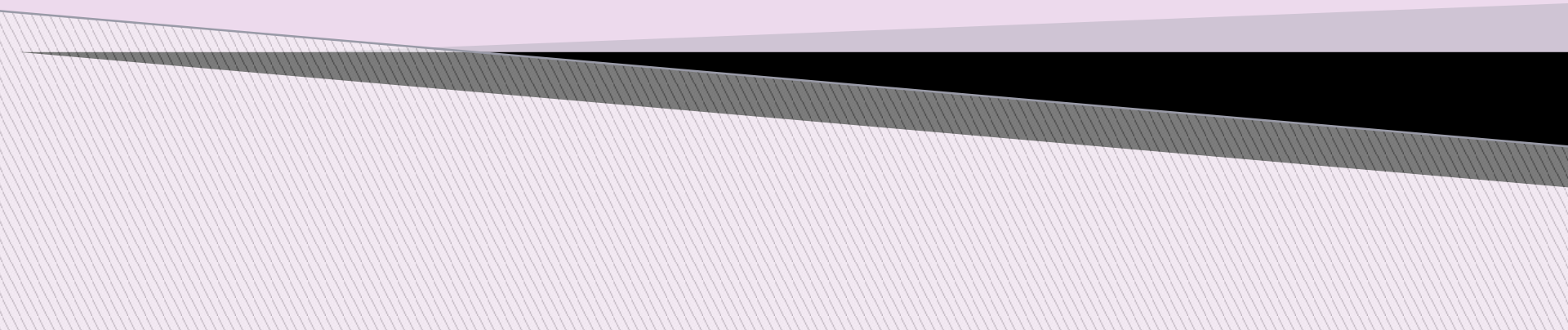


# Фрактальная графика



# Понятие фрактала

- **Фрактáл** (лат. *fractus* — дроблённый, сломанный, разбитый) — графический объект, отдельные элементы которого наследуют свойства родительских структур.
- 1977 г. - выход книги **Бенуа Мандельброта** «*The Fractal Geometry of Nature*» («Фрактальная геометрия природы»). В его работах использованы научные результаты других ученых, работавших в период 1875-1925 годов в той же области (Пуанкаре, Фату, Жюлиа, Кантор, Хаусдорф).
- «**Фракталом называется структура, состоящая из частей, которые в каком-то смысле подобны целому**» (Б. Мандельброт, математический аналитик IBM)

# Понятие фрактала

В самом простом случае небольшая часть фрактала содержит информацию о всём фрактале.

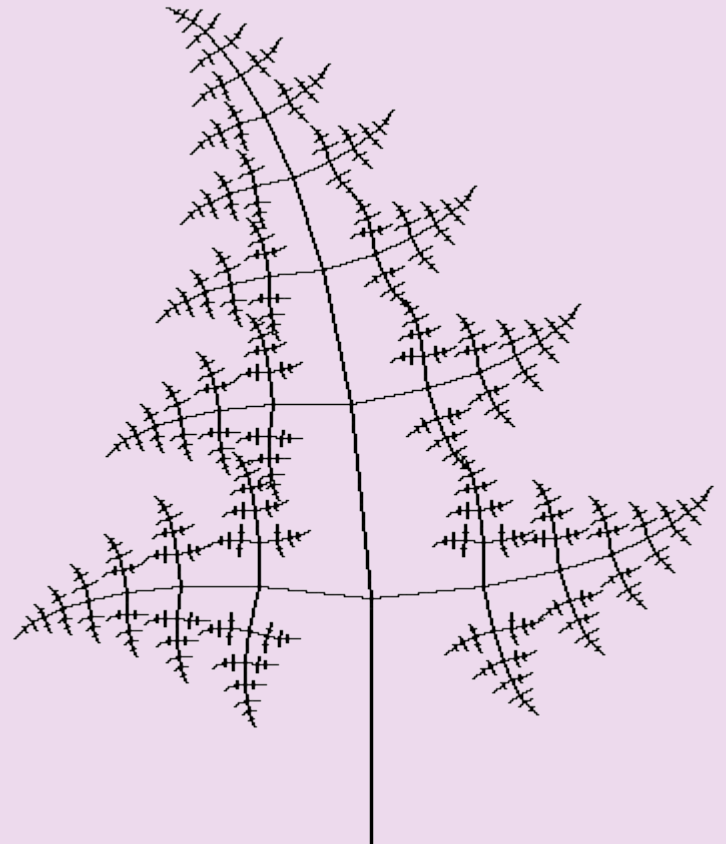
Фрактальная графика незаменима при генерации сложных неевклидовых объектов, образы которых весьма похожи на природные.

Один из наиболее и знаменитых фрактальных изображений - папоротник, в котором каждый лист в действительности представляет собой миниатюрный вариант самого папоротника



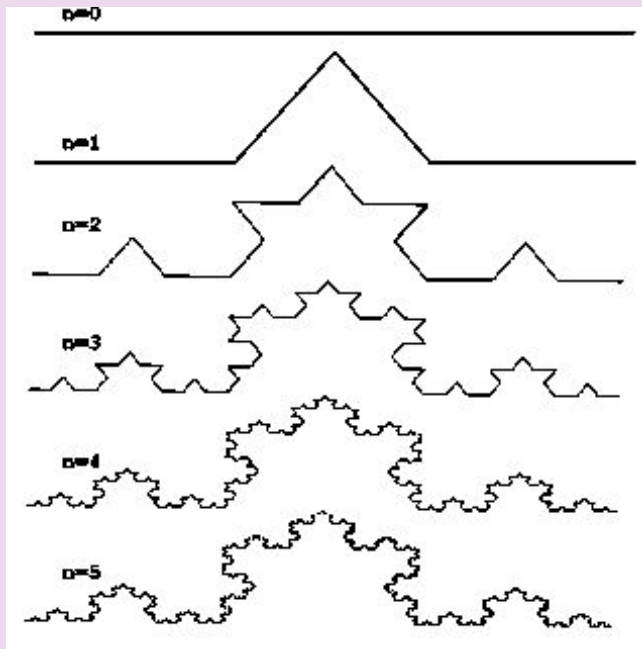
# Пример Pascal-программы для рисования фрактального папоротника

```
program Fern_Constr;  
uses Graph, CRT;  
const min = 1;  
var gd, gm : Integer;  
procedure lineto1(x, y : Integer; l, u : real);  
begin  
Line(x, y, Round(x + l * cos(u)), Round(y - l * sin(u)));  
end;  
procedure Draw(x, y : Integer; l, u : real);  
begin  
if KeyPressed then exit;  
if l > min then begin lineto1(x, y, l, u);  
x := Round(x + l * cos(u));  
y := Round(y - l * sin(u));  
Draw(x, y, l*0.4, u - 14*pi/30);  
Draw(x, y, l*0.4, u + 14*pi/30);  
Draw(x, y, l*0.7, u + pi/30);  
end;  
end;  
begin gd := Detect;  
InitGraph(gd, gm, 'c:\bp\bgi');  
Draw(320, 460, 140, pi/2);  
ReadKey; CloseGraph;  
end.
```

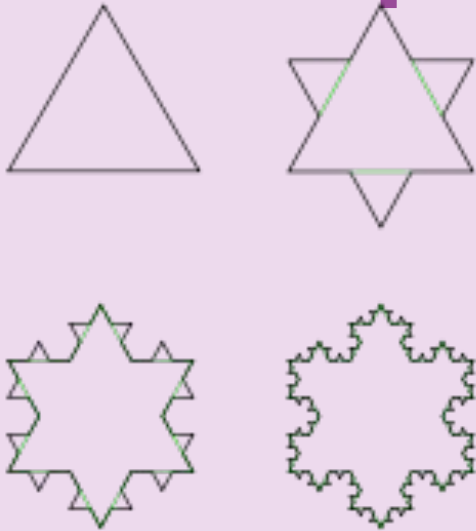


# Геометрические фракталы

- **Геометрические фракталы** - самый наглядный вид фракталов. Получают с помощью некоторой ломаной - генератора. За один шаг алгоритма каждый из отрезков, составляющих ломаную, заменяется на ломаную-генератор, в соответствующем масштабе.



