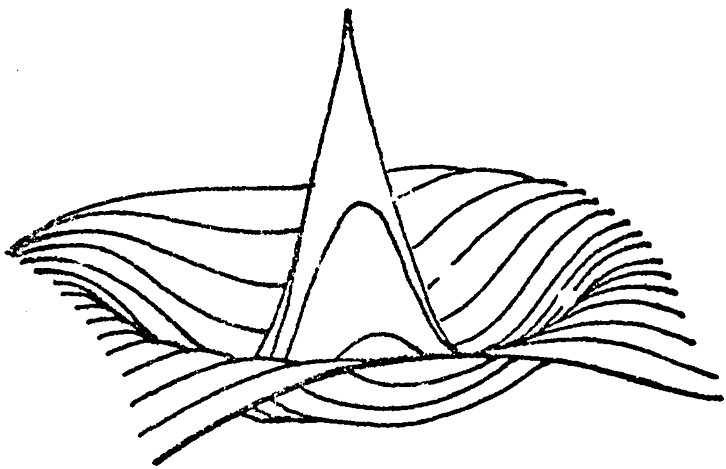
Государственный комитет Российской Федерации
по высшему образованию
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра математической логики и проектирования

УТВЕРЖДАЮ
Декан РФФ
к.ф-м.н __________ С.В.Малянов
"_1_" __ноября__ 1995г.



Лабораторные работы по МАШИННОЙ ГРАФИКЕ часть 3

Томск - 1995

Рассмотрены и утверждены методической комиссией радиофизического факультета.
Председатель комиссии
доцент Дейкова Г.М.Дейкова.

Протокол N 4 от 24.10 1995г.

Предлагаемые методические указания состоят из нескольких частей.

В данной части рассматриваются лабораторные работы по двум темам:

- отсечение отрезков выпуклым окном (алгоритм Кируса - Бека),
- удаление невидимых линий поверхности (метод плавающего горизонта).

Работы ориентированы на студентов, изучающих машинную графику.

Составители:

доцент кафедры МЛиП Быкова С.В.,

старший преподаватель кафедры программирования ФПМК - Белоусова Н.А.

ЛАБОРАТОРНАЯ РАБОТА 2а

ТЕМА: ОТСЕЧЕНИЕ ОТРЕЗКА ПРОИЗВОЛЬНЫМ ВЫПУКЛЫМ ОКНОМ (АЛГОРИТМ КИРУСА - БЕКА).

Условные обозначения

[a,b] - отрезок с концами a,b;
$\underline{a,b}$ - прямая, содержащая отрезок [a,b];
$\overline{ab}$ - вектор из точки a в точку b;
$\underline{c}(Xc,Yc)$ - точка с координатами Xc,Yc;
$\overline{n}$ - вектор внутренней нормали к краю окна;
* - знак скалярного произведения векторов.

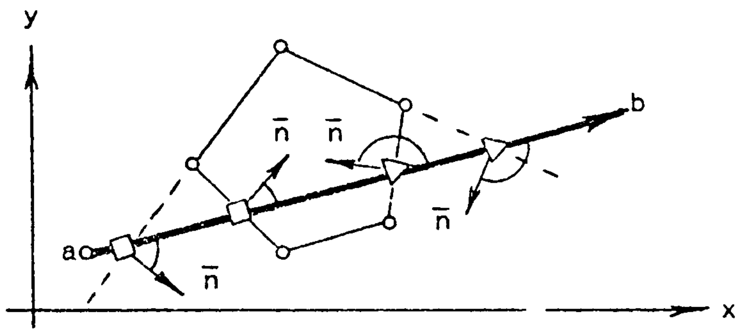


Рис.1. Типы пересечений прямой a,b и краев окна.

Каждый край окна делит плоскость на две полуплоскости: внутреннюю - содержащую окно, и внешнюю.

