

Компьютерная графика

Лекция 1

Основные понятия компьютерной графики

Основные цели курса

- Изучение основ вычислительной геометрии и компьютерной графики
- Освоение основных алгоритмов растровой графики
- Изучение вопросов проектирования и пространственного преобразования изображений
- Основы трехмерной графики

Не будут рассматриваться такие темы как:

- Применение графических пакетов и программ обработки изображений
- Особенности конкретных графических библиотек для определенных языков программирования
- Вопросы аппаратных архитектур, систем передачи данных и т.п., имеющих отношение к графике

Типы и виды компьютерной графики

- Деловая графика
- Научная графика, визуализация результатов численных экспериментов.
- Иллюстративная графика (графические редакторы, средства просмотра)
- Инженерная графика
- Создание реалистичных изображений
- Анимация и обработка видеоизображений
- Распознавание изображений
- Виртуальная реальность

Растровая и векторная графика

На аппаратном уровне

- Различают растровые и векторные устройства формирования изображений
- Растровые устройства – например мониторы, поскольку изображение строится из отдельных точек (пикселей) – раstra.
- Векторные устройства – например осциллографы, графопостроители поскольку там нет отдельных точек, а изображение состоит из линий, окружностей и других графических примитивов

На программном уровне

- Несмотря на то, что используются растровые мониторы, изображение можно хранить и обрабатывать либо как набор пикселей на экране, либо как информацию об объекте в целом.
- Например, окружность можно хранить не как набор координат точек, а в виде трех чисел: две координаты центра окружности и ее радиус.

Сравнение растровой и векторной графики

Преимущества растрового подхода

- Возможность изображать реальные объекты сколь угодно сложной формы
- Высокая скорость рисования

Недостатки растра

- Плохая масштабируемость, погрешности при поворотах
- Большой объем информации при хранении
- Сложности последующей обработки построенных изображений

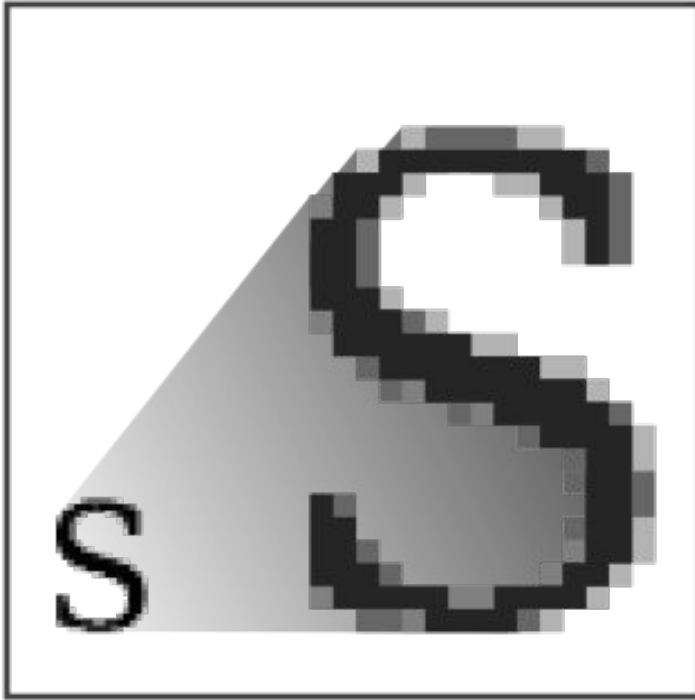
Преимущества векторного подхода

- Компактность информации при хранении
- Возможность делать любые преобразования пространства, в том числе повороты и масштабируемость
- Возможность группировки объектов
- Совместимость с устройствами вывода любого разрешения

Недостатки векторной графики

- Замедленная скорость рисования
- Проблемы с изображением реальных объектов

Отличия растровой и векторной графики



РАСТР
.jpeg .gif .png



ВЕКТОР
.svg

Каждый графический режим характеризуется:

- Разрешающей способностью экрана (разрешением), т.е. количеством точек (пикселей) по горизонтали и вертикали.
- Количеством цветов
 - Активных
 - В палитре
- Количеством видеостраниц.
- Степенью сжатия изображения по координатным осям (Aspect Ratio). Его необходимо учитывать при построении квадратов и окружностей «вручную», иначе получатся прямоугольники и эллипсы.
- Напомним, что система координат экрана имеет начало в левом верхнем углу и ось *y* направлена вниз.