# Геометрические фракталы

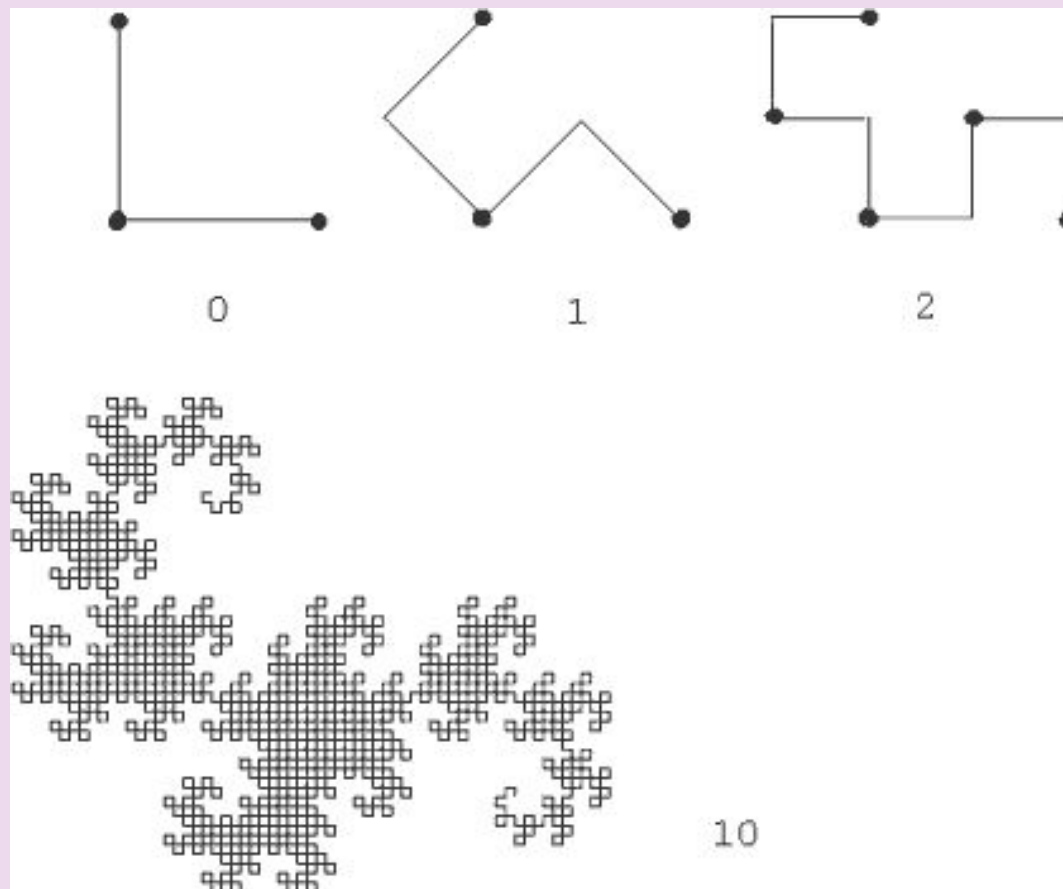


- **Фрактальная снежинка (снежинка Коха)** - один из самых известных геометрических объектов, описан **Хельгой фон Кох** в начале XX века.

Процесс её построения :

- берём единичный отрезок, делим на три равные части и заменяем средний интервал равносторонним треугольником без этого сегмента. В результате образуется ломаная, состоящая из четырех звеньев длины  $1/3$ .
- На следующем шаге повторяем операцию для каждого из четырёх получившихся звеньев и т. д

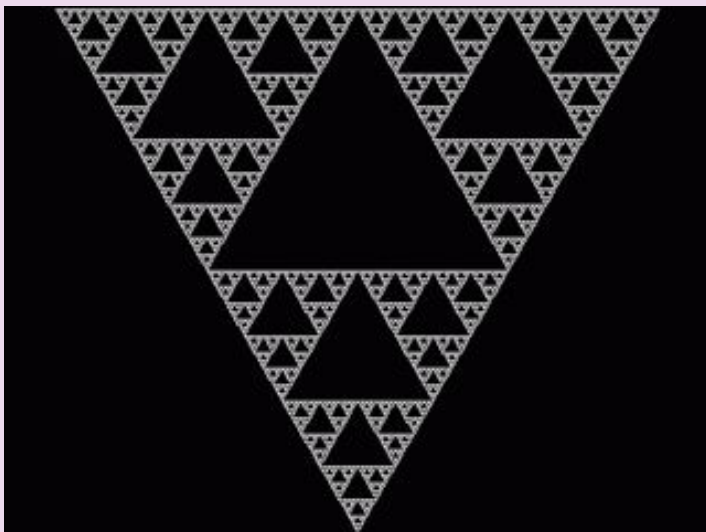
# Геометрические фракталы



Построение фрактала «Драконова ломаная»

# Геометрические фракталы

## Треугольник Серпинского

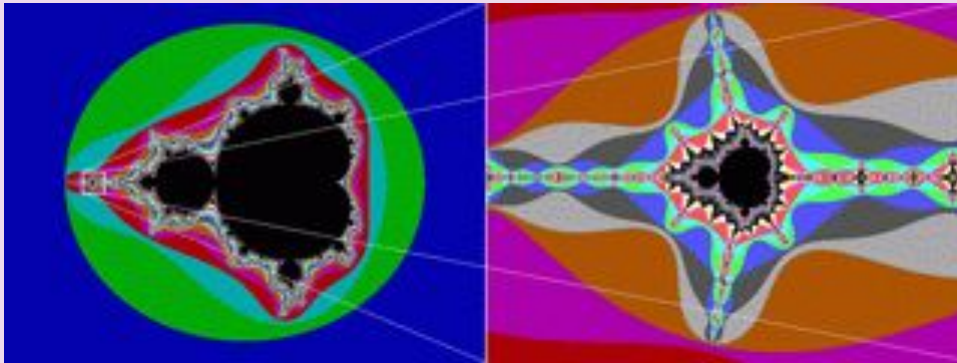


- Задаем **начальные условия**: нулевое поколение – это правильный треугольник  $S_0$  со стороной  $1$ .
- Задаем процедуру, которая преобразует нулевое поколение: делим средними линиями исходный правильный треугольник на четыре равных треугольника и центральный выбрасываем.
- С тремя оставшимися делаем то же самое и так до бесконечности. В результате возникнет убывающая последовательность замкнутых множеств

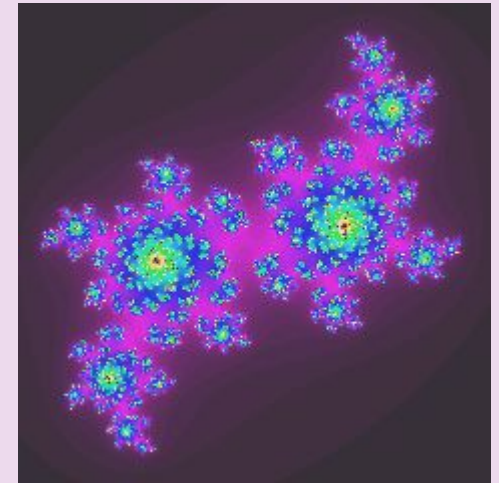


# Алгебраические фракталы

- Для построения алгебраических фракталов используются итерации нелинейных отображений, задаваемых простыми алгебраическими формулами.
- После того, как изображение построено, можно циклически изменять цвета закрашенных областей, и тогда и без того удивительное изображение «оживёт» на экране.