□ - пересечение типа "ИЗ ВНЕ" - прямая a,b пересекает край или его продолжение, переходя из полуплоскости внешней во внутреннюю: $\overline{ab} * \overline{n} > 0$;

▷ - пересечение типа "ВО ВНЕ" - прямая a,b пересекает край или его продолжение, переходя из полуплоскости внутренней во внешнюю: $\overline{ab} * \overline{n} < 0$.

Если $\overline{ab} * \overline{n} = 0$, то $\overline{ab}$ - точка или вектор, параллельный краю окна.

Параметрическое уравнение прямой a,b

```
x(t) = Xa + (Xb-Xa)*t,    (t=0  для точки a,
y(t) = Ya + (Yb-Ya)*t;     t=1  для точки b).
```

Параметр t для точки пересечения прямой a,b и края окна (или его продолжения)

$$t = - \frac{\overline{wa} * \overline{n}}{\overline{ab} * \overline{n}}$$, где w - вершина края.

Обозначения для параметра t

```
t□    - параметр t для пересечения типа "ИЗ ВНЕ";
t▷    - параметр t для пересечения типа "ВО ВНЕ";
maxt□ - максимальное  из  всех  значений  t□;
mint▷ - минимальное  из  всех  значений  t▷.
```

Параметр t для концов a',b' видимой части отрезка

```
t = max (maxt□, 0)       -  для конца a';
t = min (mint▷, 1)       -  для конца b'.
```

А л г о р и т м   К и р у с а - Б е к а

Д а н о: отрезок [ab] и выпуклое окно, заданное координатами его k вершин (перечисленных по или против часовой стрелки).

Н а й т и: тип видимости VID=0 - отрезок невидим, VID=1 - полностью или частично видим (в этом случае определить координаты концов a', b' видимой части отрезка).

- НАЧАЛО.
- Устанавливаем maxtв = 0, mintн = 1.
- ЦИКЛ по краям окна (для i=1,...,k).

  - Вычисляем внутреннюю нормаль $\overline{n}$ к i-ому краю окна (алгоритм НОРМАЛЬ).
  - Вычисляем скалярное произведение вектора $\overline{wa}$ и внутренней нормали $\overline{n}$ (числитель в формуле для параметра t).
  - Вычисляем скалярное произведение вектора $\overline{ab}$ и внутренней нормали $\overline{n}$ (знаменатель в формуле для параметра t).

  - ЕСЛИ знаменатель = 0,
    ТО $\overline{ab}$ - точка или вектор, параллельный i-ому краю окна:

    - ЕСЛИ числитель < 0,
      ТО вектор $\overline{ab}$ лежит вне окна за пределами i-го края (рис.2):
      устанавливаем признак видимости
      VID = 0 и идем на КОНЕЦ;
      ИНАЧЕ вектор $\overline{ab}$ лежит по ту же сторону от края, что и окно, значит может быть видим (рис.3), поэтому ПРОДОЛЖАЕМ ЦИКЛ по краям окна;

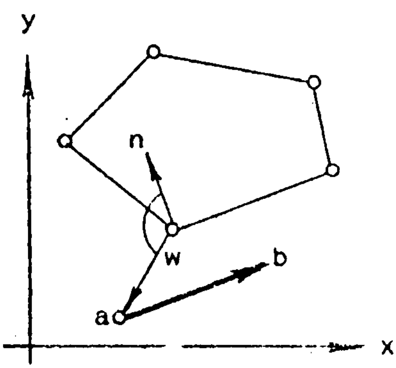


Рис.2. $\overline{ab}*\overline{n}=0$, $\overline{wa}*\overline{n}<0$.

