

# **ЛАБОРАТОРНЫЕ РАБОТЫ ПО КОМПЬЮТЕРНОЙ ГРАФИКЕ**

**Учебно-методическое пособие  
по курсу «Компьютерная графика»  
для студентов института прикладной математики  
и компьютерных наук по направлению 01.03.02 –  
Прикладная математика и информатика**



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК

ЛАБОРАТОРНЫЕ РАБОТЫ  
ПО КОМПЬЮТЕРНОЙ ГРАФИКЕ

Учебно-методическое пособие  
по курсу «Компьютерная графика»  
для студентов института прикладной математики и компьютерных наук  
по направлению 01.03.02 – Прикладная математика и информатика

Томск  
2019

РАССМОТРЕНО И УТВЕРЖДЕНО методической комиссией института  
прикладной математики и компьютерных наук

Протокол № 5 от «6» мая 2019 г.  
Председатель МК ИПМКН Костюк Ю.Л.

В данном учебно-методическом пособии даны краткие теоретические положения и задания для выполнения индивидуальных работ по темам «Движение объекта на плоскости», «Построение проекции рёберной сцены с движением», «Построение проекции граневой сцены с движением и с удалением нелицевых граней», «Построение поверхности методом плавающего горизонта». Приведены методические указания по выполнению указанных работ и примеры выполнения заданий.

Предназначено для преподавателей, аспирантов, студентов и магистрантов, слушателей ФПК.

СОСТАВИТЕЛИ: Н.Б. Буторина, Е.Г. Пахомова, В.В. Андреева

## СОДЕРЖАНИЕ

1. Двумерные геометрические преобразования .....	4
2. Построение проекции трёхмерной рёберной сцены .....	12
3. Удаление нелицевых (невидимых) граней выпуклого тела .....	27
4. Удаление невидимых линий поверхности методом плавающего горизонта	31
5. Литература .....	41

# ЛАБОРАТОРНАЯ РАБОТА №1

## Двумерные геометрические преобразования

Рассмотрим следующие преобразования точки:

1. элементарные (сдвиг, поворот и масштабирование);
2. совмещенные.

### Условные обозначения

$(x, y)$  – старые координаты точки,

$(x_1, y_1)$  – новые координаты точки.

### Формулы и матрицы преобразований

#### 1) преобразование сдвига (переноса).

Пусть  $T_x$  – константа сдвига вдоль оси  $x$ ,

$T_y$  – константа сдвига вдоль оси  $y$ .

Например,  $x = 2, y = 3, T_x = 3, T_y = 2$ . (Рис. 1)

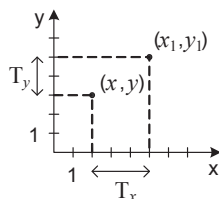


Рис. 1.

Связь координат точки выражают формулы:

$$x_1 = x + T_x;$$

$$y_1 = y + T_y,$$

или

$$[x_1 \ y_1 \ 1] = [x \ y \ 1] * [T], \quad \text{где } [T] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix}.$$

#### 2) преобразование масштабирования

Пусть  $S_x, S_y$  – коэффициенты масштабирования.

Например,  $x = 2, y = 3, S_x = 2, S_y = 1/2$ . (Рис. 2)

Связь координат точки выражают формулы:

$$x_1 = S_x * x;$$

$$y_1 = S_y * y,$$

или

$$[x_1 \ y_1 \ 1] = [x \ y \ 1] * [S], \quad \text{где } [S] = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

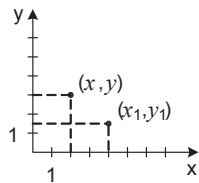


Рис. 2.

**Замечание.** Масштабирование всегда происходит относительно начала координат!

### 3) преобразование поворот

**Замечание.** Поворот всегда происходит относительно начала координат в направлении против часовой стрелки! (Рис. 3)

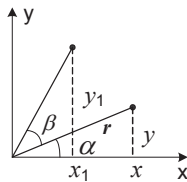


Рис. 3.

Новые координаты точки вычисляются по формуле:

$$\begin{aligned}x_1 &= x * \cos(\beta) - y * \sin(\beta); \\y_1 &= x * \sin(\beta) + y * \cos(\beta),\end{aligned}$$

или  $[x_1 \ y_1 \ 1] = [x \ y \ 1] * [R]$ , где  $[R] = \begin{bmatrix} \cos(\beta) & \sin(\beta) & 0 \\ -\sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$ .

Частный случай поворота – малая величина угла поворота  $\alpha$ . Тогда можно считать, что  $\cos(\alpha) = 1$  и новые координаты точки можно найти по формулам:

$$\begin{aligned}x_1 &= x - y * \sin(\alpha); \\y_1 &= x * \sin(\alpha) + y,\end{aligned}$$

или  $[x_1 \ y_1 \ 1] = [x \ y \ 1] * [R]$ , где  $[R] = \begin{bmatrix} 1 & \sin(\alpha) & 0 \\ -\sin(\alpha) & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

### Подходы к совмещенному двумерному преобразованию

#### 1) Последовательно одно преобразование за другим.

Последовательно выполняем действия по формулам.

$$\begin{aligned}(x_1, y_1) &= F(x, y), \\(x_2, y_2) &= F_1(x_1, y_1).\end{aligned}$$

#### 2) Суперпозиция формул.

Последовательно подставляем формулы одна в другую

$$(x_1, y_1) = F_1(F(x, y)),$$

пока не получим неприводимые формулы вида

$$A * x + B * y + C,$$

где  $A, B, C$  – коэффициенты, не зависящие от точки.

#### 3) Матричный подход

Вычисляется матрица  $[F]$  совмещенного преобразования путем перемножения матриц  $[F_1], [F_2]$  элементарных преобразований:

$$[x_1 \ y_1 \ 1] = [x \ y \ 1] * [F], \text{ где } [F] = [F_1] * [F_2].$$

**При этом надо иметь в виду, что происходит умножение вектора-строки на матрицу!!!**

## ЗАДАНИЕ

Запрограммируйте одну из приведенных ниже задач.

### РЕКОМЕНДАЦИИ

I. Фигуру, подвергающуюся преобразованию, можно задать несколькими способами:

- 1) Массивом  $Otr[Notr][4]$  координат концов отрезков, где  $Notr$  – число отрезков, составляющих фигуру, а  $i$ -я строка – координаты  $A_x, A_y, B_x, B_y$  концов  $i$ -го отрезка  $[A, B]$ .
- 2) Массивом  $Ver[Nver][2]$  вершин фигуры, где  $Nver$  – число вершин, составляющих фигуру, а  $i$ -я строка – координаты  $A_x, A_y$  точки  $A$  и массивом  $Reb[Nreb][2]$  индексов концов ребра, где  $Nreb$  – число рёбер, составляющих фигуру, а  $i$ -я строка – индексы концов  $i$ -го ребра  $[A, B]$  в массиве  $Ver$ .

Например, зададим треугольник, изображенный на рис. 4 двумя способами:

- 1) `double Otr[3][4]={{2, 2, 2, 4},{2, 4, 1, 3},{1, 3, 2, 2}};`
- 2) `double Ver[3][2]={{2, 2},{2, 4},{1, 3}};`  
`int Reb [3][2]={{0, 1},{1, 2},{2, 0}};`

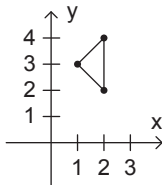


Рис. 4

II. В системе координат экрана ось  $Oy$  направлена вниз. Учтите это на этапе рисования.

III. Решение задачи разбивается на этапы: рисование фигуры на новом месте, пересчёт координат, удаление фигуры с предыдущей позиции.

Так как эффект анимации достигается чередованием прорисовки фигуры и её удалением, то рекомендуется следующий порядок этапов: пересчёт координат, удаление фигуры с предыдущей позиции, рисование фигуры на новом месте.

IV. Удаление фигуры с предыдущей позиции может осуществляться очисткой всего экрана или прорисовкой фигуры со старыми координатами цветом фона.

V. Предусмотрите «задержку» экрана после каждой смены изображения.

VI. Дополните рисунок деталями, делающими его более наглядным.



## ЗАДАЧИ

На рисунках, приведенных в задачах, жирными линиями изображено первоначальное положение фигуры.

### ЗАДАЧА 1

(Стрелка)

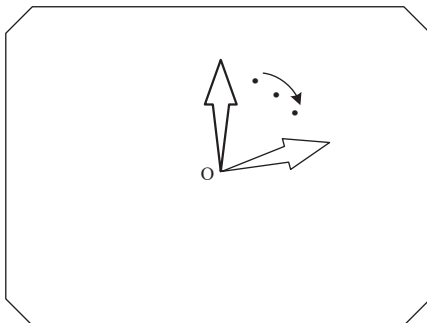
**Фигура:** замкнутая ломаная линия (секундная стрелка).

**Первоначальное расположение:** в центре экрана.

**Движение:** поворот вокруг одной из вершин (точки  $O$ ) на небольшой угол  $\alpha$  по часовой стрелке.

**Остановка:** совершить  $n$  поворотов.

**Комментарии:** на основе полученной программы создайте часы – будильник, звонящие по достижении заданного числа секунд. К рисунку добавьте стрелку будильника.



### ЗАДАЧА 2

(Колесо)

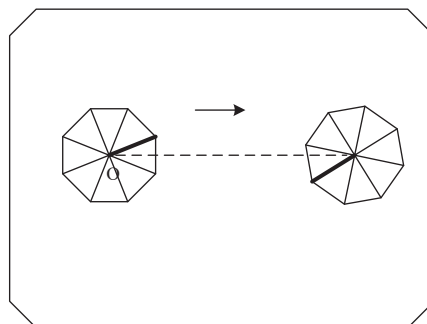
**Фигура:** отрезки прямых (велосипедное колесо). Один из отрезков (спицу колеса) изобразить контрастным цветом.