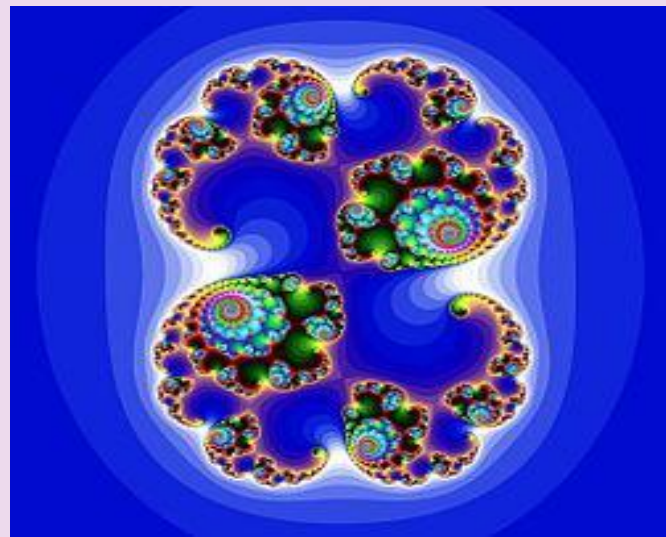
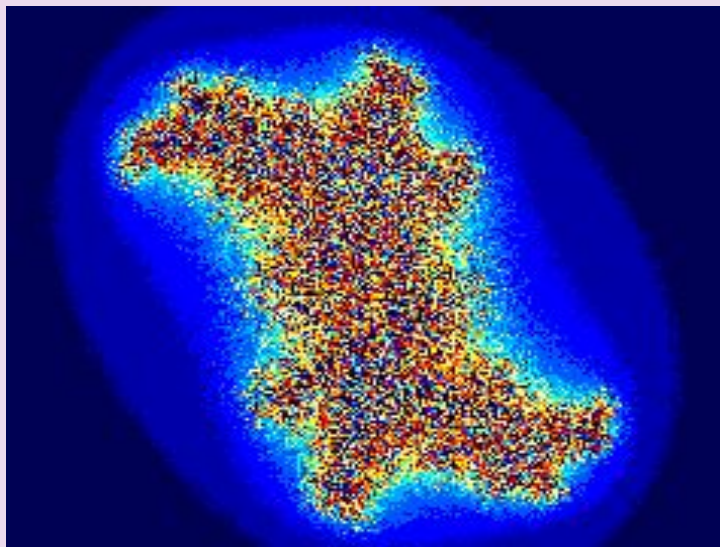
*Множество Мандельброта*



*Множество Жюлиа*

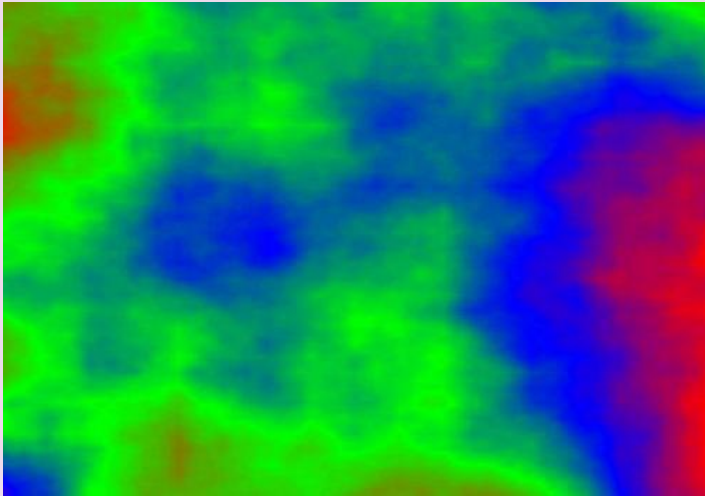
# Стохастические фракталы

- Фракталы, при построении которых в итеративной системе случайным образом изменяются какие-либо параметры, называются стохастическими.

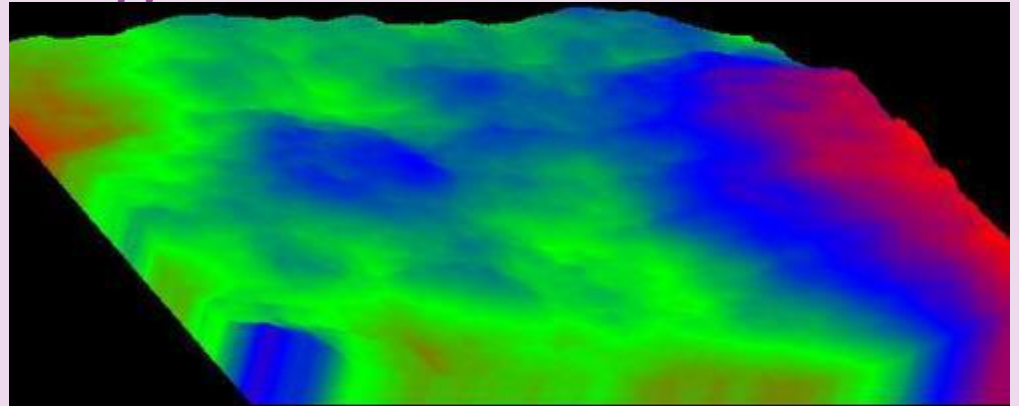


Рандомизированный стохастический фрактал на основе множества Жюлиа

# Стохастические фракталы



*Типичный стохастический фрактал – «Плазма»*



- Возьмем прямоугольник и для каждого его угла определим цвет.
- Далее находим центральную точку прямоугольника и раскрашиваем ее в цвет, равный среднему арифметическому цветов по углам прямоугольника плюс некоторое случайное число.
- Чем больше случайное число - тем более "рваным" будет рисунок.

# Применение фракталов

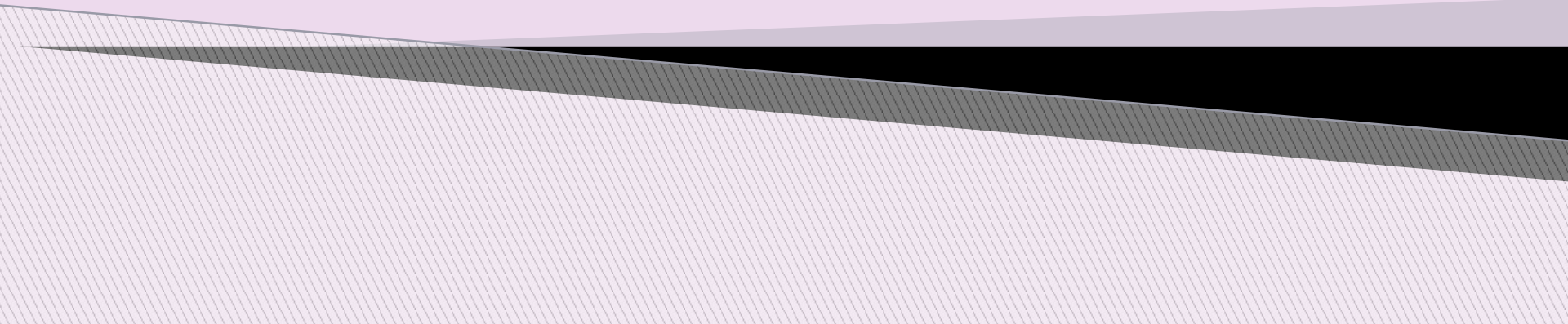
- Генерация изображений природных объектов;
- Моделирование динамических процессов в жидкости и газе;
- Моделирование биологических объектов;
- Проектирование антенных устройств;
- Алгоритмы сжатия информации ( в т.ч. В компьютерной графике);
- Создание текстур в графических программах.