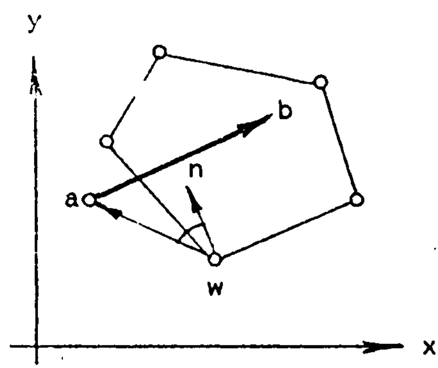


Рис.3. $\overline{ab}*\overline{n}=0$, $\overline{wa}*\overline{n}>=0$.

ИНАЧЕ (знаменатель =/= 0) прямая a,b пересекает i-ый край или его продолжение:

- вычисляем значение параметра t для точки пересечения:

$$t = - \frac{\overline{wa} * \overline{n}}{\overline{ab} * \overline{n}} ;$$

- ЕСЛИ ~~числитель <~~ знаменатель > 0,

  ТО тип пересечения □ - "ИЗ ВНЕ":
  - ЕСЛИ t□ > 1,
    ТО вектор $\overline{ab}$ кончается до пересечения прямой a,b с краем (рис.4): полагаем VID = 0 и идем на КОНЕЦ;
    ИНАЧЕ корректируем maxt□ :
    maxt□ = max (t□, maxt□);

  ИНАЧЕ тип пересечения ▷ - "ВО ВНЕ":
  - ЕСЛИ t▷ < 0,
    ТО вектор $\overline{ab}$ начинается после пересечения прямой a,b с краем (рис.5): полагаем VID = 0 и идем на КОНЕЦ;
    ИНАЧЕ корректируем mint▷ :
    mint▷ = min (t▷, mint▷).

- ПРОДОЛЖАЕМ ЦИКЛ по краям окна.

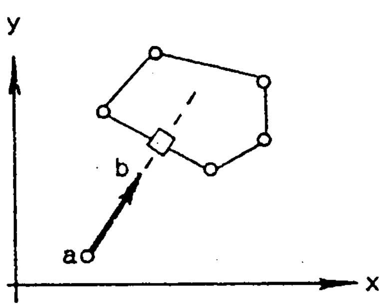


Рис.4. t□ > 1.

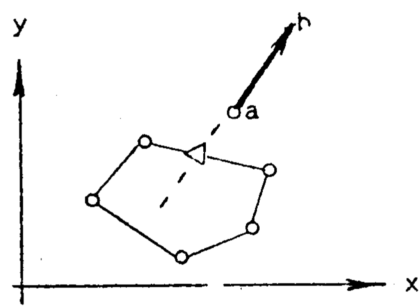


Рис.5. t▷ < 0.

- Анализируем значения параметров maxt□ и mint▷:

ЕСЛИ maxt□ > mint▷,
ТО отрезок невидим (рис.6), полагаем VID = 0;

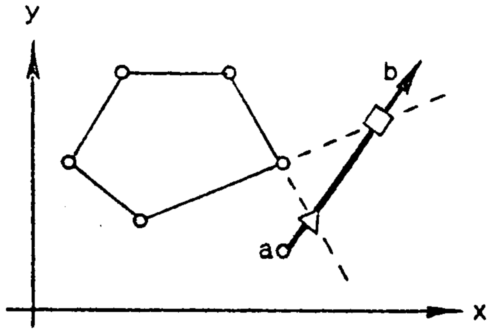


Рис.6. maxt□ > mint▷.

ИНАЧЕ полагаем VID=1 (рис.1,7) и, исходя из значений maxt□ и mint▷, находим концы a',b' отсеченного отрезка по параметрическому уравнению прямой.

- КОНЕЦ.

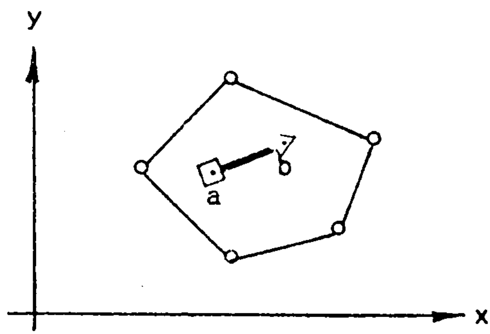


Рис.7. maxt□ <= mint▷.

