcal{P} — список многоугольников трехмерной сцены

начало алгоритма

- Для каждого многоугольника в \mathcal{P} вычислить $y_{P\min}$ и $y_{P\max}$ — значения минимальной и максимальной координаты y для вершин многоугольника;
- Удалить из \mathcal{P} все многоугольники, у которых $y_{P\min} > W_{cy} + W_y$ или $y_{P\max} < W_{cy}$;
- Определить значение y_{Pnext} — минимальное значение $y_{P\min}$ для многоугольников в \mathcal{P} ;
- $y_t = y_{Pnext}$;
- **цикл пока** $y_t \leq W_{cy} + W_y$ **выполнять**
 - **если** $y_t = y_{Pnext}$, **то**
 - Занести в список APL все многоугольники из \mathcal{P} , для которых $y_{P\min} = y_t$, удалив их из \mathcal{P} ;
 - $y_{APLnext} = y_t$;
 - **если** список \mathcal{P} *пуст*, **то** присвоить $y_{Pnext} = \infty$;
 - **иначе**
 - Определить значение y_{Pnext} — минимальное значение $y_{P\min}$ для многоугольников в \mathcal{P}
 - **если** $y_t \geq W_{cy}$ **то**
 - Инициализировать строку растра, соответствующую координате y_t ;
 - Заполнить строку растра цветом фона. Определить вещественнозначный массив Z с количеством элементов равным количеству пикселей в строке растра;
 - **если** $y_t = y_{APLnext}$, **то**
 - Добавить в AEL все семерки $\left(P, C, x_1, z_1, y_2, \frac{x_2 - x_1}{y_2 - y_1}, \frac{z_2 - z_1}{y_2 - y_1}\right)$, составленные для ребер многоугольников из APL , у которых $y_1 = y_t$ и $y_1 \neq y_2$;
 - **если** $y_t \geq W_{cy}$ **то**
 - Для всех ребер многоугольников в APL , у которых $y_1 = y_t$ и $y_1 = y_2$ выполнить $\text{SHOWLINE}(y_t, (x_1, z_1), (x_2, z_2), C)$, где C — цвет соответствующего многоугольника;
 - Для ребер многоугольников в APL найти $y_{APLnext}$ — минимальное значение y_1 такое, что $y_1 > y_t$. Если таких ребер нет в APL , то присвоить $y_{APLnext} = \infty$;
 - Упорядочить AEL по значению 1-го элемента семерки, затем по возрастанию 3-го, и затем по возрастанию 6-го;
 - Найти $y_{AELnext}$ — минимальное значение пятого элемента в семерках в AEL ;
 - **если** $y_t \geq W_{cy}$ **то**
 - $i = 1$;
 - **цикл пока** $i \leq |AEL|$ **выполнять**
 - Выполнить $\text{SHOWLINE}(y_t, (x_i, z_i), (x_{i+1}, z_{i+1}), C_i)$, где $(P_i, C_i, x_i, z_i, y_i, \Delta_i x, \Delta_i z)$ обозначает i -й элемент списка AEL ;
 - $i = i + 2$;
 - $y_t = y_t + 1$;
 - **если** $y_t \geq y_{AELnext}$, **то**
 - удалить из AEL семерки с пятым элементом меньшим или равным y_t ;
 - Обновить значение $y_{AELnext}$;
 - В каждой семерке $(P_j, C_j, x_j, z_j, y_j, \Delta_j x, \Delta_j z)$ в AEL заменить x_j на $x_j + \Delta_j x$, z_j на $z_j + \Delta_j z$;

конец алгоритма

Алгоритм 2: SHOWLINE

Вход: y_0 — координата y строки растра, (x_1, z_1) — координаты x и z первой точки, (x_2, z_2) — координаты x и z второй точки. C — цвет соответствующего многоугольника.

Выход: Измененные Z -буфер и область рисования.

начало алгоритма

- **если** $y_0 < W_{cy}$ **или** $y_0 > W_{cy} + W_y$, **то закончить алгоритм;**
- **если** $x_2 < W_{cx}$ **или** $x_1 > W_{cx} + W_x$, **то закончить алгоритм;**
- $x = x_1, z = z_1; \Delta z = \frac{z_2 - z_1}{x_2 - x_1};$
- **если** $x < W_{cx}$, **то**
 - $$z = z + (W_{cx} - x)\Delta z;$$
$$x = W_{cx};$$
- **цикл пока** $x \leq x_2$ **и** $x \leq W_{cx} + W_x$ **выполнять**
 - **если** $Z[x] \leq z \leq 1$, **то**
 - Присвоить $Z[x] = z$ и закрасить в растровой области точку с координатами (x, y_0) цветом C ;
 - Присвоить $x = x + 1, z = z + \Delta z$;

конец алгоритма
