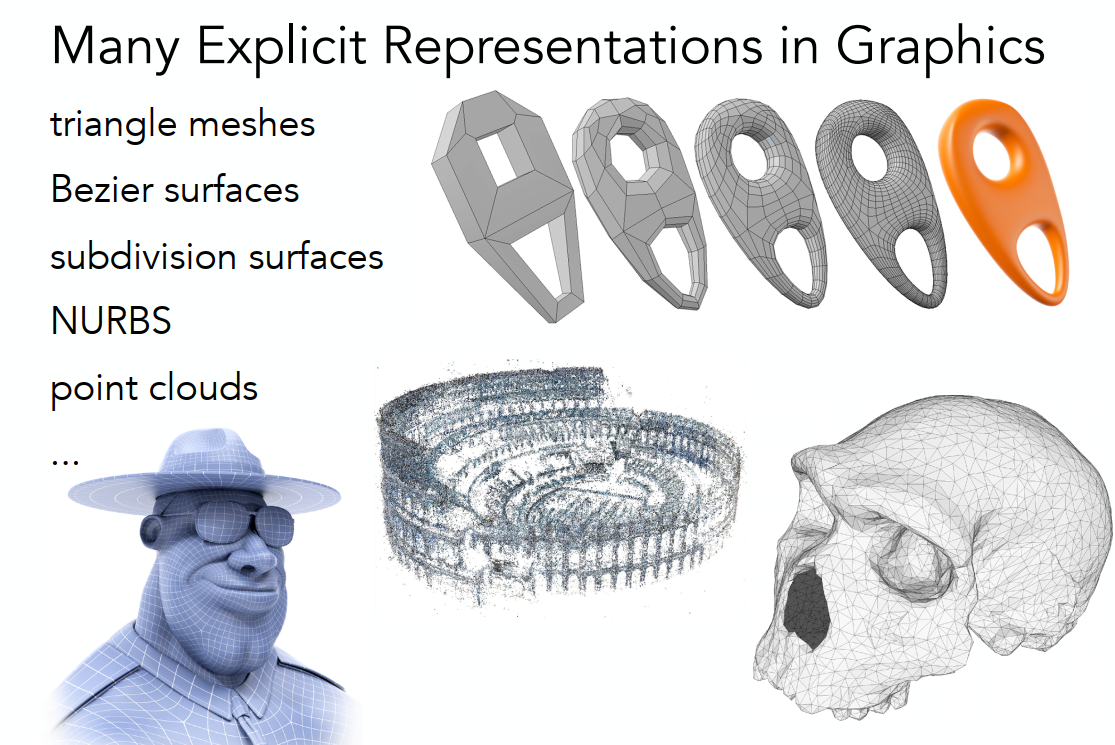
# (四)Explicit Representations in Computer Graphics

Many Explicit(显式) Representations in Graphics



## 1、Point Cloud (Explicit)

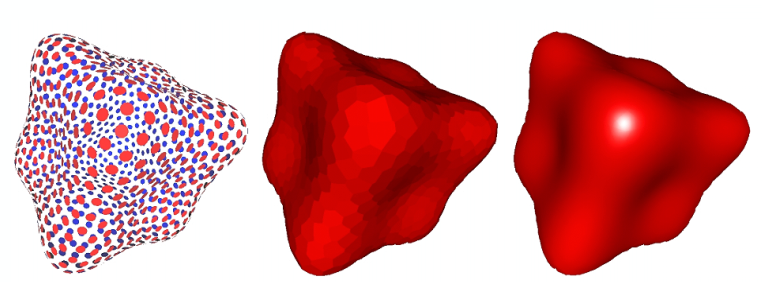
Easiest representation: list of points (x,y,z)(仅用XYZ就可以代表一个三维model)

Easily represent any kind of geometry(易于表示各种几何图形)

Useful for LARGE datasets (>>1 point/pixel)(但是需要的存储空间很大)

Often converted into polygon mesh(经常会转换成多边形面)