П р и м е ч а н и е. Факт невидимости отрезка (рис.6) можно установить раньше, если каждый раз после коррекции значений maxt□ и mint▷ проверять условие maxt□ > mint▷.

А л г о р и т м НОРМАЛЬ

Д а н о: $\overline{wu}$ - край окна, v - любая вершина окна, отличная от w и u .
Н а й т и: $\overline{n}$ - вектор внутренней нормали к краю $\overline{wu}$.

- НАЧАЛО.
- Вычисляем компоненты вектора $\overline{wu}$:

  wuX = Xu - Xw, wuY = Yu - Yw.
- Из равенства треугольников (рис.8) вычисляем компоненты вектора нормали:

  nX = wuY, nY = - wuX

  (длина нормали в алгоритме Кируса-Бека произвольна, поэтому полагаем ее равной длине вектора $\overline{wu}$).

- Проверяем, является ли нормаль внутренней:

  ЕСЛИ вершина v и нормаль $\overline{n}$ лежат по разные стороны от края $\overline{wu}$, т.е. $\overline{n}$ * $\overline{wv}$ < 0,

  ТО нормаль внешняя, поэтому меняем знаки:

  nX = -nX, nY = -nY.
- КОНЕЦ.

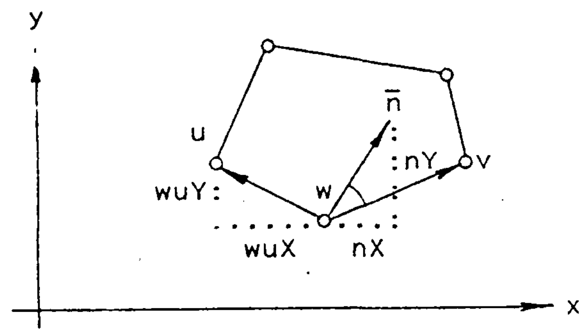
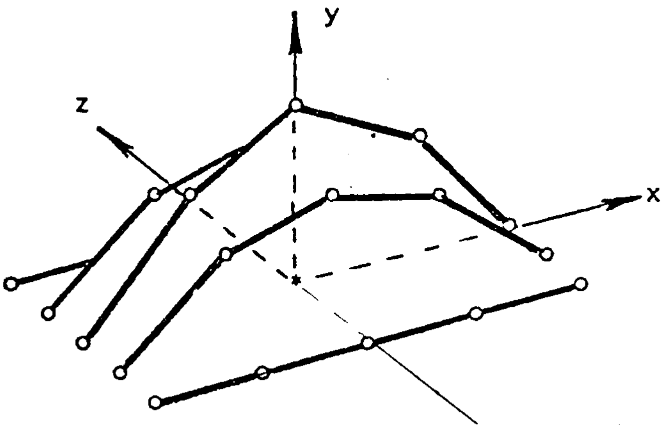


Рис.8. Нормаль $\overline{n}$ - внутренняя: $\overline{n}$ * $\overline{wv}$ > 0.

П р и м е ч а н и е. Если вершины w,u перечислены в порядке обхода краев окна по часовой стрелке, то вычисленная нормаль всегда внутренняя, поэтому можно опустить проверку.

ЛАБОРАТОРНАЯ РАБОТА 5

ТЕМА: УДАЛЕНИЕ НЕВИДИМЫХ ЛИНИЙ ПОВЕРХНОСТИ МЕТОДОМ ПЛАВАЮЩЕГО ГОРИЗОНТА



Условные обозначения

y = f(x,z) - поверхность с областью определения
Xmin <= x <= Xmax, Zmin <= z <= Zmax;

z = Zmin ,..., Zmax - семейство секущих плоскостей;
Dz - шаг по z (между секущими плоскостями);
Dx - шаг по x (между точками сечения).