# Примеры фракталов



# Трёхмерная графика



# Понятие трехмерной графики

- **Трёхмерная графика** (3D Graphics) — совокупность приемов и инструментов (как программных, так и аппаратных), предназначенных для изображения объёмных объектов.
- Трёхмерное изображение на плоскости включает построение геометрической проекции трёхмерной **модели сцены** (виртуального пространства моделирования) на плоскость с помощью специализированных программ.

# Понятие трехмерной графики

**Сцена** включает в себя несколько категорий объектов:

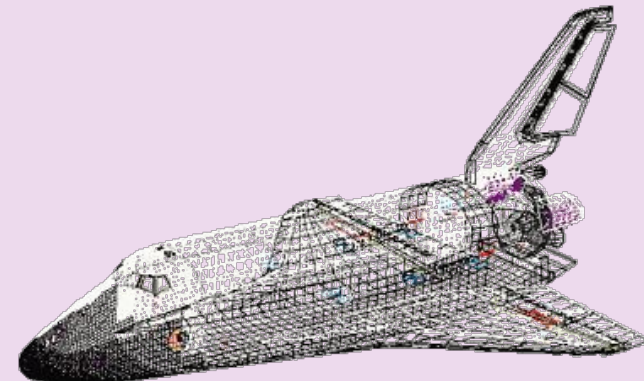
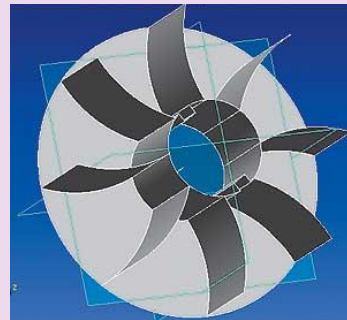
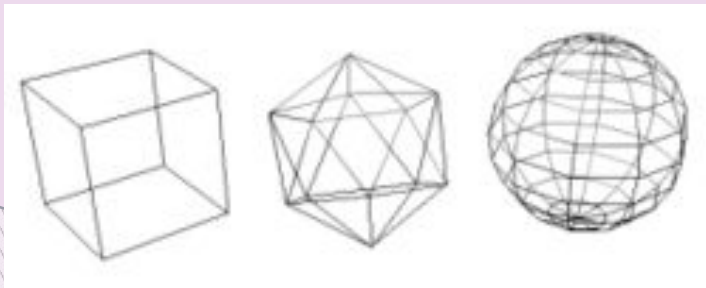
- ▣ **Геометрия** - построенная с помощью различных техник модель;
- ▣ **Материалы** - информация о визуальных свойствах модели (например, цвет стен и отражающая/преломляющая способность окон);
- ▣ **Источники света** - настройки направления, мощности, спектра освещения;
- ▣ **Виртуальные камеры** - выбор точки и угла построения проекции;
- ▣ **Силы и воздействия** - настройки динамических искажений объектов (применяется в основном в анимации);
- ▣ **Дополнительные эффекты** - объекты, имитирующие атмосферные явления: свет в тумане, облака, пламя и пр.;

**Задача трёхмерного моделирования** — описать эти объекты и разместить их в сцене с помощью геометрических преобразований в соответствии с требованиями к будущему изображению.



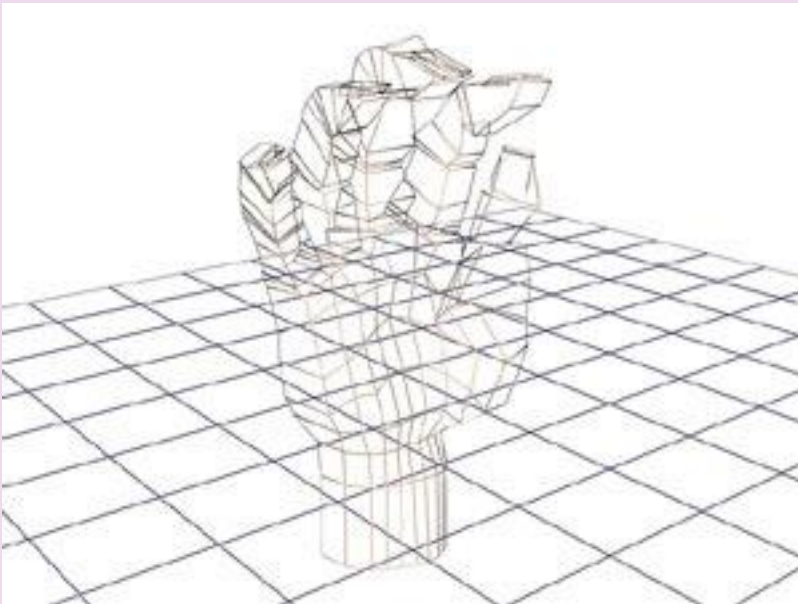
# Виды 3-D моделирования

- ▣ **Каркасная модель** - полностью описывается в терминах точек и линий.
- ▣ **Поверхностная модель** — это совокупность поверхностей, ограничивающих и определяющих трехмерный объект в пространстве. Моделирование поверхностей применяется для детальной отработки вне
- ▣ **Твердотельное моделирование** - самое полное и достоверное построение реального объекта. Результатом является монолитный образец объекта, который включает в себя такие компоненты как линии, грани, объем и масса.