**Первоначальное расположение:** у левого края экрана.

**Движение:** поворот вокруг заданной точки  $O$  (оси колеса) на угол  $\alpha$  по часовой стрелке, сдвигая фигуру по горизонтали вправо на величину  $T$ .

**Остановка:** перемещать до правого края экрана.

**Комментарии:** проведите пунктирную горизонталь через точки вращения. Величину сдвига выберите такой, чтобы создавалось впечатление катящегося колеса.



### ЗАДАЧА 3 (Вертушка)

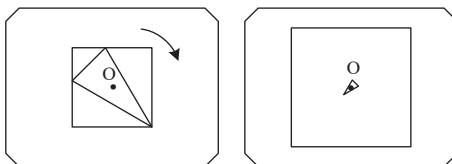
**Фигура:** любые две замкнутые ломаные линии (треугольник внутри квадрата).

**Первоначальное расположение:** в центре экрана.

**Движение:** поворачивать фигуру вокруг точки  $O$  (центр квадрата) на угол  $\alpha$  по часовой стрелке, незначительно уменьшая внутреннюю часть фигуры (треугольник) и во столько же раз увеличивая внешнюю часть (квадрат).

**Остановка:** преобразования повторять до тех пор, пока уменьшающаяся часть не превратится в точку.

**Комментарии:** Картина должна напоминать «вертушку», состоящую из удаляющегося при вращении треугольника и приближающегося квадрата.



### ЗАДАЧА 4 (Маятник)

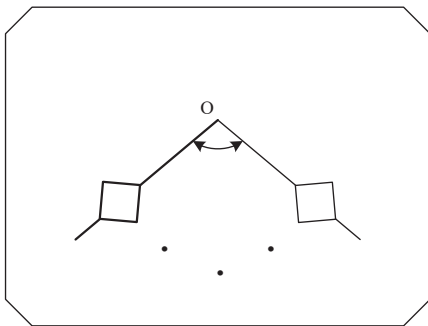
**Фигура:** отрезки прямых.

**Первоначальное расположение:** маятник в крайнем левом положении, отклоненный на  $\beta$  градусов от вертикали.

**Движение:** поворот вокруг точки  $O$  в центре верхней половины экрана (оси маятника) на угол  $\alpha$  против часовой стрелки. Совершив  $n$  поворотов (достигнув крайнего правого положения) поменять направление движения (по часовой стрелке).

**Остановка:** выполнить  $m$  колебаний маятника.

**Комментарии:** процесс должен напоминать колебания маятника. На его основе создайте часы – «ходики».



## ЗАДАЧА 5

(Лист)

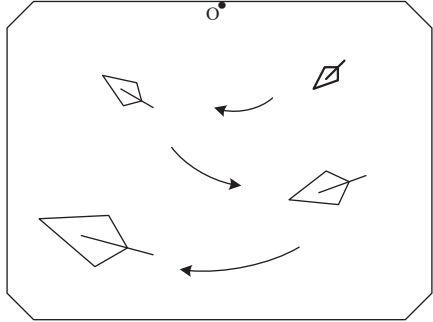
**Фигура:** отрезки прямых (лист дерева).

**Первоначальное расположение:** правая верхняя половина экрана.

**Движение:** поворот вокруг точки  $O$  (центр верхнего края экрана) на угол  $\alpha$  против часовой стрелки, каждый раз незначительно увеличивая размер фигуры. Совершив  $n$  поворотов (достигнув крайнего левого положения) поменять направление движения (против часовой стрелки).

**Остановка:** пока лист не упадет (достижение нижнего правого или левого угла экрана).

**Комментарий:** картина должна напоминать падение осеннего листа, сорвавшегося с дерева.



## ЗАДАЧА 6

(Колесо обозрения)

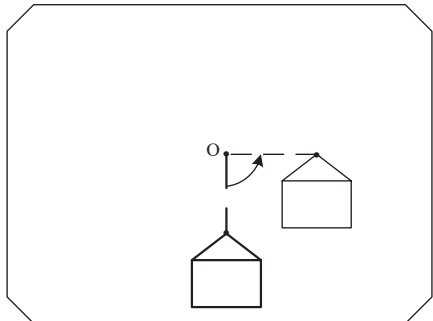
**Фигура:** отрезки прямых (треугольник и прямоугольник с общей стороной, напоминающие кабину колеса обозрения). Одна из точек фигуры особо выделена (точка подвески кабины).

**Первоначальное расположение:** самое нижнее положение кабины.

**Движение:** поворот особой точки вокруг центра экрана (точки  $O$ ) на угол  $\alpha$  против часовой стрелки.

**Остановка:** совершить полный оборот фигуры вокруг центра экрана.

**Комментарий:** картина должна напоминать движение кабины колеса обозрения.



## ЗАДАЧА 7 (Снежный ком)

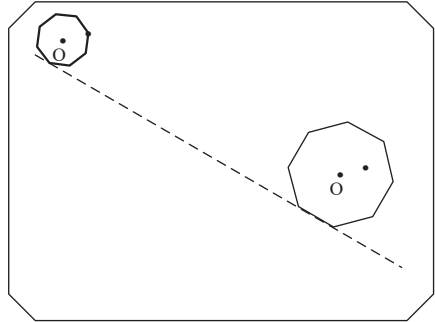
**Фигура:** замкнутая ломаная линия с яркой точкой в одной из вершин (многоугольник, похожий на круг, с ярким «пятном» на окружности).

**Первоначальное расположение:** верхний левый угол экрана.

**Движение:** поворот вокруг центра фигуры (точки  $O$ ) на угол  $\alpha$  по часовой стрелке, незначительно увеличивая размер фигуры и сдвигая фигуру так, чтобы она перемещалась по наклонной прямой. Яркое «пятно» должно совершать те же перемещения, но не увеличивая первоначальное расстояние до точки  $O$ .

**Остановка:** достижение нижнего правого угла экрана.

**Комментарии:** картина должна напоминать катящийся с горы нарастающий снежный ком.



## ЗАДАЧА 8 (Карусель)

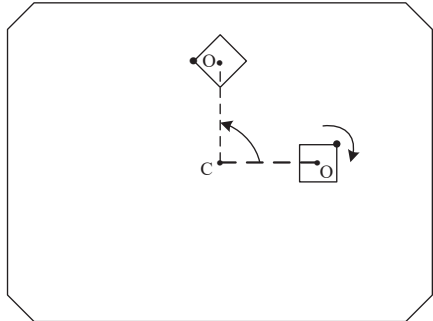
**Фигура:** замкнутая ломаная линия (квадрат).

**Первоначальное расположение:** крайнее правое положение кабины.

**Движение:** поворот вокруг центра фигуры (точки  $O$ ) на угол  $\alpha$  против часовой стрелки и одновременно вокруг центра экрана (точки  $C$ ) на угол  $\beta$  по часовой стрелке

**Остановка:** совершить полный оборот фигуры вокруг центра экрана.

**Комментарии:** одну из вершин фигуры выделить особо. Картина должна напоминать карусель с вращающейся кабиной.



ЗАДАЧА 9  
(Кувырок)

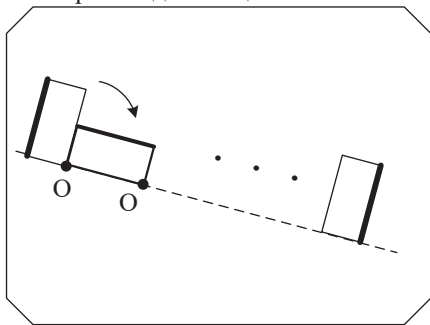
**Фигура:** замкнутая ломаная линия (вытянутый прямоугольник).

**Первоначальное расположение:** крайнее левое положение фигуры, лежащей на наклонной прямой.

**Движение:** поворот вокруг нижней правой вершины (точки  $O$ ) на угол  $\alpha$  по часовой стрелке

**Остановка:** достижение нижнего правого угла экрана.

**Комментарии:** одну из сторон выделить цветом.



## ЛАБОРАТОРНАЯ РАБОТА №2

### Построение проекции трёхмерной реберной сцены

#### Определения

**Сценой** называется совокупность трёхмерных объектов.

**Точкой зрения** называется точка, из которой эта сцена наблюдается.

**Наблюдатель (камера)** – тот, который наблюдает сцену.

**Ось зрения** или **ось наблюдения** проходит через точку зрения и начало координат.

#### Условные обозначения

СКМ – мировая система координат (правая);

СКН – система координат наблюдателя;

СКК – система координат картинной плоскости;

СКЭ<sup>и</sup> – система координат идеализированного экрана (Рис. 5);

СКЭ<sup>ф</sup> – система координат физического экрана (Рис 6);

**Координаты всех точек начальной сцены задаются в СКМ!**

$(x, y, z)$  – координаты точки в СКМ;

$(x_n, y_n, z_n)$  – координаты точки в СКН;

$(x_k, y_k)$  – координаты точки в СКК;

$(x_э, y_э)$  – координаты точки в СКЭ.

$(x_0, y_0, z_0)$  – координаты точки зрения наблюдателя.

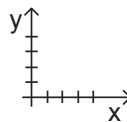


Рис. 5

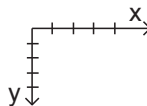
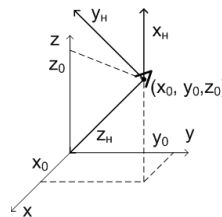


Рис. 6

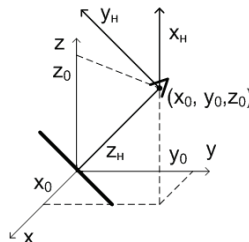
#### Определение СКН

1. Начало СКН в точке зрения  $(x_0, y_0, z_0)$ ;
2. СКН – левая;
3. ось  $y_n$  имеет тенденцию вверх;
4. ось  $z_n$  лежит на оси зрения и направлена в начало СКМ.



#### Определение СКК

1. Проходит через начало СКМ;
2. перпендикулярна оси зрения;
3.  $x_k \parallel x_n, y_k \parallel y_n$ .