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

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

Сечением назовем линию пересечения поверхности и секущей плоскости. Экранным сечением - проекцию сечения (повернутого на углы Uy и Ux) на экран.

Верхним (нижним) горизонтом для данного экранного сечения будем считать огибающую сверху (снизу) всех предыдущих экранных сечений.

Точка Хэ,Yэ экранного сечения видна, если лежит выше верхнего или ниже нижнего горизонтов для данного сечения:

Yэ >= UP(Хэ) или Yэ <= DOWN(Хэ),

где UP[MaxX] и DOWN[MaxX] - массивы горизонтов.

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.

Вариант 1: точками с поворотом вокруг оси x;
вариант 2: точками с поворотом вокруг осей y,x;
вариант 3: отрезками с поворотом вокруг оси x;
вариант 4: отрезками с поворотом вокруг осей y,x.

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

О б щ а я   ч а с т ь   в с е х   в а р и а н т о в

Д а н о: функция $y=f(x,z)$, Xmin,Xmax, Zmin,Zmax;
шаги Dx,Dz; углы поворота Uy,Ux;
поле вывода Cx,Cy, Px,Py.

- 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 назовем старыми;NewUP,NewDOWN - новыми).

- ЦИКЛ по секущим плоскостям для Zmin <= Z <= Zmax с шагом Dz.

  - ЦИКЛ по точкам сечения для Xmin <= X <= Xmax с шагом Dx.

    - Вычисляем значение функции $Y = f(X,Z)$.
    - Повернув точку X,Y,Z, имеем ее проекцию Xк,Yк на картинную плоскость.
    - Учтя размеры окна и поля вывода, вычисляем экранные координаты точки Xэ,Yэ.
    - Формируем признак ее видимости, используя старые горизонты (UP и DOWN):
      VID=0 - точка не видна;
      VID=1 - видна над верхним горизонтом;
      VID=2 - видна под нижним горизонтом;
      VID=3 - видна над верхним и
      под нижним горизонтами.

    - В с т а в к а для вариантов 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 ее точки, лежащие над верхним горизонтом, нарисованы одним цветом; точки, лежащие под нижним горизонтом - другим цветом; а точки первого сечения - третьим цветом;

- в-третьих, вычисление значения функции y=f(x,z) можно оформить в виде подпрограммы, тогда алгоритм станет универсальным (ориентирорaнным на любую функцию).

В с т а в к а для п е р в о г о варианта

```
...
   - ЕСЛИ точка видна,
     ТО   рисуем ее и корректируем горизонты
          UP и DOWN в данной точке.
...
```

В с т а в к а для в т о р о г о варианта

```
...
   - Обозначим текущую точку Хэ,Yэ через t,
     а ее признак видимости  - через  VIDt.

   - ЕСЛИ точка t видна,
     ТО   рисуем ее.

   - 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,  VIDp = VIDt.
...
```

В с т а в к а   для   т р е т ь е г о   варианта

Возможные в этом варианте ситуации сведены в таблицу, изображенную на стр. 14.

...

- Обозначим текущую точку Хэ,Yэ через t,
  а ее признак видимости - через VIDt.

- ЕСЛИ точка t не первая в сечении,
  ТО анализируем видимость отрезка [p,t] по признакам видимости VIDp и VIDt его концов:

  - ЕСЛИ видна хотя бы одна из точек предыдущая (p) или текущая (t),
    ТО виден весь или часть отрезка:

    - ЕСЛИ виден весь 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].