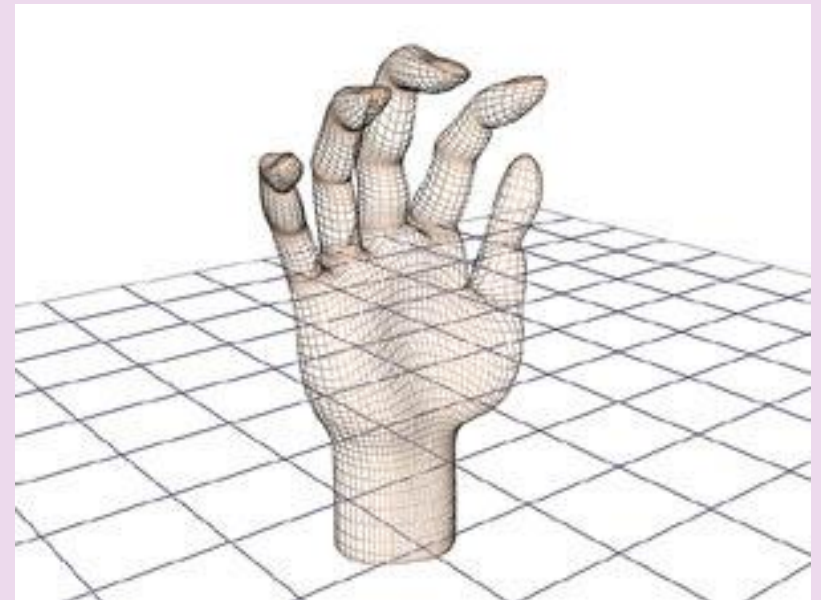


# Этапы 3-D моделирования

## 1 этап - построение каркасной модели



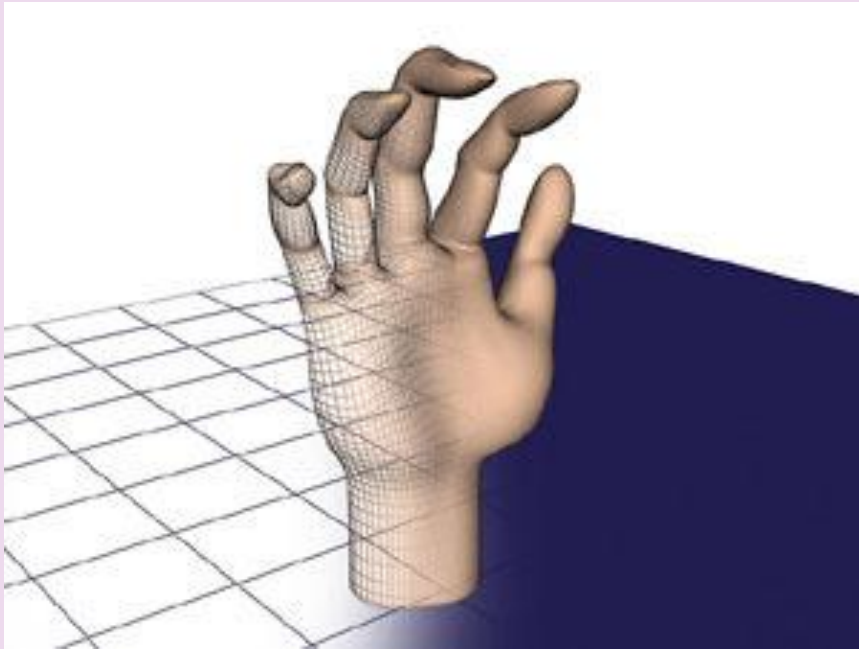
каркас руки, изготовленный из малого количества полигонов - 862



Здесь замыкающие объект линии выглядят более естественными и закругленными. Но это требует большего количества полигонов: 3444.

# Этапы 3-D моделирования

## 2 этап - определение поверхностных текстур



### Цвет:

Какого поверхность цвета?

Однородно ли она окрашена?

### Текстура:

Ровная ли поверхность или на ней есть вмятины, бугры, рихтовка или что-то подобное?

### Отражающая способность:

Отражает ли поверхность свет?

Четкие ли отражения или они размазаны?

# Этапы 3-D моделирования

## 3 этап - настройка освещения



**Освещение** выражается в двух эффектах: затенения (shading) и тени (shadow).

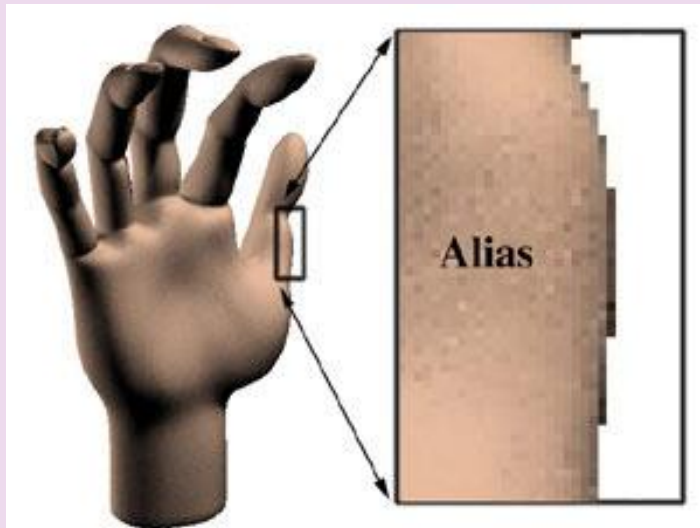
**Затенение** - изменение интенсивности освещения объекта от одной его стороны к другой. Это создает иллюзию, что объект кроме высоты и ширины имеет еще и глубину.

Иллюзия веса создается вторым эффектом: **тенью**.

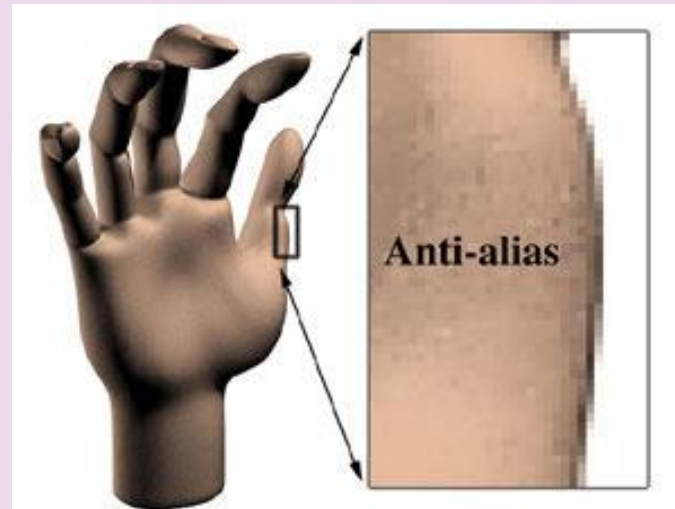
# Этапы 3-D моделирования

## 4 этап - сглаживание

процесс добавления пикселей для реалистичного представления неровных (жестких) участков.



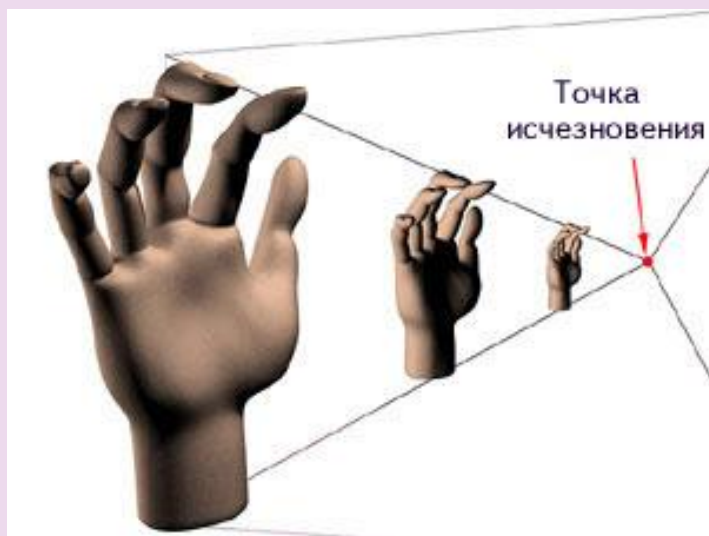
«Лесенки» появляются при прорисовке пиксельного изображения с помощью прямых линий.



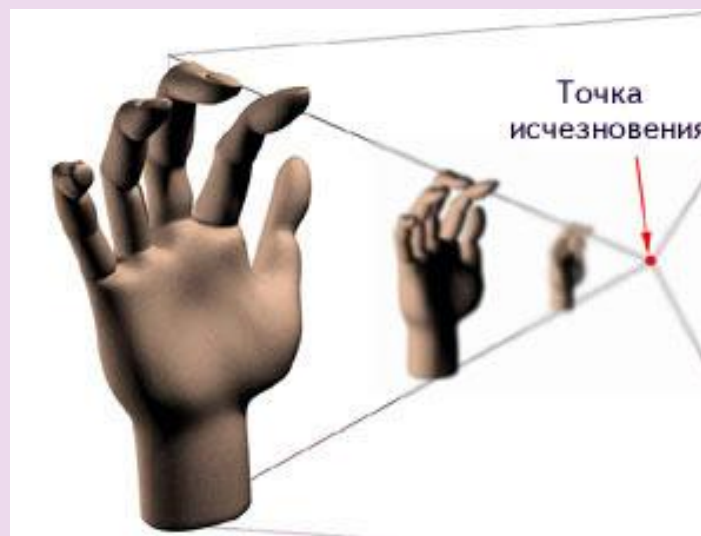
Прорисовка "серых" пикселей вокруг линий несколько их размывает. Эффект помогает убрать "лесенки" и делает объект более реалистичным.