## Представление трёхмерной рёберной сцены

$Ver[Nver][3]$  – массив вершин сцены, где  $Nver$  – число вершин, составляющих сцену, а  $i$ -я строка – координаты  $A_x, A_y, A_z$  точки  $A$  –  $i$ -й вершины сцены;

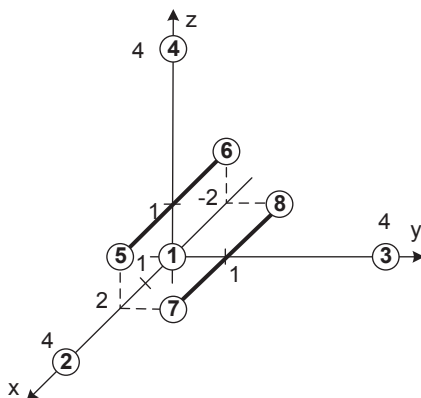
$Reb[Nreb][2]$  – массив индексов концов ребра, где  $Nreb$  – число рёбер, составляющих фигуру, а  $i$ -я строка – индексы концов  $i$ -го ребра  $[A, B]$  в массиве  $Ver$ .

Например, зададим сцену из 5 рёбер (3 оси координат и 2 отрезка).

$$Never = 8, Nreb = 5$$

$$Ver(Nver,3) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 4 & 0 & 0 \\ 3 & 0 & 4 & 0 \\ 4 & 0 & 0 & 4 \\ 5 & 2 & 0 & 1 \\ 6 & -2 & 0 & 1 \\ 7 & 2 & 1 & 0 \\ 8 & -2 & 1 & 0 \end{bmatrix},$$

$$Reb(Nreb,2) = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 1 & 3 \\ 3 & 1 & 4 \\ 4 & 5 & 6 \\ 5 & 7 & 8 \end{bmatrix}.$$



### Алгоритм проецирования

Алгоритм будем демонстрировать на рассмотренном примере для точки зрения  $(x_0, y_0, z_0)$  с координатами (3, 4, 6).

Попробуем наглядно представить ориентацию наблюдателя в мире, пользуясь «астрономической» терминологией. Расположим центр мира в

центре Земли, а точку  $(0,0, +\infty)$  примем за Полярную звезду. Тогда ориентация наблюдателя может быть описана так:

- ось  $y_H$  (тело) – по меридиану головой на север;
- ось  $x_H$  (правая рука) – по параллели на восток;
- ось зрения  $z_H$  (глаз) – к центру Земли.

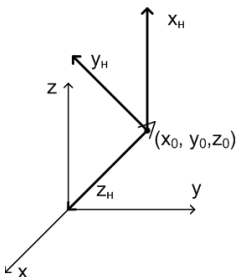
Имеем:  $(x, y, z)$  – первоначальные координаты точек в СКМ;

$(x_1, y_1, z_1)$  – преобразуемые координаты точек.

Алгоритм разбивается на 4 этапа.

### Этап 1. Переход от СКМ к СКН

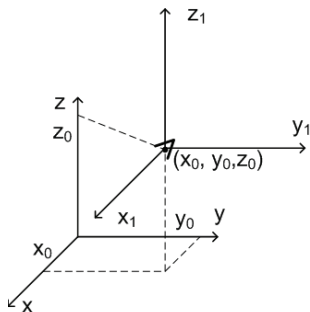
При помощи элементарных преобразований преобразуем мировую систему координат СКМ в систему координат наблюдателя СКН в соответствии с определением СКН. На рисунке слева – СКМ, справа – СКК.



#### **1. Начало СКН в точке зрения $(x_0, y_0, z_0)$ .**

СКМ сдвигаем так, чтобы точка зрения стала началом координат. Соответствующая матрица сдвига

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_0 & -y_0 & -z_0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & -4 & -6 & 1 \end{bmatrix}$$

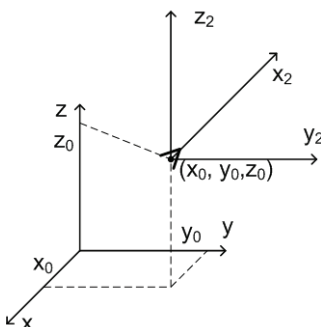




## 2. СКН – левая.

Смена ориентации системы (из правой в левую) происходит за счет изменения направление оси  $OX$  (зеркальное отражение). Соответствующая матрица

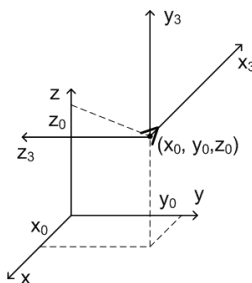
$$[S] = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$



## 3. Ось $y_n$ имеет тенденцию вверх.

Меняем направление оси  $OY$  (поворот вокруг оси  $OX_2$  на  $90^\circ$ ). Угол поворота положительный, т.к. меняется не координаты точки, а система координат и система левая, а не правая (дважды смена знака). Соответствующая матрица

$$[R_x^{90}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$



#### 4. Ось $z_n$ лежит на оси зрения и направлена в начало СКМ.

Так как поворот можно выполнять только вокруг какой-нибудь координатной оси (а не вокруг произвольной прямой), то поворот оси  $OZ_3$  разбивается на два.

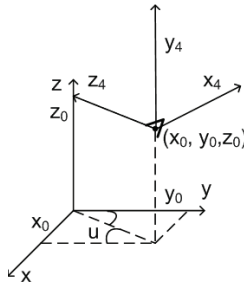
Направляем ось  $OZ_3$  так, чтобы она пересекла ось  $OZ$  (поворот вокруг оси  $OY_3$  на угол  $u$ ).

$$d = \sqrt{x_0^2 + y_0^2} = \sqrt{3*3 + 4*4} = \sqrt{25} = 5$$

$$\cos(u) = \frac{y_0}{\sqrt{x_0^2 + y_0^2}} = \frac{y_0}{d} = \frac{4}{5} = 0,8$$

$$\sin(u) = \frac{x_0}{d} = \frac{3}{5} = 0,6$$

$$[R_y^u] = \begin{bmatrix} \cos(u) & 0 & \sin(u) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(u) & 0 & \cos(u) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0,8 & 0 & 0,6 & 0 \\ 0 & 1 & 0 & 0 \\ -0,6 & 0 & 0,8 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



**Особый случай:**  $d = 0$ , то есть точка зрения принадлежит оси  $OZ$ .

В этом случае угол  $u$  не может быть вычислен ( $\cos(u) = y_0 / 0$ ), но может быть определен как угодно, то есть доопределяется положение наблюдателя на оси  $OZ$ . Положим для определенности  $u = 0$ , то есть делаем матрицу  $R_y^u$  единичной.

В «астрономической» аналогии это означает, что:

- наблюдатель «парит» над Северным полюсом и, значит, всегда ориентирован по меридиану;
- ось  $u_n$  (тело) – по меридиану 90 (головой на Канаду);
- ось  $x_n$  (правая рука) – по меридиану 180 (на Чукотку);
- ось зрения  $z_n$  (глаз) – к центру Земли.

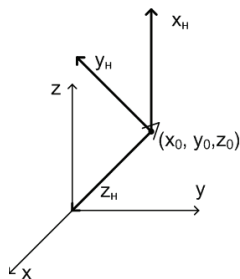
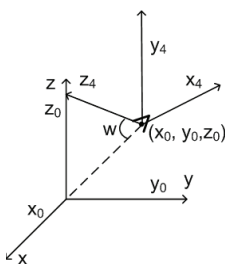
## 5. Поворачиваем ось $OZ_4$ так, чтобы она смотрела в начало координат.

$$s = \sqrt{x_0^2 + y_0^2 + z_0^2} = \sqrt{d^2 + z_0^2} = \sqrt{5*5 + 6*6} = \sqrt{64} \approx 7,8;$$

$$\cos(w) = \frac{d}{\sqrt{x_0^2 + y_0^2 + z_0^2}} = \frac{d}{s} = \frac{5}{7,8} = 0,64;$$

$$\sin(w) = \frac{z_0}{s} = \frac{6}{7,8} = 0,77.$$

$$[R_x^w] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(w) & -\sin(w) & 0 \\ 0 & \sin(w) & \cos(w) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0,64 & -0,77 & 0 \\ 0 & 0,77 & 0,64 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



**Особый случай:**  $s = 0$ , то есть точка зрения находится в начале координат СКМ.

В этом случае не только угол  $u$  не может быть вычислен ( $\cos(u) = y_0 / 0$ ), но и угол  $w$ . Поэтому положим для определенности  $w = u = 0$ , то есть делаем обе матрицы  $R_y^u$  и  $R_x^w$  единичными.

В «астрономической» аналогии это означает, что:

- наблюдатель находится в центре Земли;
- ось  $y_n$  (тело) – по земной оси головой на Северный полюс;
- ось  $x_n$  (правая рука) – на пересечение экватора с меридианом 180 (в центр Тихого океана);
- ось зрения  $z_n$  (глаз) – на пересечение экватора с меридианом 90 западной долготы (в район Панамского канала).

Таким образом, переход СКМ  $\rightarrow$  СКН происходит посредством 5 преобразований и матрица перехода  $[V]$  будет равна

$$[V] = [T][S][R_x^{90}][R_y^u][R_x^w].$$

$$\begin{matrix} \text{СКН} & \text{СКМ} \\ \Rightarrow & [x_n \ y_n \ z_n \ 1] = [x \ y \ z \ 1] [V] \end{matrix}$$

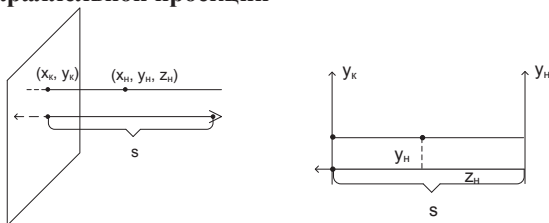
Используя полученное преобразование  $V$  (матрицу перехода), построим массив  $VerNa[Nver, 3]$ .

$$VerNa[8,3] \approx \begin{bmatrix} x_H & y_H & z_H \\ 0 & 0 & 7,8 \\ -3,2 & -1,8 & 6,3 \\ \dots & \dots & \dots \\ 2,2 & 0,3 & 8,1 \end{bmatrix}$$

## Этап 2. Переход от СКН к СКК

Построим проекцию сцены на картинную плоскость (образ сцены), то есть построим массив  $VerKa[8, 2]$ ,  $i$ -я строка которого представит координаты  $x_k, y_k$   $i$ -й вершины в СКК. Картинная плоскость проходит через начало мировой системы координат перпендикулярно оси зрения и расстояние до нее равно  $s \approx 7,8$ .