Построение графиков функций в декартовой системе координат

- Требуется построить график функции $y=f(x)$, непрерывной на отрезке $[a, b]$.
- Пусть имеется экран (окно) с разрешением $max_x * max_y$.
- Требуется обеспечить автоматическое масштабирование.
- Различают два способа решения вопроса масштабирования.
 - Независимое по осям (рис. 1, и 2)
 - Пропорциональное (рис. 3)
- Мы будем рассматривать независимое масштабирование

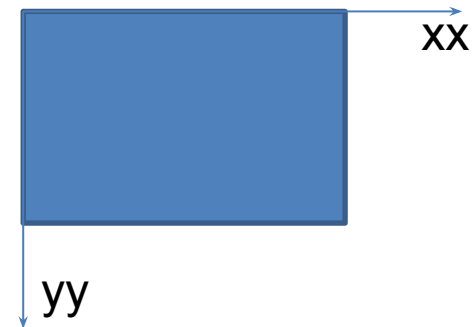


Рис. 1



Рис. 2



Рис. 3

Организация построений непрерывной линии

Как НЕ надо делать

- Разбивать область определения функции, т.е. отрезок $[a, b]$ на фиксированное число N частей.
- Для каждого $x \in [a, b]$ вычислить $y=f(x)$
- Масштабировать, т.е. найти по x экранную координату xx , и аналогично yy по y .
- Построить на экране очередную точку с координатами (xx, yy) .

Рекомендуемый способ

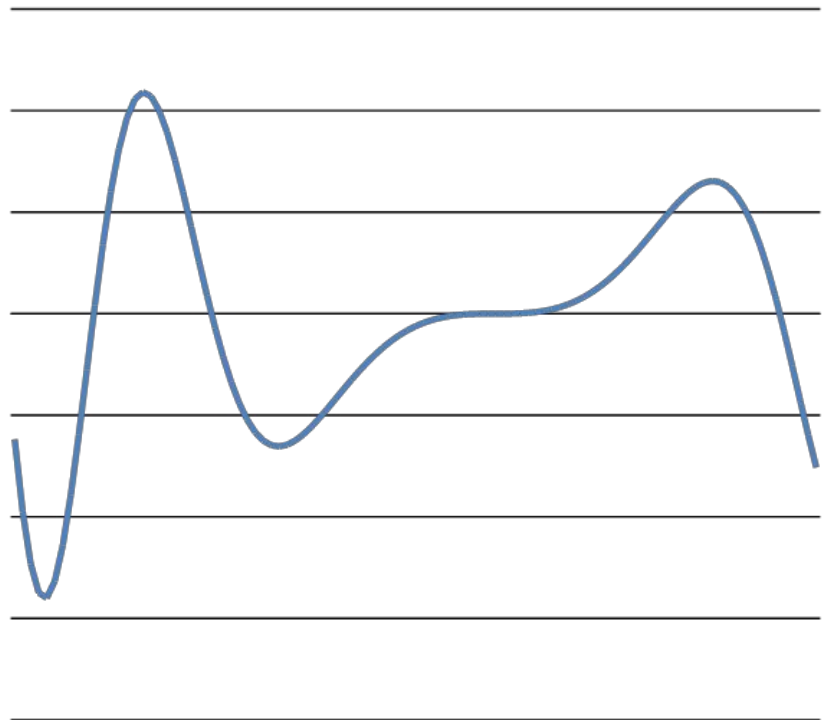
- Рассмотреть каждый пиксел экрана по горизонтали, т.е.
`for (int xx=0;xx<maxx;++xx)`
- Для координаты xx вычислить $x \in [a, b]$ по формуле
 $x=a+xx*(b-a)/maxx$;
- Вычислить $y=f(x)$
- Масштабировать, т.е. найти по y экранную координату yy .
- Построить на экране линию от текущей точки до новой точки (xx, yy) .

Программа построения графика

- Прежде чем писать цикл, рисующий график, для последующего масштабирования по y необходимо вычислить максимум и минимум функции на нашем отрезке $[a,b]$.
- Это такой же цикл, но вместо момента рисования – 2 оператора if.

```
ymin=ymax=f(a);
for (int xx=0;xx<maxx;++xx)
{
    x=a+xx*(b-a)/maxx;
    y=f(x);
    if (y<ymin) ymin=y;
    if (y>ymax) ymax=y;
}
yy=(f(a)-ymin)*maxy/(ymax-ymin);
Form1->Canvas->MoveTo(0,yy);
for (int xx=1;xx<maxx;++xx)
{
    x=a+xx*(b-a)/maxx;
    y=f(x);
    yy=(y-ymin)*maxy/(ymax-ymin);
    Form1->Canvas->LineTo(xx,yy);
}
```