# Этапы 3-D моделирования

## 5 этап - настройка движения



Перспектива



Глубина резкости

# Рендеринг

Для того чтобы художник, создавая модель, мог видеть результат своей работы, компьютер непрерывно строит проекции, отсекает невидимые грани, просчитывает источники освещения, переводя огромные массивы числовых данных в форму, доступную для восприятия человеческим глазом. Этот процесс называется **рендерингом**.

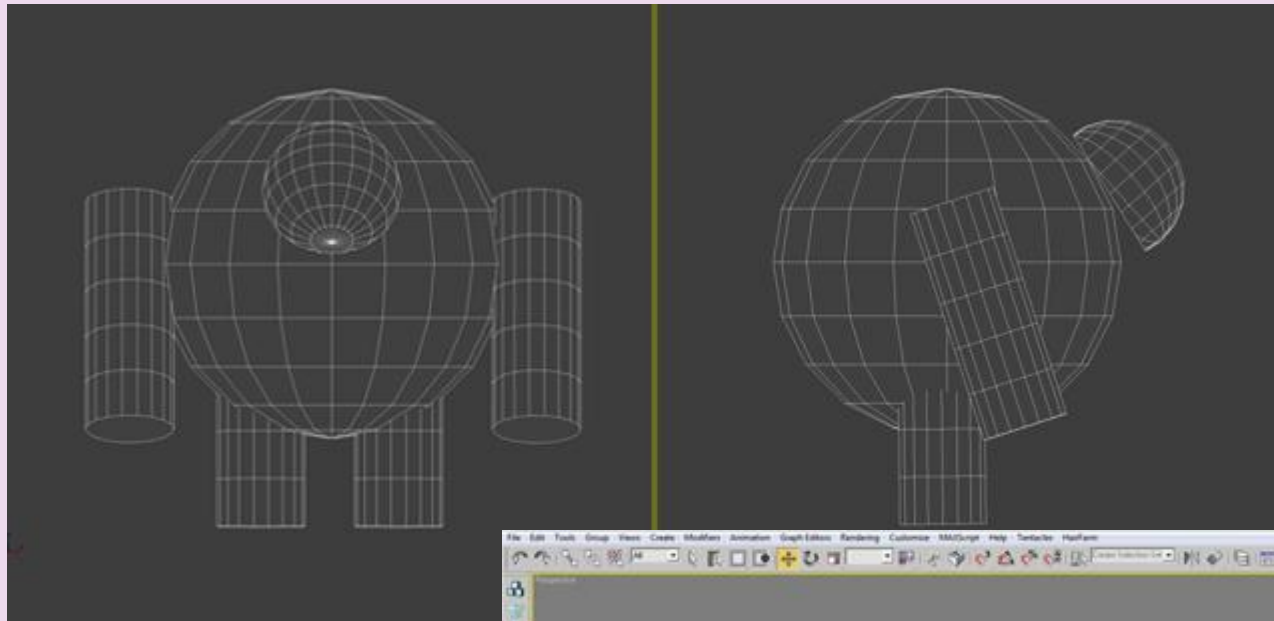
**Рендеринг** — процесс получения изображения по модели с помощью компьютерной программы.

# Методы рендеринга

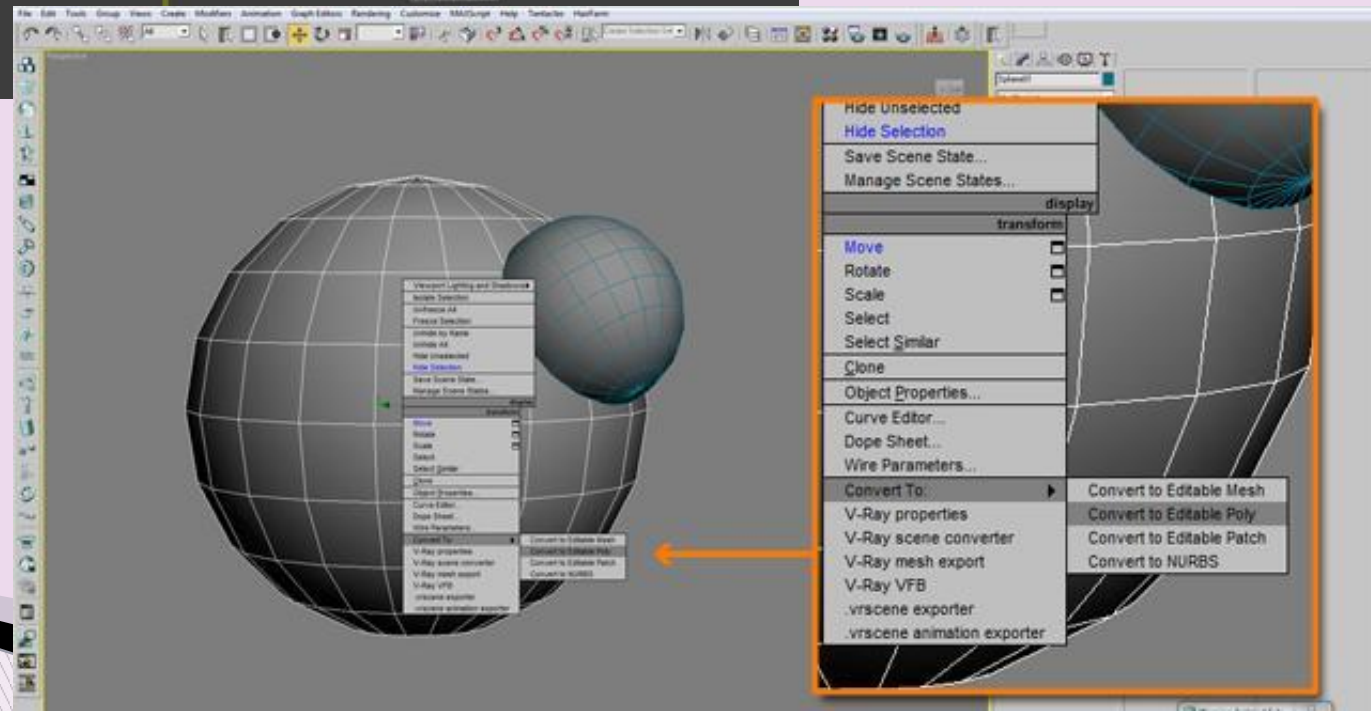
- ▣ **Растреризация, метод сканирования строк** - визуализация производится проецированием объектов сцены на экран без рассмотрения эффекта перспективы относительно наблюдателя.
- ▣ **Метод бросания лучей** - сцена рассматривается, как наблюдаемая из определённой точки. Из точки наблюдения на объекты направляются лучи, с помощью которых определяется цвет пикселя на двумерном экране. Лучи прекращают своё распространение, когда достигают любого объекта сцены либо её фона.
- ▣ **Трассировка лучей** - из точки наблюдения на объекты сцены направляются лучи, с помощью которых определяется цвет пикселя на двумерном экране. Но луч не прекращает своё распространение, а разделяется на три компонента, каждый из которых вносит свой вклад в цвет пикселя отражённый, теневой и преломленный. Количество таких разделений на компоненты определяет глубину трассирования и влияет на качество и фотореалистичность изображения. Метод позволяет получить очень фотореалистичные изображения, но при этом он очень ресурсоёмкий.