- Для параллельной проекции

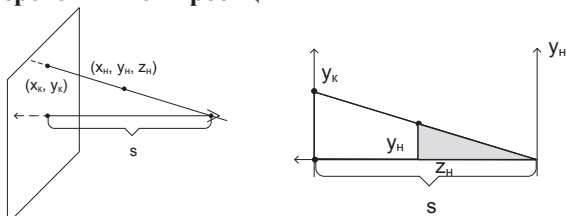


$$x_k = x_H; \quad y_k = y_H.$$

То есть массив  $VerKa$  – это первые два столбца массива  $VerNa$ :

$$VerKa[8,2] \approx \begin{bmatrix} x_K & y_K \\ 0 & 0 \\ -3,2 & -1,8 \\ \dots & \dots \\ 2,2 & 0,3 \end{bmatrix}.$$

- Для перспективной проекции



$$y_K = \frac{y_H}{z_H} s, \quad x_K = \frac{x_H}{z_H} s.$$

**Особый случай:**  $z_n = 0$  для некоторой вершины, то есть проектор такой вершины (прямая, проходящая через вершину и точку зрения), не пересекается с картинной плоскостью. Это возможно, если проектор параллелен либо совпадает с картинной плоскостью. Поэтому данная вершина проекции не имеет.

Для нашего примера:

$$\text{VerKa}[8,2] \approx \begin{bmatrix} x_k & y_k \\ 0 & 0 \\ -3,2 & -1,8 \\ \dots & \dots \\ 2,2 & 0,3 \end{bmatrix}.$$

### Этап 3. Переход от СКК к СКЭ<sup>и</sup>

Зададим в картинной плоскости квадратное окно видимости, полуразмером  $P_k$  (с центром в начале СКК и краями параллельными осям СКК), выбрав окно таким, чтобы проекция полностью поместилась в окне. При выборе полуразмера окна следует иметь в виду, что от размера окна зависят относительные размеры проекции: чем больше окно, тем меньшую его часть занимает проекция.

Положим, например,  $P_k = 5$  единицам.

Зададим на идеализированном экране поле вывода. Пусть, для примера, его параметры таковы:

$$\begin{aligned} x_c &= 320: \text{ центр поля по оси } x \text{ экрана;} \\ y_c &= 175: \text{ центр поля по оси } y \text{ экрана;} \\ x_3 &= 200: \text{ полуразмер поля по оси } x; \\ y_3 &= 200: \text{ полуразмер поля по оси } y. \end{aligned}$$

Отобразим окно картинной плоскости в поле вывода на экране, то есть перейдем к образу сцены в экранных координатах.

Построим массив  $\text{VerEk}[N\text{ver}, 2]$ ,  $i$ -я строка которого представит  $i$ -ю вершину с координатами  $x_3, y_3$ .

$$x_3 = \frac{x_k}{P_k} x_3 + x_c, \quad y_3 = \frac{y_k}{P_k} y_3 + y_c.$$

- Для параллельной проекции:

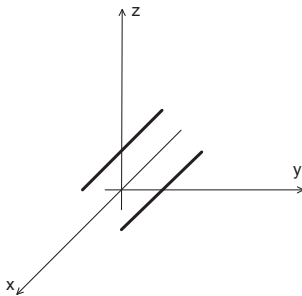
$$\text{VerEk}[8,2] \approx \begin{bmatrix} x_3 & y_3 \\ 0 & 0 \\ -3,2 & -1,8 \\ \dots & \dots \\ 2,2 & 0,3 \end{bmatrix}.$$

- Для перспективной проекции:

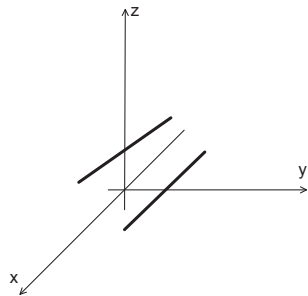
$$VerEk[8,2] \approx \begin{bmatrix} x_3 & y_3 \\ 0 & 0 \\ -3,2 & -1,8 \\ \dots & \dots \\ 2,2 & 0,3 \end{bmatrix}.$$

#### Этап 4. Переход от СКЭ<sup>н</sup> к СКЭ<sup>ф</sup>

Этап 4 – это прорисовка полученных экранных координат на дисплее. Используется массив *VerEk* и заданный массив ребер *Reb*. При прорисовке следует учесть, что в СКЭ<sup>ф</sup> ось *y* направлена вниз. Для нашего примера экранный образ выглядит так:



Параллельная проекция



Перспективная проекция

## ЗАДАНИЕ

Запрограммируйте алгоритмы построения параллельной и перспективной проекций, придерживаясь указанных ниже рекомендаций.

### РЕКОМЕНДАЦИИ

В качестве сцены для отладки программы используйте демонстрационный пример.

Отладив на этом примере программу, задайте реальную сцену из Вашей задачи и проверьте работоспособность программы.

Для наглядности выделите одно ребро и одну вершину другим цветом, оси нарисуйте пунктиром.

Измените несколько раз точку зрения  $(x_0, y_0, z_0)$ , в частности, посмотрите на сцену

- «спереди» – с оси  $OX$ ,
- «сбоку» – с оси  $OY$ ,
- «почти сверху» – из точек, близких к оси  $OZ$ ,
- «сверху» – с оси  $OZ$ .

Для создания эффекта движения сцены дополните алгоритмы циклическим перемещением точки зрения (тип перемещения указан в задаче), стирая старое изображение.

### ЗАДАЧИ

Во всех предложенных задачах реберная сцена представляет собой выпуклое тело (хотя это совсем не обязательно: сцена может быть любой реберной).

На первом рисунке показано первоначальное положение тела, предложенное в задаче. Для наглядности «невидимые» ребра тела изображены пунктиром, оси координат изображены тонкой линией (чтобы не загромождать рисунок), а начало координат обозначено '\*' звездочкой.

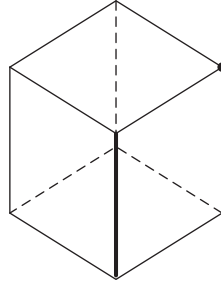
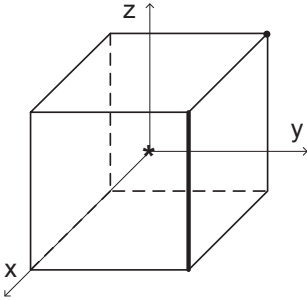
На втором рисунке дан образ, который должен получиться на экране, если наблюдатель находится в предлагаемой точке.

ЗАДАЧА 1  
(Куб)

**Фигура:** куб, центр которого лежит в начале координат, а ребра имеют длину 2 единицы измерения.

**Проекция:** параллельная.

**Движение:** уменьшение координат  $x_0$  и  $y_0$  точки зрения на 1 единицу измерения.



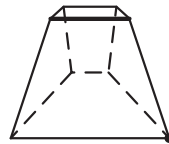
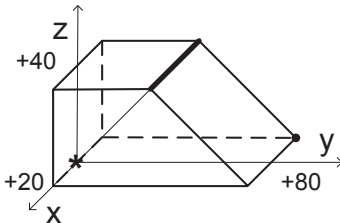
Вид из точки зрения (6, 6, 5).

ЗАДАЧА 2  
(Горка)

**Фигура:** усеченная пирамида (горка) с основанием 40 на 80 единиц измерения, лежащим в плоскости  $XOY$ , размеры верхней площадки 40 на 40, высота горки 40 единиц измерения.

**Проекция:** перспективная.

**Движение:** поворот точки зрения вокруг оси  $ox$  на 15 градусов.



Вид из точки зрения (0, 100, 45).

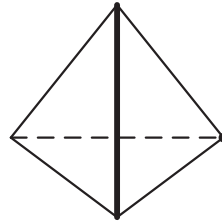
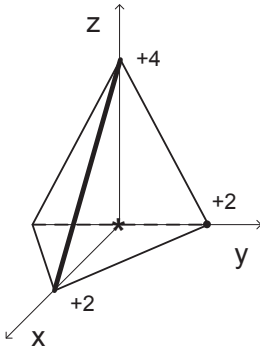


**ЗАДАЧА 3**  
(Пирамида треугольная)

**Фигура:** треугольная пирамида, основанием которой служит равнобедренный прямоугольный треугольник, лежащий в плоскости  $XOY$  так, что центр его гипотенузы и основание высоты совпадают с началом координат, длина гипотенузы и высота пирамиды 4 единицы измерения.

**Проекция:** параллельная.

**Движение:** поворот точки зрения вокруг оси  $OY$  на 15 градусов.



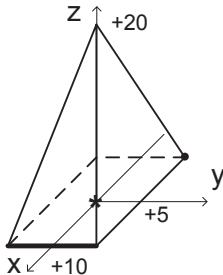
Вид из точки зрения (6, 6, 5).

**ЗАДАЧА 4**  
(Пирамида четырехугольная)

**Фигура:** четырехугольная пирамида, основанием которой служит прямоугольник, лежащий в плоскости  $XOY$  так, что его центр совпадает с началом координат, длина основания – 20 единиц измерения, ширина основания – 10 единиц измерения, высота пирамиды – 20 единиц измерения.