Difficult to draw in undersampled regions



把物体表面上一堆点的每一个点都表示为一个点，也就是说模型由一个个点点组成。如上图，当点云的密度非常大的时候，模型看起来没有什么问题，但是图片自上往下点云密度逐渐稀疏，整个模型也就看不出来了。因此如果想用点云表示物体需要点云的密度非常大。一般在扫描模型的时候会用到（然后还会再转化为一个个面）

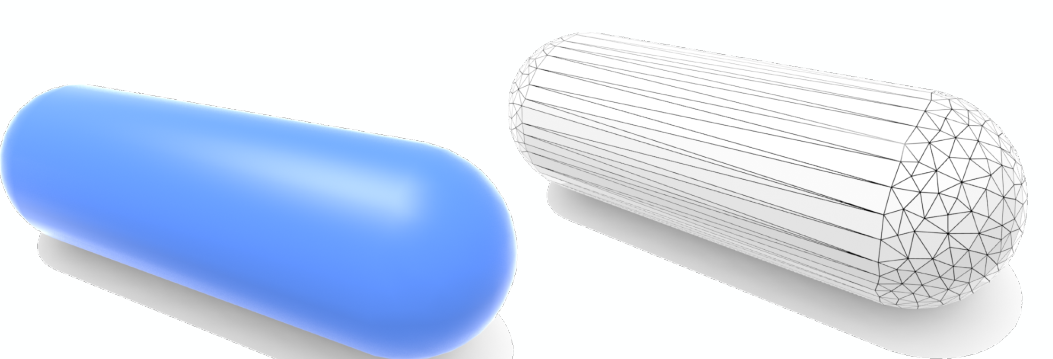
## 2、Polygon Mesh(多边形曲面) (Explicit)

Store vertices & polygons (often triangles or quads(四边形))

Easier to do processing / simulation, adaptive sampling

More complicated data structures

Perhaps most common representation in graphics

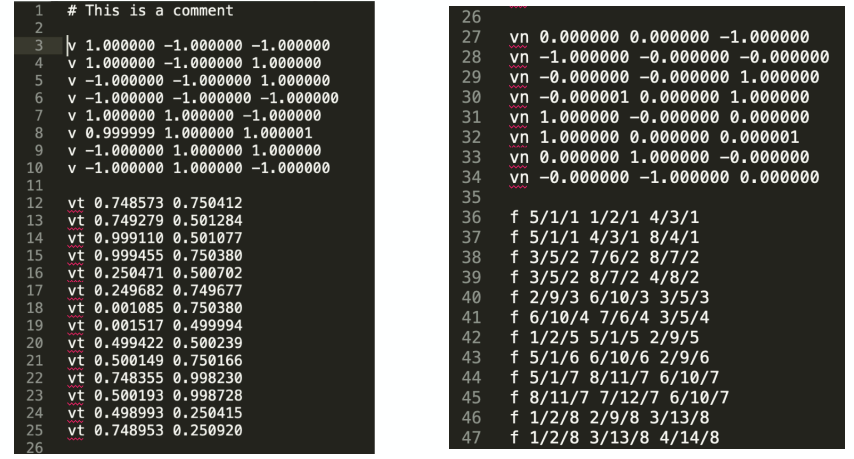


## 3、The Wavefront Object File (.obj) Format

Commonly used in Graphics research

Just a text file that specifies vertices(顶点), normals(法向量), texture coordinates(纹理坐标) and their connectivities(连接关系)

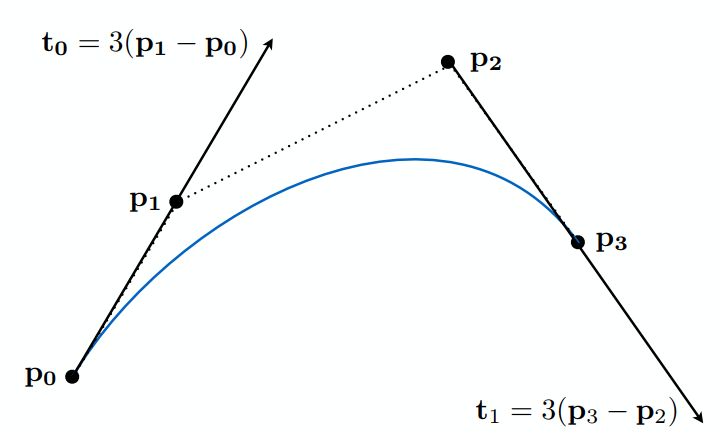
在图形学中如何表示用三角形面形成的物体？



将.obj文件拓展名改为文本文件，就会得到上图的一系列数据.v开头的是记录的顶点信息，vt记录的是纹理信息，vn记录的是面法线信息，f记录的是由哪三个（顶点/纹理/面法线）组成一个三角形。上图记录的是一个立方体的模型数据(8顶点 6法线 12纹理)，有一些多余的数据是由于自动建模中数据冗杂造成的。