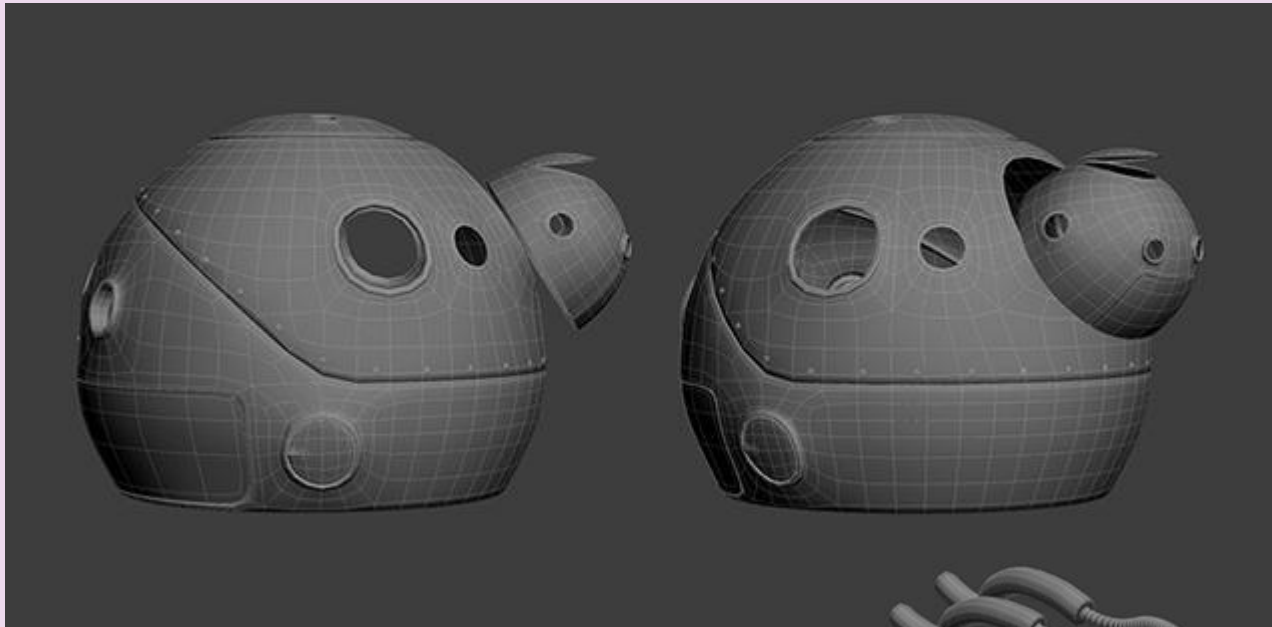
# Пример построения трехмерной модели в 3-D Max