**Проекция:** перспективная.

**Движение:** уменьшение координаты  $x_0$  точки зрения на 10 единиц измерения.



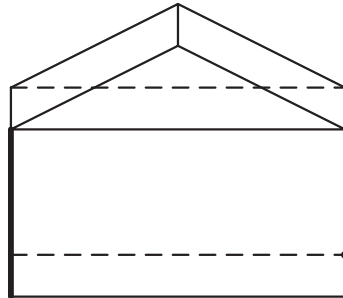
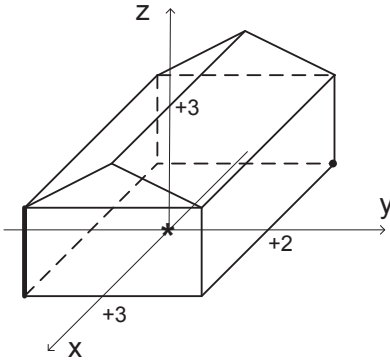
Вид из точки зрения (50, 0, 50).

ЗАДАЧА 5  
(Дом)

**Фигура:** «дом», длина которого 6 единиц измерения, ширина 4 единицы измерения, высота стен 2 единицы измерения, высота крыши 1 единица измерения, пол лежит в плоскости  $XOY$ , а центр пола – в начале координат.

**Проекция:** параллельная.

**Движение:** поворот вокруг оси  $OZ$  на 30 градусов.



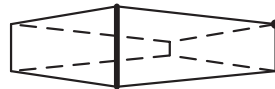
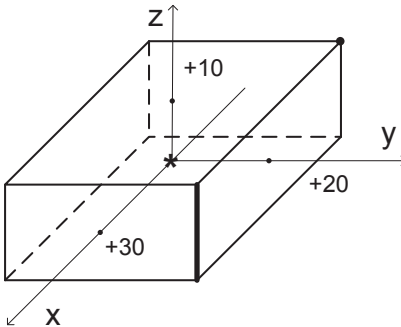
Вид из точки зрения  $(5, 0, 1)$ .

ЗАДАЧА 6  
(Брусок)

**Фигура:** параллелепипед («брусок»), длина которого 6 единиц измерения, ширина 4 единицы измерения, высота 2 единицы измерения, а центр находится в начале координат.

**Проекция:** перспективная.

**Движение:** поворот вокруг оси  $OZ$  на 30 градусов.



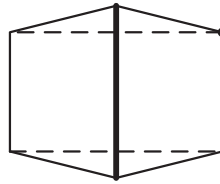
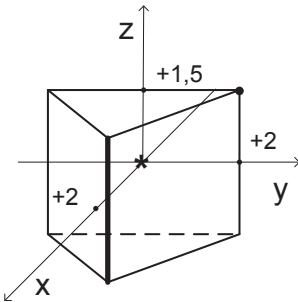
Вид из точки зрения  $(10, 10, 0)$ .

ЗАДАЧА 7  
(Призма)

**Фигура:** прямоугольная призма, в основании которой лежит равнобедренный прямоугольный треугольник. Длина гипотенузы треугольника 4 единицы измерения, высота призмы 3 единицы измерения, центр большой грани совпадает с началом координат.

**Проекция:** перспективная.

**Движение:** уменьшение координаты  $x$  на 2 единицы и увеличение координаты  $OZ$  на 4 единицу.



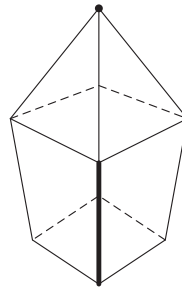
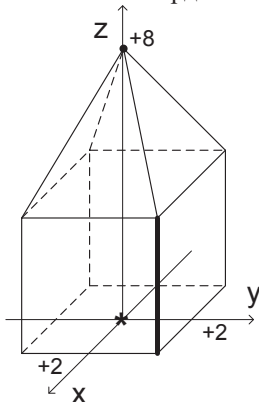
Вид из точки зрения  $(10, 0, 0)$ .

ЗАДАЧА 8  
(Терем)

**Фигура:** четырехугольный «терем», стороны основания, высота стен и высота купола которого составляют 4 единицы измерения, а центр основания лежит в начале координат.

**Проекция:** перспективная.

**Движение:** уменьшение координат  $x$  и  $y$  на 1 единицу.



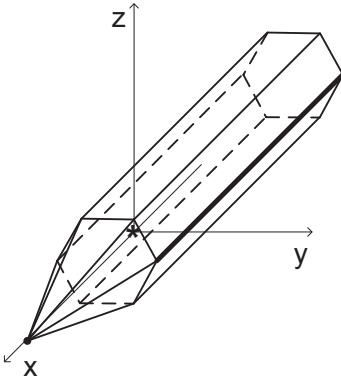
Вид из точки зрения  $(10, 10, 10)$ .

ЗАДАЧА 9  
(Карандаш)

**Фигура:** «карандаш», грифель которого лежит на оси  $x$ , длина карандаша 6 единиц измерения, а толщина – 12 единиц измерения.

**Проекция:** параллельная.

**Движение:** поворот вокруг оси  $x$  на 30 градусов.



Вид из точки зрения  $(0, 10, 0)$ .

## ЛАБОРАТОРНАЯ РАБОТА №3

### Удаление нелицевых (невидимых) граней выпуклого тела

#### Условные обозначения

$N_{gr}$  – число граней, образующих выпуклое тело;

$N_{ver}$  – число вершин в теле;

$\left. \begin{matrix} x_1, y_1, z_1 \\ x_2, y_2, z_2 \\ x_3, y_3, z_3 \end{matrix} \right\}$  – три точки плоскости (три вершины грани);

$Ax + Bx + Cz + D = 0$  – уравнение плоскости;

$[W] = \begin{bmatrix} a_1 & a_2 & \dots & a_{N_{gr}} \\ b_1 & b_2 & \dots & b_{N_{gr}} \\ c_1 & c_2 & \dots & c_{N_{gr}} \\ d_1 & d_2 & \dots & d_{N_{gr}} \end{bmatrix}$  – матрица тела (коэффициенты плоскостей);

$X_c, Y_c, Z_c$  – центр тяжести тела;

$X_0, Y_0, Z_0$  – точка зрения.

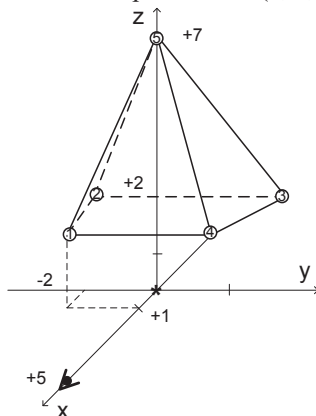
#### Представление выпуклого тела гранями

Предлагается одно из простейших представлений выпуклого тела (на примере четырёхугольной пирамиды, симметричной относительно плоскости  $y = 0$  и «парящей» над плоскостью  $z = 0$ ) двумя массивами:

$Ver(N_{ver}, 3)$  – массив координат вершин,

$Gran(N_{gr}, N_{ver}+1)$  – массив номеров вершин, образующих грани.

Точка зрения находится в точке с координатами  $(5, 0, 0)$ .



$$\text{Ver}(\text{Nver}, 3) = 3 \begin{matrix} x & y & z \\ 1 \left[ \begin{array}{ccc} 1 & -2 & 2 \\ 2 & -1 & -3 & 2 \\ -1 & 3 & 2 \\ 4 & 1 & 2 & 2 \\ 0 & 0 & 7 \end{array} \right. \end{matrix}$$

$$\text{Gran}(\text{Ngr}, \text{Nver} + 1) = 3 \begin{matrix} 1 \left[ \begin{array}{ccccc} 1 & 2 & 3 & 4 & 1 & 0 \\ 2 & 1 & 2 & 5 & 1 & 0 & 0 \\ 2 & 3 & 5 & 2 & 0 & 0 \\ 4 & 3 & 4 & 5 & 3 & 0 & 0 \\ 5 & 1 & 4 & 5 & 1 & 0 & 0 \end{array} \right. \end{matrix} \begin{matrix} \text{нижняя} \\ \text{левая} \\ \text{задняя} \\ \text{правая} \\ \text{передняя} \end{matrix}$$

### Формулы для вычисления коэффициентов плоскостей

$$\begin{aligned} A &= (y_3 - y_1)(z_2 - z_1) - (y_2 - y_1)(z_3 - z_1) \\ B &= (x_2 - x_1)(z_3 - z_1) - (x_3 - x_1)(z_2 - z_1) \\ C &= (x_3 - x_1)(y_2 - y_1) - (x_2 - x_1)(y_3 - y_1) \\ D &= -(Ax_1 + By_1 + Cz_1) \end{aligned}$$

### Формулы для вычисления центра тяжести тела

$$\begin{aligned} X_c &= (x_1 + x_2 + \dots + x_{\text{Nver}}) / \text{Nver}; \\ Y_c &= (y_1 + y_2 + \dots + y_{\text{Nver}}) / \text{Nver}; \\ Z_c &= (z_1 + z_2 + \dots + z_{\text{Nver}}) / \text{Nver}. \end{aligned}$$

### Правило приведения матрицы тела к центру тяжести

Если  $AX_c + BY_c + CZ_c + D < 0$ , то изменить знаки коэффициентов  $A, B, C, D$  на противоположные.

### Правило определения нелицевых граней по приведенной матрице

**Случай СКМ** – тело задано в мировой системе координат, точка зрения произвольна, ось зрения ориентирована в начало координат.

- для перспективной проекции:  
если  $AX_0 + BY_0 + CZ_0 + D \geq 0$ , то грань нелицевая
- для параллельной проекции:  
если  $AX_0 + BY_0 + CZ_0 \geq 0$ , то грань нелицевая

**Случай СКН** – тело задано в системе координат наблюдателя, т. е. точка зрения находится в начале координат, ось зрения совпадает с осью  $z$  и ориентирована в сторону  $z = +\infty$ .