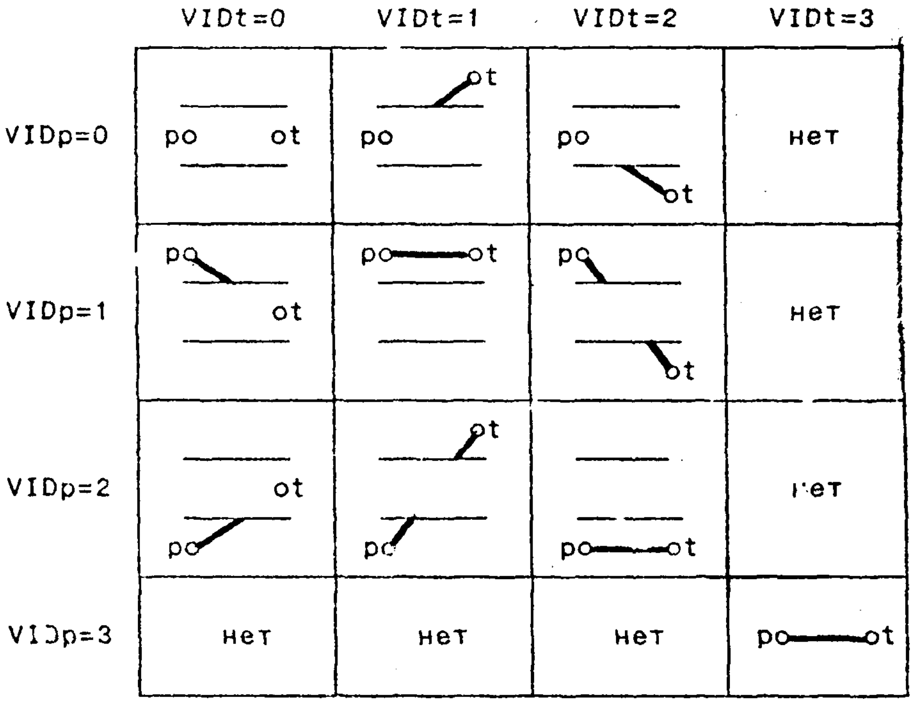
      - ЕСЛИ виден конец t,
        ТО, найдя точку пересечения q отрезка с горизонтом (алгоритм ПЕРЕСЕЧЕНИЕ), рисуем отрезок [q,t];

    - корректируем новые горизонты NewUP и NewDOWN отрезком [p,t] (алгоритм КОРРЕКЦИЯ).

- Запоминаем текущую точку в качестве предыдущей, полагая: p = t, VIDp = VIDt.

...

ТАБЛИЦА возможных ситуаций для третьего варианта:



В с т а в к а для ч е т в е р т о г о варианта

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 Хэ,Yэ через t, а ее признак видимости через VIDt.

- ЕСЛИ точка t не первая в сечении,
  ТО анализируем видимость отрезка [p,t] по признакам видимости VIDp и VIDt 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=Xt-Xp и YY=Yt-Yp:

    - ЕСЛИ /YY/ <= /XX/
      ТО dy = YY:XX, dx = 1;
      ИНАЧЕ dx = /XX:YY/,
      dy = 1, если Yt > Yp,
      dy = -1, если Yt < Yp.

    - ЦИКЛ по точкам отрезка [p,t] от p к t с шагом dx,dy.

      - Округляя координаты очередной точки, получаем пиксел растра: X,Y.

      - ЕСЛИ пиксел X,Y виден по отношению к старым горизонтам UP и DOWN,
        ТО рисуем пиксел и корректируем новые горизонты NewUP и NewDOWN.

    - ПРОДОЛЖАЕМ ЦИКЛ по точкам отрезка.

- Запоминаем текущую точку в качестве предыдущей, полагая p = t, VIDp = VIDt.

...

Алгоритм ОКНО

Дано: функция y=f(x,z), Xmin,Xmax,Zmin,Zmax;
шаги Dx,Dz; углы поворота Uy,Ux.
Найти: полуразмеры окна Pkx и Pky.

- Вычисляем картинные абсциссы X1, X2, X3, X4 угловых точек области определения:
  Xmin,Zmin; Xmax,Zmin; Xmin,Zmax; Xmax,Zmax;
  выполнив поворот этих точек вокруг оси y.