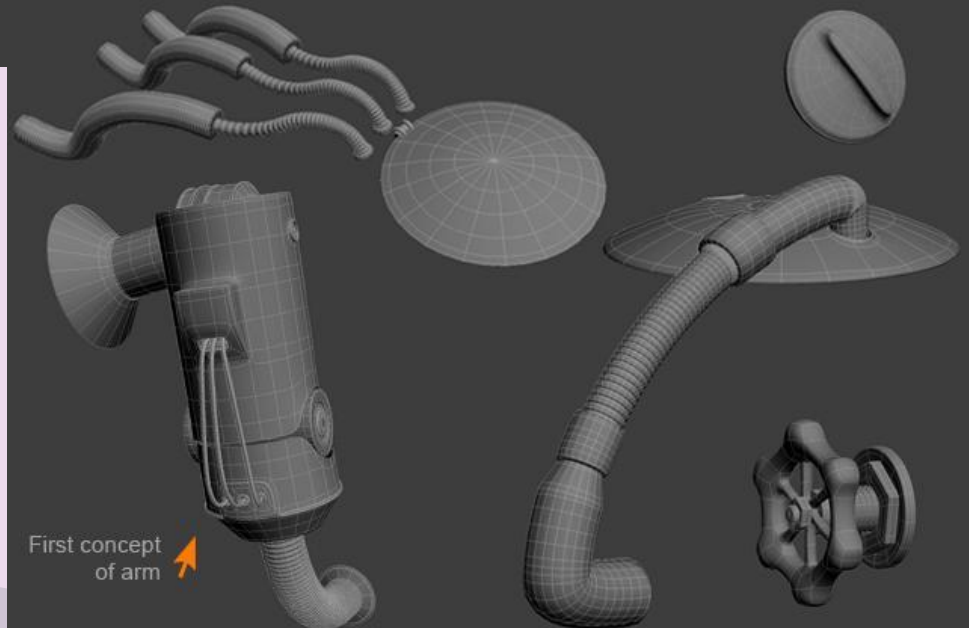
*Построение  
каркаса*



# Пример построения трехмерной модели в 3-D Max

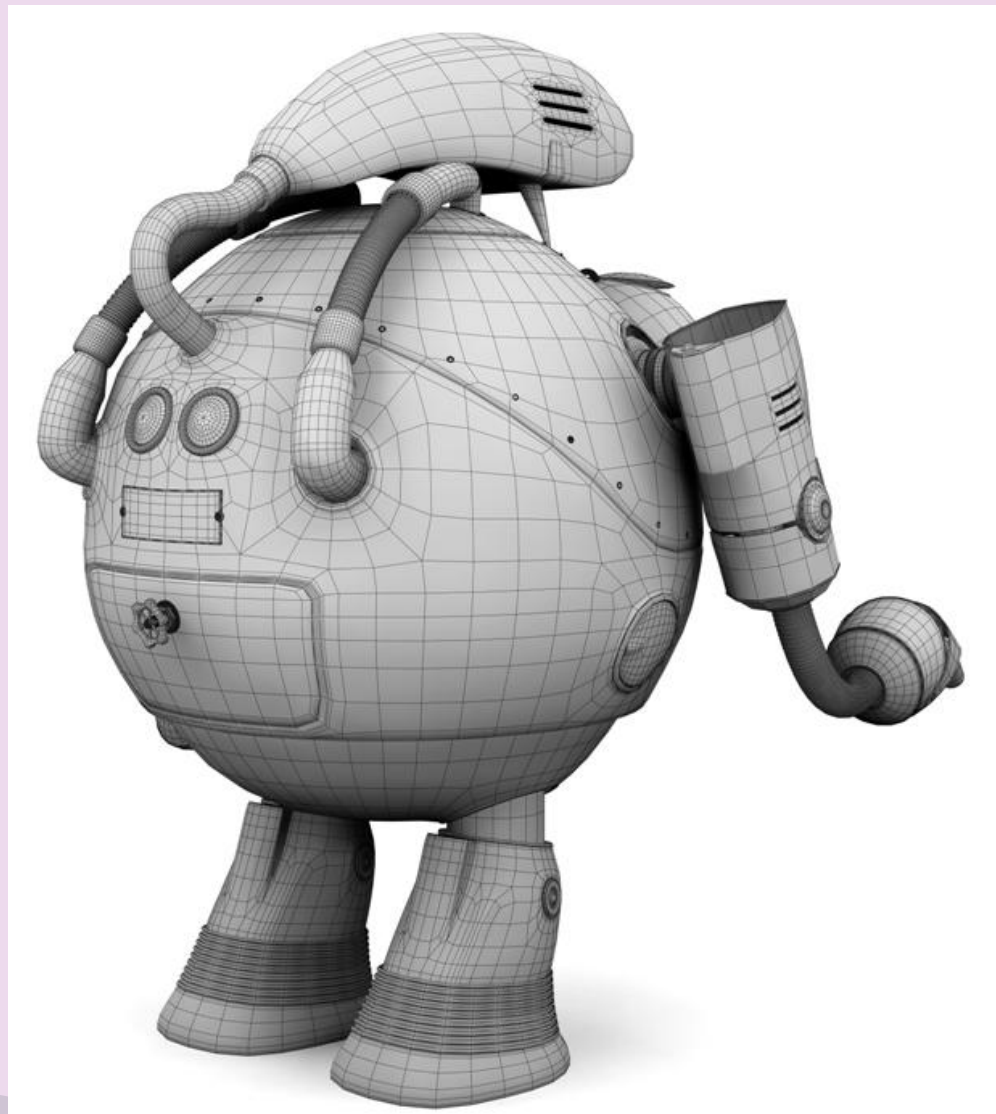


*Построение  
каркаса*

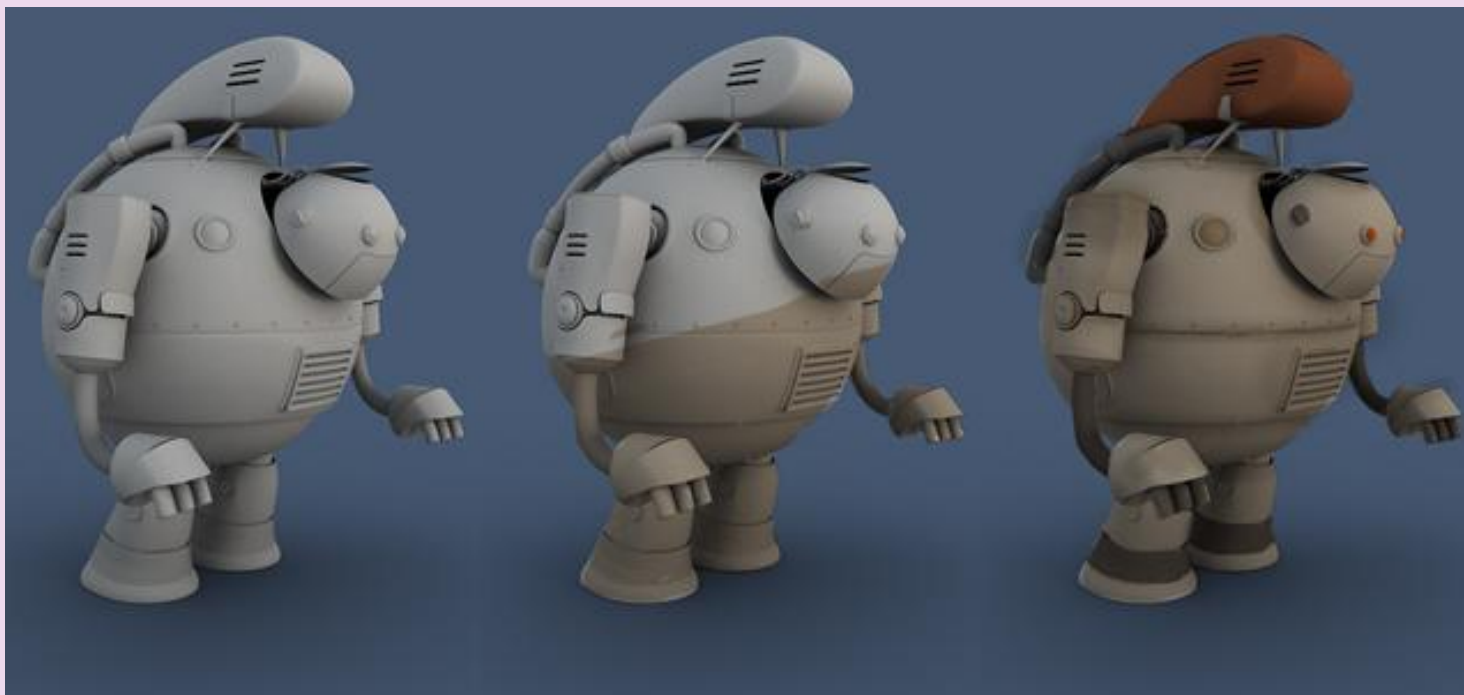


First concept  
of arm

## Пример построения трехмерной модели в 3-D Max



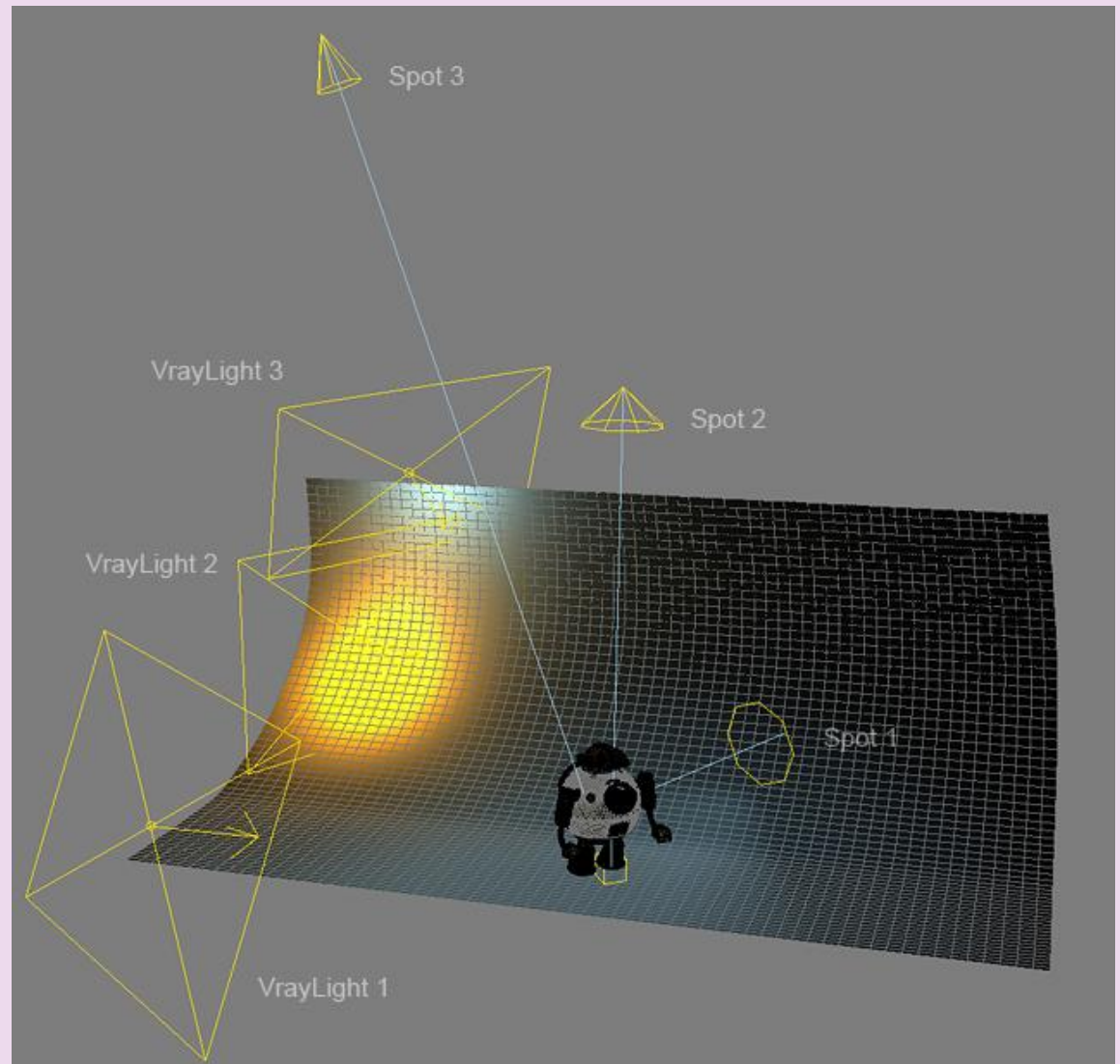
## Пример построения трехмерной модели в 3-D Max



*Текстурирование объекта*

# Пример построения трехмерной модели в 3-D Max

*Настройка  
пространства и  
освещения*



# Пример построения трехмерной модели в 3-D Max

*Рендеринг*