- для перспективной проекции:  
если  $D \geq 0$ , то грань нелицевая
- для параллельной проекции:  
если  $C \leq 0$ , то грань нелицевая

### Алгоритм определения нелицевых граней

1. Вычислите матрицу тела (ее можно проверить подстановкой координат вершин в уравнения плоскостей). Для демонстрационного примера:

$$W = \begin{bmatrix} 0 & -5 & 30 & -5 & 20 \\ 0 & 10 & 0 & -10 & 0 \\ -12 & -5 & -6 & -5 & 4 \\ 24 & 35 & 42 & 35 & -28 \end{bmatrix}.$$

2. Найдите центр тяжести тела  $X_c, Y_c, Z_c$ . Для демонстрационного примера это точка  $C(0, 0, 3)$ .
3. Приведите матрицу  $W$  к центру тяжести. Для демонстрационного примера приведутся первый и последний столбцы матрицы:

$$W^* = \begin{bmatrix} 0 & -5 & 30 & -5 & -20 \\ 0 & 10 & 0 & -10 & 0 \\ 12 & -5 & -6 & -5 & -4 \\ -24 & 35 & 42 & 35 & 28 \end{bmatrix}.$$

4. Для каждой грани проверьте, не является ли она нелицевой, и сформулируйте массив  $\text{GraVid}(N_{\text{vid}}, N_{\text{ver}}+1)$  из видимых граней, где  $N_{\text{vid}}$  – их число. Для демонстрационного примера результаты таковы:
  - для перспективной проекции видны две грани: нижняя (первая) и передняя (пятая);
  - для параллельной проекции видны три грани: передняя (пятая) и боковые (вторая и четвертая).

## ЗАДАНИЕ

Запрограммируйте алгоритм удаления нелицевых граней, ориентированный на один из вариантов:

- 1) перспективная проекция, тело задано в СКМ;
- 2) параллельная проекция, тело задано в СКМ;
- 3) перспективная проекция, тело задано в СКН;
- 4) параллельная проекция, тело задано в СКН.

При выполнении задания, учитывайте следующие замечания:

- При отладке используйте демонстрационный пример и в качестве результата выдайте массив `GraVid`.
- Испытайте программу, задавая различные точки зрения.
- Для прорисовки полученного результата на экране, соедините данную программу с программой построения проекций (лабораторная работа 2).
- Объединённую программу проверьте на демонстрационном примере, а затем на сцене из Вашей задачи к лабораторной работе 2 (с циклическим перемещением точки зрения).
- Попробуйте построить тоновое изображение, закрасив каждую из граней своим цветом.
- Желающие могут использовать возможность представления выпуклого тела координатами вершин без явного задания граней.

## ПРИМЕЧАНИЕ

При стыковке программ удаление нелицевых граней тела и построения его проекций обратите внимание на следующее.

- **Если тело задано в СКМ**, то рекомендуется следующая последовательность действий:
  1. строим матрицу тела;
  2. удаляем нелицевые грани;
  3. СКМ  $\rightarrow$  СКН;
  4. СКН  $\rightarrow$  СКЭ;
  5. новая точка зрения;
  6. повтор пунктов 2 – 5.

То есть матрица тела строится один раз и далее не пересчитывается.

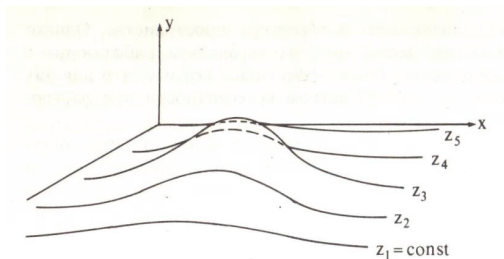
- **Если тело задано в СКН**, то рекомендуется следующая последовательность действий:
  1. СКМ  $\rightarrow$  СКН;
  2. строим матрицу тела;
  3. удаляем нелицевые;
  4. СКН  $\rightarrow$  СКЭ;
  5. новая точка зрения;
  6. повтор пунктов 1 – 5.

Аналитически оцените вычислительную сложность того и другого переходов и остановитесь на более быстродействующем варианте.



## ЛАБОРАТОРНАЯ РАБОТА №4

### Удаление невидимых линий поверхности методом плавающего горизонта



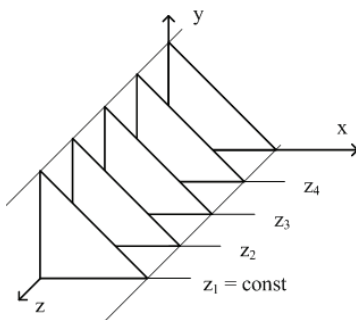
**ИДЕЯ МЕТОДА:** сведение трехмерной задачи к двумерной путём пересечения исходной поверхности последовательностью параллельных секущих плоскостей, имеющих постоянные значения координат  $x$ ,  $y$  или  $z$ .

#### Ограничения

1. Вид поверхности  $y = f(x, z)$ .
2. Находимся в СКН.
3. Рассматриваем только параллельную проекцию.
4. Наблюдатель находится на отрицательной полуоси  $z$ , то есть его координаты  $(0, 0, -\infty)$ .
5. Картинная плоскость совпадает с плоскостью  $xy$ .
6. Повороты либо только вокруг оси  $x$ , либо вокруг осей  $y, x$  (порядок зафиксирован).

#### Определения

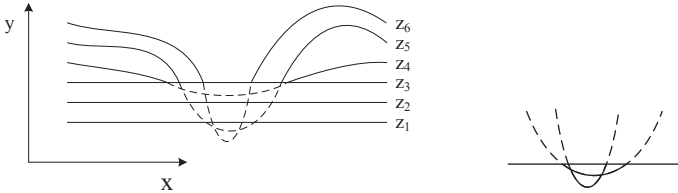
Плоскости, расположенные на одинаковом расстоянии перпендикулярно оси зрения называются **секущими**.



Линия пересечения поверхности секущей плоскостью называется **сечением**, а его проекция на экран – **экранном сечением**.

**Нижним горизонтом  $i$ -го сечения** называют огибающую снизу всех предыдущих сечений.

**Верхним горизонтом  $i$ -го сечения** называют огибающую сверху всех предыдущих сечений.



Наблюдатель находится на отрицательной полуоси  $z$ , видит перед собой рассечённую поверхность, которая до проецирования поворачивается сначала вокруг оси  $y$ , затем вокруг оси  $x$  (обратите внимание, что именно на такую последовательность поворотов ориентированы предлагаемые здесь алгоритмы).

### Условные обозначения

- $y = f(x, z)$  – поверхность с областью определения  $X_{\min} \leq x \leq X_{\max}$ ,  $Z_{\min} \leq z \leq Z_{\max}$  ( $X_{\min} \leq X_{\max}$ ,  $Z_{\min} \leq Z_{\max}$ );
- $z = Z_{\min}, \dots, Z_{\max}$  – семейство секущих плоскостей;
- $Dz$  – шаг по  $z$  (между секущими плоскостями);
- $Dx$  – шаг по  $x$  (между точками сечения);
- $Uy, Ux$  – углы поворота,  $-\pi/2 < Uy, Ux < \pi/2$ ;
- $Cx, Cy$  – центр поля вывода на экране;
- $Px, Py$  – полуразмеры поля вывода на экране;
- $MaxX, MaxY$  – размер экрана;
- $Pkx, Pky$  – полуразмеры окна на картинной плоскости.

### Тест видимости точки

Точка  $Xэ, Yэ$  экранного сечения видна, если лежит выше верхнего и/или ниже нижнего горизонтов для данного сечения:

$$Yэ \geq Up(Xэ) \text{ и/или } Yэ \leq Down(Xэ),$$

где  $Up[MaxX]$  и  $Down[MaxX]$  – массивы горизонтов.

Поэтому признак видимости VID формируется следующим образом:

$$VID = \begin{cases} 0(00), & \text{если точка ниже верхнего и выше нижнего горизонтов;} \\ 1(01), & \text{если точка выше верхнего горизонта;} \\ 2(10), & \text{если точка ниже нижнего горизонта;} \\ 3(11), & \text{если точка выше верхнего и ниже нижнего горизонтов.} \end{cases}$$

## Варианты алгоритма плавающего горизонта

Различные варианты алгоритма могут отличаться по тому, как изображается поверхность:

- точками, в которых вычисляются значения функции, или отрезками между двумя соседними точками;
- в параллельной или перспективной проекциях;
- с поворотами вокруг одной или двух осей.

Рассмотрим некоторые варианты алгоритма, предполагая параллельную проекцию:

- **1 вариант:** поворот вокруг оси  $x$ , рисовать точками;
- **2 вариант:** поворот вокруг осей  $y, x$ , рисовать точками;
- **3 вариант:** поворот вокруг оси  $x$ , рисовать отрезками;
- **4 вариант:** поворот вокруг осей  $y, x$ , рисовать отрезками.

Конечно, варианты 1 и 3 являются частными случаями вариантов 2 и 4, но могут быть реализованы проще, потому рассматриваются отдельно.

### ОБЩАЯ ЧАСТЬ ВСЕХ ВАРИАНТОВ

ДАНО: функция  $y = f(x, z)$ ,  $X_{\min}$ ,  $X_{\max}$ ,  $Z_{\min}$ ,  $Z_{\max}$ ;  
шаги  $Dx$ ,  $Dz$  ;  
поле вывода  $Cx$ ,  $Cy$ ,  $Px$ ,  $Pz$ .

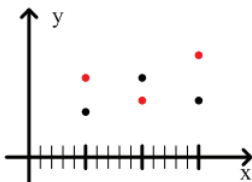
- Вычисляем полуразмеры окна в картинной плоскости (алгоритм ОКНО).
- Устанавливаем горизонты  $Up[MaxX]$  и  $Down[MaxX]$  в 0 и  $MaxY$ . Для вариантов 3, 4 дублируем горизонты ( $Up$ ,  $Down$  назовём старыми;  $NewUp$   $NewDown$  – новыми).
- ЦИКЛ по секущим плоскостям для  $Z_{\min} \leq z \leq Z_{\max}$  с шагом  $Dz$ .
  - ЦИКЛ по точкам сечения  $X_{\min} \leq x \leq X_{\max}$  с шагом  $Dx$ .
    - Вычисляем значение функции  $y = f(x, z)$ .
    - Повернув точку  $x, y, z$ , имеем её проекцию  $X_k, Y_k$  на картинную плоскость.
    - Учтя размеры окна и поля вывода, вычисляем экранные координаты точки  $Y_{\varepsilon}, X_{\varepsilon}$ .
    - Формируем признак её видимости  $VID$ , используя старые горизонты ( $Up$  и  $Down$ ).
    - Вставка для варианта 1-4.
  - ПРОДОЛЖАЕМ ЦИКЛ по точкам сечения.
  - Для вариантов 3, 4 заменяем старые горизонты  $Up$  и  $Down$  новыми  $NewUp$  и  $NewDown$ .
- ПРОДОЛЖАЕМ ЦИКЛ по секущим плоскостям.
- КОНЕЦ.