- Определяем Pkx = max (/X1/,/X2/,/X3/,/X4/).

- Определяем Pky, положив первоначально Pky=0 и перебирая значения функции:

- ЦИКЛ по секущим плоскостям Z
  для Zmin <= Z <= Zmax с шагом Dz.

  - ЦИКЛ по точкам X,Y сечения Z
    для Xmin <= X <= Xmax с шагом Dx.

    - Вычисляем значение функции Y=f(X,Z).
    - Повернув точку X,Y,Z на углы Uy,Ux, имеем ее картинную ординату Yk, которой корректируем Pky = max(Pky,/Yk/).

  - 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 [p,t] на экране (Xp <= Xt).
Найти: точку пересечения Xo,Yo отрезка с горизонтом (с округлением до пиксела).

- ЕСЛИ отрезок вертикален,
  ТО полагаем Xo = Xp, Yo = GOR(Xo).

  ИНАЧЕ вычисляем коэффициент наклона:
  m = (Yt - Yp) : (Xt - Xp);

  - определяем знак Sp отклонения от горизонта точки p:

$$Sp = \begin{cases} 0, & \text{если } Yp = GOR(Xp), \\ 1, & \text{если } Yp > GOR(Xp), \\ -1, & \text{если } Yp < GOR(Xp); \end{cases}$$

  - ЦИКЛ по точкам X,Y отрезка [p,t] с шагом 1 пиксел по оси x и шагом m по оси y.

    - Определяем знак S отклонения от горизонта точки X,Y (аналогично Sp).
    - ЕСЛИ S = Sp, ТО

  - ПРОДОЛЖАЕМ ЦИКЛ по точкам отрезка.

  - Пересечение обнаружено. Анализируя две соседние точки отрезка, лежащие по разные стороны от горизонта (правую r и левую l):
    Xr=X,Yr=Y и Xl=X-1,Yl=Y-m,
    и две точки горизонта:
    Xr,GOR(Xr) и Xl,GOR(Xl),
    найдем точку пересечения Xo,Yo, выбрав из левой и правой точек отрезка ту, что менее отклонена от горизонта, т.е.

    Xo=Xl, Yo=]Yl[, если /Yl-Gor(Xl)/<=
    <= /Yr-Gor(Xr)/,
    Xo=Xr, Yo=]Yr[, в противном случае.

  - ЕСЛИ горизонт на интервале [Xl,Xr] более пологий, чем отрезок, т.е.
    /m/ <= /g/, где g = GOR(Xr) - GOR(Xl),
    ТО скорректируем ординату - будем считать, что пересечение лежит на горизонте:
    Yo = GOR(Xo).

- КОНЕЦ.

Если быть более точным, то в случае, когда и отрезок, и горизонт не пологи, т.е. /m/>1 и /g/>1, ординату Yo надо вычислять по формуле:

$$Yo = \left] \frac{GOR(Xr) * m - Yr * g}{m - g} \right[,$$

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

А л г о р и т м КОРРЕКЦИЯ

Д а н о: горизонты GORu[MaxX] и GORd[MaxX]; отрезок [p,t], заданный целочисленными экранными координатами Xp,Yp и Xt,Yt его концов (Xp <= Xt).

Н а й т и: горизонты, откорректированные отрезком.

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

  GORu(Xp) = max (Yp, Yt, GORu(Xp)),
  GORd(Xp) = min (Yp, Yt, GORd(Xp)).

  ИНАЧЕ
  - Вычисляем коэффициент наклона:
    m = (Yt - Yp) : (Xt - Xp).
  - Полагаем Y = Yp, X = Xp.
  - 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 задание:

- 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 $z = const$ одновременно два семейства: $z = const$ и $x = const$.