（比如f 5/1/1 1/2/1 4/3/1就是说用第5 1 4个顶点构建的三角面 用第1 2 3个纹理坐标 用第1 1 1个法线向量）

# Bézier Curves(贝塞尔曲线)



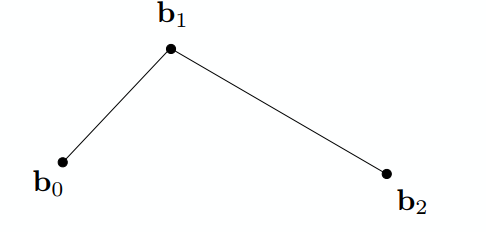
贝塞尔曲线如上，从p0开始，到p3结束。开始是沿着p0p1方向，结束是沿着p2p3方向

那么给定任意多的点，如何画出贝塞尔曲线？

## 1、Evaluating Bézier Curves (de Casteljau Algorithm)

以三点的贝塞尔曲线举例

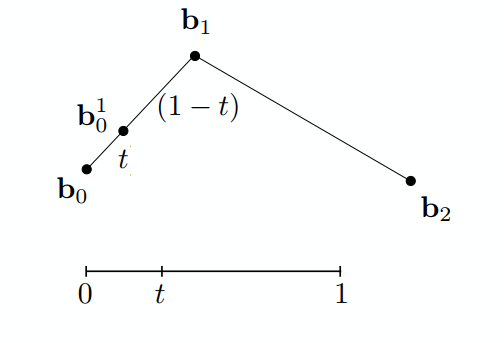
1. Consider three points (quadratic Bezier(二次贝塞尔))



首先这条线一定要从b0开始，到b2结束。b1决定了整条线要向哪个方向弯曲。

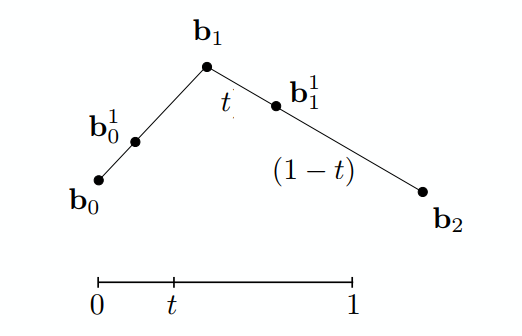
②Insert a point using linear interpolation

给定一个时间t，从0开始到1结束。



三个输入的点b0、b1、b2确定了两条线段b0b1、b1b2，在b0b1上认为b0是0，b1是1，那么在这个线段上，找一个点t。假设此时t=1/3，这个点记为b10。

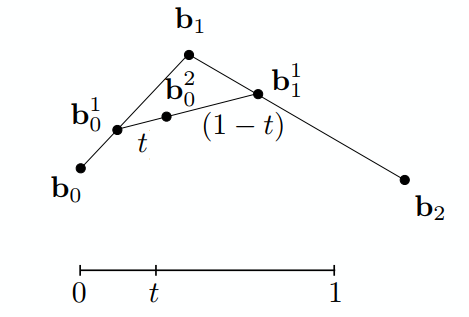
③Insert on both edges



同样的，在b1b2上，认为b1是0，b2是1，也找这样一个点t=1/3，这个点记为b11。

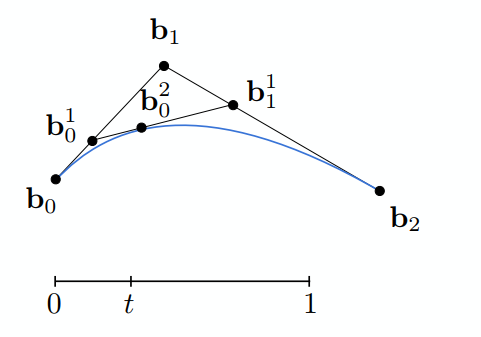
到此我们通过三个点b0、b1、b2得到了两个点b10、b11。

④Repeat recursively(递归)