$$y=x*\sin(x*x)$$



Модификация программы в случае пропорционального масштабирования

- Первый цикл, где вычисляется u_{\min} – минимальное значение функции и u_{\max} – максимальное соответственно, остается без изменений.
- Вычисляем коэффициенты сжатия изображения по осям координат. Пусть мы получили dx и dy соответственно.
- Возможны 2 случая: либо график будет максимально растянут по горизонтали, либо по вертикали. Тогда xx и yy вычисляются по формулам:
 - 1) $xx = (x - a) * \max_x / (b - a);$
 $yy = (y - y_{\max}) * \max_x / (b - a) * dy / dx;$
 - 2) $xx = (x - a) * \max_y / (y_{\min} - y_{\max}) * dx / dy;$
 $yy = (y - y_{\max}) * \max_y / (y_{\min} - y_{\max});$

Полярная система координат

- Кроме прямоугольной декартовой системы можно работать с полярными координатами.
- Точка там определяется парой чисел: угол φ и радиус r .
- Преобразования из полярных координат в декартовые производятся по формулам:
- $x = r \cdot \cos(\varphi)$;
- $y = r \cdot \sin(\varphi)$;
- Основная проблема при построении графика функции в полярных координатах – непригодность равномерного разбиения отрезка по углу.

Проблемы с полярными координатами

Если мы фиксируем количество разбиений области определения функции, т.е. величину шага по углам, то возможна одна из следующих ситуаций

Малое число разбиений

- Точки могут находиться на слишком большом расстоянии друг от друга.
- Возможно будут пропущены некоторые особенности графика, наблюдающиеся на малых изменениях углов.

Большое число разбиений

- Много точек после всех вычислений и масштабирования могут оказаться в одном пикселе экрана.
- Получим низкую скорость работы программы.

Алгоритм с динамическим изменением шага

- Решение может быть в динамическом (т.е. меняющемся в процессе работы программы) шаге по углу.
- Выбираем некоторое пороговое значение расстояния в пикселах. Назовем его R . Как правило берут $R=10$.
- Помним координаты предыдущей построенной на экране точки.
- Вычисляем следующую точку, меняя угол с текущим шагом dU .
- Если расстояние от новой (пробной) точки до старой больше R , уменьшаем шаг, например, $dU=dU/a$;
- Если новая точка совпадает со старой, увеличиваем шаг, например, $dU=dU*b$;
- Числа a и b лучше брать взаимно простыми, например, $a=3$, $b=2$
- Если изменять шаг не пришлось, рисуем новую точку и продолжаем построения в цикле.

ЗАДАЧИ

- В декартовой прямоугольной системе координат нарисовать график функции $y=f(x, a_1, a_2, \dots)$ (вид аналитической зависимости определяет преподаватель), заданной на отрезке $[a, b]$. Для вывода графика функции задана прямоугольная область пиксельными координатами левого верхнего и правого нижнего углов. При изменении размеров этой области, а также расположения ее на экране монитора во время выполнения программы изображение графика должно быть верным. Должны быть предусмотрены окна для ввода параметров – a, b, a_1, a_2, \dots . Возможные асимптоты можно не рисовать.
- Аналогичная задача для построения графика функции в полярной системе координат.

ТЕСТ

- При использовании растровой графики
 - a. **Изображение строится из отдельных точек**
 - b. Изображение состоит из графических примитивов
 - c. Изображение использует информацию об объекте в целом
- Преимущества векторной графики над растровой заключаются в
 - a. Высокой скорости рисования
 - b. Возможности изображать объекты сложной формы
 - c. **Совместимости с устройствами вывода любого разрешения**
 - d. В возможности хранения большого объема информации
- При построении непрерывной линии рекомендуется
 - a. Разбивать область определения функции на фиксированное число частей
 - b. Масштабировать, т.е. найти по x экранную координату xx , и аналогично yy по y
 - c. **Для координаты xx вычислять $x \in [a, b]$ по формуле $x = a + xx * (b - a) / \max x$**
- При построении графика функции в полярной системе координат необходимо
 - b. Взять малое число разбиений области определения функции
 - c. Взять большое число разбиений области определения функции
 - d. Выбрать равномерное разбиение области определения функции
 - e. **Выбрать динамический, т.е. меняющийся в процессе работы программы, шаг по углу**

ТЕСТ

- Графический режим нельзя характеризовать
 - a. Количеством цветов
 - b. **Размером экрана**
 - c. Количеством видеостраниц
 - d. Степенью сжатия изображения по координатным осям