ЗАДАЧИ

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

```
-1   ----------------- "Уголок" ------------------
                                      -5 <= x <= 5,
     y = /x/ + /z/;                   -1 <= z <= 1.

 2   ----------------- "Птица" -------------------
                                      -2 <= x <= 2,
     y = /x * z/;                     -2 <= z <= 0.

-3   ----------------- "Седло" -------------------
                                      -2 <= x <= 2,
     y = x*x - z*z;                   -2 <= z <= 2.

-4   ----------------- "Гамак" -------------------
                                      -2 <= x <= 2,
     y = x*x + z*z;                   -2 <= z <= 2.

-5   ----------------- "Лента" -------------------
                                      -5 <= x <= 5,
     y = z*z*z + x*z;                 -2 <= z <= 2.

-6   --------------- "Пропеллер" -----------------
                                      -2 <= x <= 2,
     y = x*z * (x+z) * (x-z);         -2 <= z <= 2.

 7   ------------------ "Пик" --------------------
             ______________
         - \/ x*x + z*z               -2 <= x <= 2,
     y = e              - 0.5;        -2 <= z <= 2.

 8   ----------------- "Волна" -------------------
                  _____________    -2Pi <= x <= 2Pi,
     y = cos \/ x*x + z*z;         -2Pi <= z <= 2Pi.

-9   ----------------- "Бугор" -------------------
                                      0 <= x <= Pi,
     y = sin x * sin z;               0 <= z <= Pi.
```

```
10 ----------- "Волна затухающая" -- -------------

    y = (cos a) / (a + 1);
                 ___________         -2Pi <= x <= 2Pi,
        a = \/  x*x + z*z;           -2Pi <= z <= 2Pi.

-11 ----------- "Бугор волнистый" -----------------

                     sin (16*x)          0 <= x <= Pi,
    y = (sin x + ----------) * sin z;
                        16               0 <= z <= Pi.

12 -------- "Бугры и ямы затухающие" ---------- ---

        cos x * cos z
    y = -------------;        -1.5 Pi <= x <= 1.5 Pi,
          /x/ + 1               -Pi/2 <= z <= Pi/2.

13 ---------------- "Воронка" ---------------- ---
                                    -a
    y = - 1.5 * cos(1.75 * a) * e    +
        + 0.2 * sin (x+Pi) * cos (z+Pi);

        a = x*x + z*z;                   -Pi <= x <= Pi,
                                         -Pi <= z <= Pi.

14 ----------- "Две волны затухающие" ------------

        cos a      cos b
    y = -----  +  -----;          -6Pi <= x <= 6Pi,
        a + 1      b + 1          -3Pi <= z <= 3Pi;
             _________________________
    a =  \/  (x+3Pi)*(x+3Pi) + z*z,
             _________________________
    b = \/  (x-3Pi)*(x-3Pi) + z*z.

15 ------------"Поверхность Бохан"----------------

    y  = - ln (/x*x * (x*x-z*z)/ + 0.15);

                                            -1 <= x <=1,
                                            -1 <= z <=1.
```

ЛИТЕРАТУРА

1. Д.Роджерс. Алгоритмические основы машинной графики. - М.: Мир, 1989.

2. Е.В.Шикин, А.В.Боресков. Компьютерная графика. - М.: Диалог-МИФИ, 1995.

3. Е.В.Шикин, А.В.Боресков, А.А.Зайцев. Начала компьютерной графики. - М.: Диалог-МИФИ, 1993.

Лабораторная работа 2а: С. 166-180 [1].

Лабораторная работа 5: С. 233-250 [1],
С. 135-144 [2],
С. 76- 80 [3].

ЛАБОРАТОРНЫЕ РАБОТЫ ПО МАШИННОЙ ГРАФИКЕ, ЧАСТЬ 3 /Томский государственный университет - Томск, 1995 - 22 с.

Подп. в печать 9.XI.1995. Тираж 100 экз. Бесплатно

Заказ N 276 УОП ТГУ, Томск, Никитина,4