### Примечания:

- во-первых, поиск параллельной проекции здесь выродился т.к. наблюдатель уже находится на оси  $z$ ;
- во-вторых, поверхность выглядит более наглядной, если её точки, лежащие над верхним горизонтом, нарисованы одним цветом; точки, лежащие под нижним горизонтом – другим цветом; а точки первого сечения – третьим цветом;
- в-третьих, вычисления значения функции  $y = f(x, z)$  можно оформить в виде подпрограммы, тогда алгоритм станет универсальным (ориентированным на любую функцию).

### Вставка для первого варианта

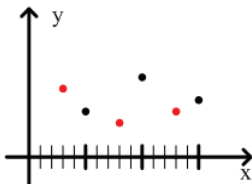
Решаемые задачи – анализ видимости, рисование. На рисунке изображено возможное положение точек предыдущего и текущего сечений.



- ЕСЛИ точка видна, то рисуем её и корректируем горизонты UP и DOWN в данной точке.

### Вставка для второго варианта

Решаемые задачи – коррекция горизонта на отрезке. На рисунке изображено возможное положение точек предыдущего и текущего сечений.

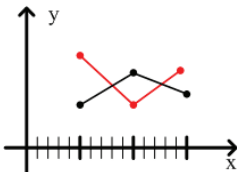


- Обозначим текущую точку  $Y_x, X_x$  через  $t$ , а её признак видимости – через  $VID_t$ .
- ЕСЛИ точка  $t$  видна, то рисуем её.
- ЕСЛИ точка  $t$  не первая в сечении и ЕСЛИ хотя бы одна из точек  $t$  (текущая) или  $p$  (предыдущая) видны, ТО корректируем оба горизонта UP и DOWN отрезком  $[p, t]$  (алгоритм КОРРЕКЦИЯ).
- Запоминаем текущую точку в качестве предыдущей, полагая  $p=t$ ,  $VID_a=VID_t$ .

### Вставка для третьего варианта

Решаемые задачи – нахождение точки пересечения, коррекция горизонта.

На рисунке изображено возможное положение точек предыдущего и текущего сечений.



Возможные в этом варианте ситуации сведены в таблицу

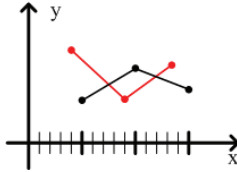
p \ t	00	01	10	11
00				
01				
10				
11				

- Обозначим текущую точку  $Y_x, X_x$  чрез  $t$ , а её признак видимости – через  $VID_t$ .
- ЕСЛИ точка  $t$  не первая в сечении, ТО анализируем видимость отрезка  $[p, t]$  по признакам видимости  $VID_p$  и  $VID_t$  его концов:
  - ЕСЛИ видна хотя бы одна из точек предыдущая ( $p$ ) или текущая ( $t$ ), ТО виден весь или часть отрезка.
    - ЕСЛИ виден отрезок, ТО рисуем его.
    - ИНАЧЕ видна часть.
      - ЕСЛИ виден конец  $p$ , ТО найдя точку пересечения  $s$  отрезка с горизонтом (алгоритм ПЕРЕСЕЧЕНИЕ), рисуем отрезок  $[p, s]$ .
      - ЕСЛИ виден конец  $t$ , ТО, найдя точку пересечения  $q$  отрезка с горизонтом (алгоритм ПЕРЕСЕЧЕНИЕ), рисуем отрезок  $[q, t]$ .
    - Корректируем новые горизонты  $NewUp$  и  $NewDown$  отрезком  $[p, t]$  (алгоритм КОРРЕКЦИЯ)
- Запоминаем текущую точку в качестве предыдущей, полагая:  $p=t$   $VID_p=VID_t$ .

## Вставка для четвертого варианта

Решаемые задачи – нахождение точки пересечения, коррекция горизонта, рисование отрезка.

На рисунке изображено возможное положение точек предыдущего и текущего сечений.



В данном варианте используют растровый подход, а именно, отрезок, соединяющий две точки сечения, разлагают в растр и каждый его пиксел анализируют: видимый – рисуют и им корректируют горизонты.

При этом исчезает необходимость искать точку пересечения отрезка и горизонта. Этот поиск доставляет особые неприятности в ситуациях, отмеченных в таблице словом «нет» (именно они отличают третий вариант от четвертого). Кроме того, двукратное разложение отрезка в растр (для рисования и для коррекции горизонтов) заменяется однократным.

- Обозначим текущую точку  $X_t, Y_t$ , через  $t$ , а её признак видимости через  $VID_t$ .
- ЕСЛИ точка  $t$  не первая в сечении, то анализируем видимость отрезка  $[p, t]$  по признакам видимости  $VID_p$  и  $VID_t$  его концов:
  - ЕСЛИ видна хотя бы одна из точек предыдущая ( $p$ ) или текущая ( $t$ ), ТО виден весь или часть отрезка  $[p, t]$ .  
Для разложения отрезка в растр формируем шаги  $dx$  и  $dy$  по осям, сравнивая  $XX = X_t - X_p$  и  $YY = Y_t - Y_p$ :
$$\ell = \max(|XX|, |YY|).$$
Если  $\ell = 0$ , то  $dx = dy = 0$ ,  
иначе  $dx = xx / \ell$ ;  $dy = yy / \ell$ .
- ЦИКЛ по точкам отрезка  $[p, t]$  от  $p$  к  $t$  с шагом  $dx, dy$ , ( $i$  от 1 до  $\ell$ ).
  - Округляя координаты очередной точки, получаем пиксел растра:  $X, Y$ .
  - ЕСЛИ пиксел  $X, Y$  виден по отношению к старым горизонтам  $Up$  и  $Down$ , ТО рисуем новые горизонты  $NewUp$  и  $NewDown$  этим пикселом.
  - ПРОДОЛЖАЕМ ЦИКЛ по точкам отрезка.
- Запоминаем текущую точку в качестве предыдущей, полагая:  $p = t$ ,  $VID_p = VID_t$ .

## Алгоритм ОКНО

**Дано:** функция  $y = f(x, z)$ ,  $X_{\min}$ ,  $X_{\max}$ ,  $Z_{\min}$ ,  $Z_{\max}$ ;  
шаги  $D_x$   $D_z$ ; углы поворота  $U_y$ ,  $U_x$ .

**Найти:** полуразмеры окна  $P_{kx}$  и  $P_{ky}$ .

- Вычисляем картинные абсциссы  $X_1$ ,  $X_2$ ,  $X_3$ ,  $X_4$  угловых точек области определения:  
 $X_{\min}, Z_{\min}$ ;  $X_{\max}, Z_{\min}$ ;  $X_{\min}, Z_{\max}$ ;  $X_{\max}, Z_{\max}$ ;  
выполнив поворот этих точек вокруг оси  $y$ .
- Определяем  $P_{kx} = \max(|X_1|, |X_2|, |X_3|, |X_4|)$ .
- Определяем  $P_{ky}$ , положив первоначально  $P_{ky} = 0$  и перебирая значения функции:
  - ЦИКЛ по секущим плоскостям  $Z$  для  $Z_{\min} \leq Z \leq Z_{\max}$  с шагом  $D_z$ .
    - ЦИКЛ по точкам  $X, Y$  сечения  $Z$  для  $X_{\min} \leq X \leq X_{\max}$  с шагом  $D_x$ .
      - Вычисляем значение функции  $Y = f(X, Z)$ .
      - Повернув точку  $X, Y, Z$  на углы  $U_y, U_x$ , имеем ескартинную ординату  $Y_k$ , корректируем  $P_{ky} = \max(P_{ky}, |Y_k|)$ .
      - ПРОДОЛЖАЕМ ЦИКЛ по точкам сечения.
    - ПРОДОЛЖАЕМ ЦИКЛ по секущим плоскостям.
- КОНЕЦ.

## Алгоритм ПЕРЕСЕЧЕНИЕ

(предполагается, что пересечение существует)

**Дано:** один из горизонтов – массив  $Gor[MaxX]$ ;  
отрезок  $[r, t]$  на экране ( $X_r \leq X_t$ ).

**Найти:** точку пересечения  $X_0, Y_0$  отрезка с горизонтом (с округлением до пиксела).

- ЕСЛИ отрезок вертикален, ТО полагаем  $X_0 = X_r, Y_0 = Gor(X_0)$ .  
ИНАЧЕ вычисляем коэффициент наклона:  
$$m = (Y_t - Y_r) / (X_t - X_r)$$
  - определяем знак  $S_p$  отклонения от горизонта точки  $p$  :  
$$S_p = \begin{cases} 0, & \text{если } Y_p = Gor(X_p), \\ 1, & \text{если } Y_p > Gor(X_p), \\ -1, & \text{если } Y_p < Gor(X_p). \end{cases}$$
  - ЦИКЛ по точкам  $X, Y$  отрезка  $[r, t]$  с шагом 1 пиксел по оси  $x$  и шагом  $m$  по оси  $y$ .
    - Определяем знак  $S$  отклонения от горизонта точки  $X, Y$  (аналогично  $S_p$ ).
    - ЕСЛИ  $S = S_p$ , ТО продолжаем цикл по точкам отрезка.

- Пересечение обнаружено. Анализируя две соседние точки отрезка, лежащие по разные стороны от горизонта (правую  $r$  и левую  $l$ ):

$$X_r = X, Y_r = Y \quad \text{и} \quad X_l = X - 1, Y_l = Y - m,$$

и две точки горизонта:

$$X_r, \text{Gor}(X_r) \quad \text{и} \quad X_l, \text{Gor}(X_l),$$

найдем точку пересечения  $X_0, Y_0$ , выбрав из левой и правой точек отрезка ту, что менее отклонена от горизонта, т.е.

$$X_0 = X_l, Y_0 = Y_l, \quad \text{если} \quad |Y_l - \text{Gor}(X_l)| \leq |Y_r - \text{Gor}(X_r)|,$$

$$X_0 = X_r, Y_0 = Y_r, \quad \text{в противном случае.}$$

- ЕСЛИ горизонт на интервале  $[X_l, X_r]$  более пологий, чем отрезок, т.е.

$$|m| \leq |g|, \quad \text{где} \quad g = \text{Gor}(X_r) - \text{Gor}(X_l),$$

то скорректируем ординату – будем считать, что пересечение лежит на горизонте:

$$Y_0 = \text{Gor}(X_0).$$

- КОНЕЦ.

**Примечание.** Если быть более точным, то в случае, когда и отрезок, и горизонт не пологи, т.е.  $|m| > 1$  и  $|g| > 1$ , ординату  $Y_0$  надо вычислять по формуле:

$$Y_0 = \frac{\text{Gor}(X_r) * m - Y_r * g}{m - g},$$

которая в частных случаях (при малых  $m$  и  $g$ ) дает результаты, используемые нами в алгоритме.

### Алгоритм КОРРЕКЦИЯ

**Дано:** горизонты  $\text{GorU}[\text{MaxX}]$  и  $\text{GorD}[\text{MaxX}]$ ;

отрезок  $[p, t]$ , заданный целочисленными экранными координатами  $X_p, Y_p$  и  $X_t, Y_t$  его концов ( $X_p \leq X_t$ ).

**Найти:** горизонты, скорректированные отрезком.

- ЕСЛИ отрезок  $[p, t]$  вертикален, ТО корректируем горизонты точками  $p$  или  $t$ :

$$\text{GorU}[X_p] = \max(Y_p, Y_t, \text{GorU}(X_p)),$$

$$\text{GorD}[X_p] = \min(Y_p, Y_t, \text{GorD}(X_p)).$$

ИНАЧЕ

- Вычисляем коэффициент наклона:

$$m = (Y_t - Y_p) / (X_t - X_p).$$

- полагаем  $Y = Y_p, X = X_p$ .

- ЦИКЛ по точкам  $X, Y$  отрезка  $[p, t]$  с шагом пиксел по оси  $x$  и с шагом  $m$  по оси  $y$ .

- Корректируем горизонты очередной точкой  $X, Y$  (если она видна).

- ПРОДОЛЖАЕМ ЦИКЛ по точкам отрезка.

- КОНЕЦ.



## ЗАДАНИЕ

Запрограммируйте один из вариантов алгоритма, придерживаясь следующего порядка действий:

- сосредоточьтесь прежде на простой поверхности (задачи 1–4) и на одном, самом простом сечении ( $z = 0$ ), выберите крупный шаг по  $x$ , окно оцените вручную, игнорируйте повороты и получите правильное изображение выбранного сечения;
- увеличьте количество сечений до трех – пяти и добейтесь правильного удаления невидимого;
- добавьте поворот поверхности;
- добавьте вычисление окна;
- задайте более мелкие шаги по  $x$  и по  $z$ ;
- исследуйте различные поверхности (варьируя их области определения).

### Желающие могут расширить задания:

- использовать алгоритм деления пополам при поиске пересечения отрезка с горизонтом;
- для разложения отрезка в растр реализовать более быстродействующий алгоритм, например, целочисленный алгоритм Брезенхема;
- устранить дефект «боковых зазубрин», иногда проявляющийся при повороте вокруг оси  $y$ , в частности, для поверхности «пропеллер» (задача 6) при  $Dz = 0.1$ ,  $Dx = 0.1$ ,  $Uy = 10$  градусов;
- модернизировать алгоритм, рассматривая вместо одного семейства сечений  $z = \text{const}$  одновременно два семейства:  $z = \text{const}$  и  $x = \text{const}$ .

## ЗАДАЧИ

Знаком \* отмечены номера тех задач, для которых предпочтительны второй и четвертый варианты алгоритма.

- 1\*) «Уголок»  
 $y = |x| + |z|, \quad -5 \leq x \leq 5, \quad -1 \leq z \leq 1.$
- 2) «Птица»  
 $y = |x * z|, \quad -2 \leq x \leq 2, \quad -2 \leq z \leq 0.$
- 3\*) «Седло»  
 $y = x * x - z * z, \quad -2 \leq x \leq 2, \quad -2 \leq z \leq 2.$

- 4\*) «Гамак»  
 $y = x * x + z * z, \quad -2 \leq x \leq 2, \quad -2 \leq z \leq 2.$
- 5\*) «Лента»  
 $y = z * z * z + x * z, \quad -5 \leq x \leq 5, \quad -2 \leq z \leq 2.$
- 6\*) «Пропеллер»  
 $y = x * z * (x + z) * (x - z), \quad -2 \leq x \leq 2, \quad -2 \leq z \leq 2.$
- 7) «Пик»  
 $y = e^{-\sqrt{x * x + z * z}} - 0.5, \quad -2 \leq x \leq 2, \quad -2 \leq z \leq 2.$
- 8) «Волна»  
 $y = \cos \sqrt{x * x + z * z}, \quad -2\pi \leq x \leq 2\pi, \quad -2\pi \leq z \leq 2\pi.$
- 9\*) «Бугор»  
 $y = \sin x * \sin z, \quad 0 \leq x \leq \pi, \quad 0 \leq z \leq \pi.$
- 10) «Волна затухающая»  
 $y = \frac{\cos a}{a + 1}, \quad a = \sqrt{x * x + z * z} \quad -2\pi \leq x \leq 2\pi, \quad -2\pi \leq z \leq 2\pi.$
- 11\*) «Бугор волнистый»  
 $y = \left( \sin x + \frac{\sin(16 * x)}{16} \right) * \sin z, \quad 0 \leq x \leq \pi, \quad 0 \leq z \leq \pi.$
- 12) «Бугры и ямы затухающие»  
 $y = \frac{\cos x * \cos z}{|x| + 1}, \quad -1.5\pi \leq x \leq 1.5\pi, \quad -\pi/2 \leq z \leq \pi/2.$
- 13) «Воронка»  
 $y = -1.5 * \cos(1.75 * a) * e^{-a} + 0.2 * \sin(x + \pi) * \cos(z + \pi),$   
 $a = x * x + z * z, \quad -\pi \leq x \leq \pi, \quad -\pi \leq z \leq \pi.$
- 14) «Две волны затухающие»  
 $y = \frac{\cos a}{a + 1} + \frac{\cos b}{b + 1},$   
 $a = \sqrt{(x + 3\pi) * (x + 3\pi) + z * z},$   
 $b = \sqrt{(x - 3\pi) * (x - 3\pi) + z * z},$   
 $-6\pi \leq x \leq 6\pi, \quad -3\pi \leq z \leq 3\pi.$
- 15) «Поверхность Бохан»  
 $y = -\ln(|x * x * (x * x - z * z)| + 0.15), \quad -5 \leq x \leq 5, \quad -5 \leq z \leq 5.$

## ЛИТЕРАТУРА

1. *Дегтярев В.М., Затыльников В.П.* Инженерная и компьютерная графика : учебник для студентов высших учебных заведений, обучающихся по техническим направлениям, 3-е изд., стер. Москва : Академия, 2012.
2. *Ванишина Е.А., Ларченко Н.В., Шевченко О.Н.* Изображения. Виды. Оренбург : ОГУ, 2014. – 100 с.
3. *Дегтярев В.М.* Компьютерная геометрия и графика : учебник для студентов высших учебных заведений, обучающихся по специальности «Информационные системы и технологии» направления подготовки «Информационные системы», 2-е изд., стер. Москва : Академия, 2011. – 191 с.
4. *Голованов Н.Н., Ильютко Д.П., Носовский Г.В. и др.* Компьютерная геометрия: учебное пособие. Москва : Академия, 2006. – 510 с.
5. *Никулин Е.А.* Компьютерная геометрия и алгоритмы машинной графики : учебное пособие для студентов вузов, обучающихся по направлению 654600 – «Информатика и вычислительная техника». СПб. : БХВ-Петербург, 2005. – 560 с.
6. *Боресков А.В., Шикин Е.В.* Компьютерная графика : учебник и практикум для прикладного бакалавриата. М. : Издательство Юрайт, 2016. – 219 с. – Режим доступа ЭБС Юрайт: <https://www.biblio-online.ru/book/4B1B1827-EB9A-4FF5-8AF1-1CA9159ED4CC>
7. *Катханова Ю.Ф.* Теория компьютерной графики и дизайна [Электронный ресурс] : онлайн курс. // TeachPro.ru : образовательный портал / ООО «МультиМедиа Пабблишинг». М., 2013.  
URL: <http://teachpro.ru/Course/ComputerGraphicAndDesignTheory>
8. *Дёмин А.Ю., Кудинов А.В.* Компьютерная графика [Электронный ресурс] Томск : ТПГ, 2005. URL: <http://compgraph.tpu.ru/index.html>.
9. *Иванов А., Ильютко Д., Носовский Г. и др.* Практикум по компьютерной геометрии [Электронный ресурс] // «ИНТУИТ» Национальный Открытый Университет. М., 2003-2016.  
URL: <http://www.intuit.ru/studies/courses/15818/501/info>

*Издание подготовлено в авторской редакции*

Отпечатано на участке цифровой печати  
Издательского Дома Томского государственного университета

Заказ № 3782 от «14» мая 2019 г. Тираж 50 экз.