把刚刚得到的两个点b10、b11连起来，再认为这条线段又是从0到1，b10是0，b11是1，再取同样的t=1/3，这时我们找到了1个点b20，这个点即为由b0、b1、b2三点确定的一条贝塞尔曲线在时间t=1/3上的点。

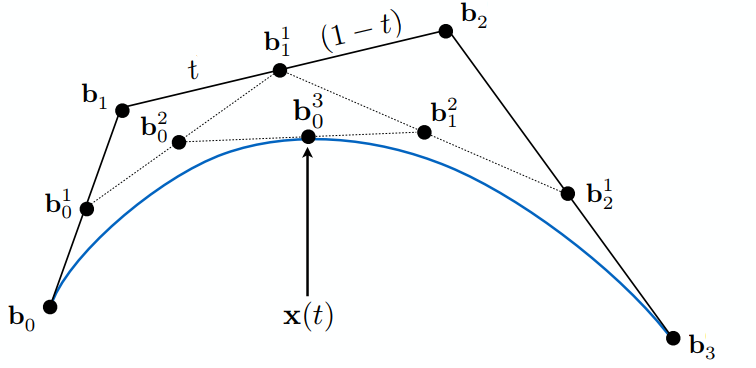
⑤Run the same algorithm for every t in [0,1]



t的取值从0到1，每取一个值都重复上述步骤，然后得到了一系列点，将所有的点连起来就得到了一条贝塞尔曲线。

Four input points in total

Same recursive linear interpolations

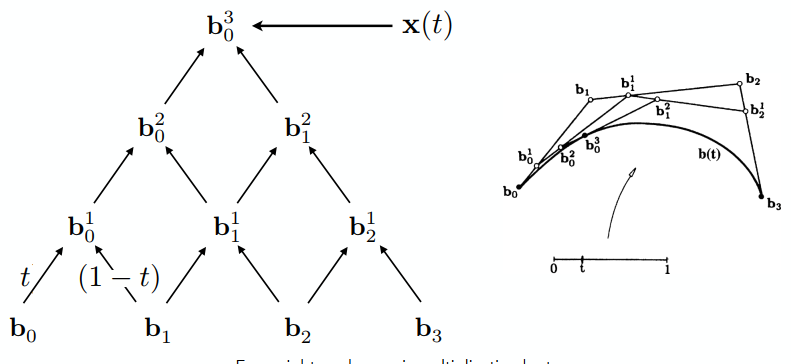


四个点确定的一条贝塞尔曲线同理，只不过这次的操作需要先由4个点b0、b1、b2、b3组成的3条线段b0b1、b1b2、b2b3确定3个点b10、b11、b12，再由3个点b10、b11、b12确定的2条线段b10b11、b11b12确定2个点b20、b21，再由2个点b20、b21确定的1条线段b20b21确定一个点b30。

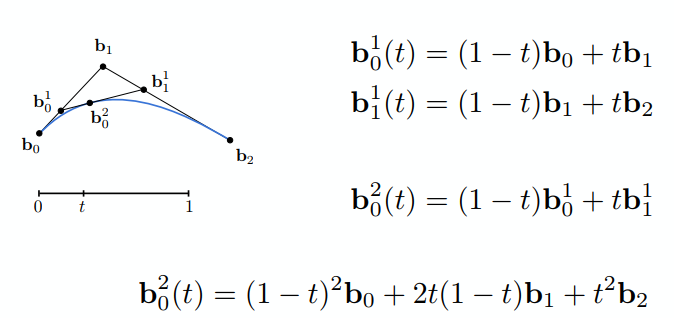
然后再对t取从0到1的所有值，重复上述步骤画出贝塞尔曲线。

## 2、Evaluating Bézier Curves Algebraic Formula

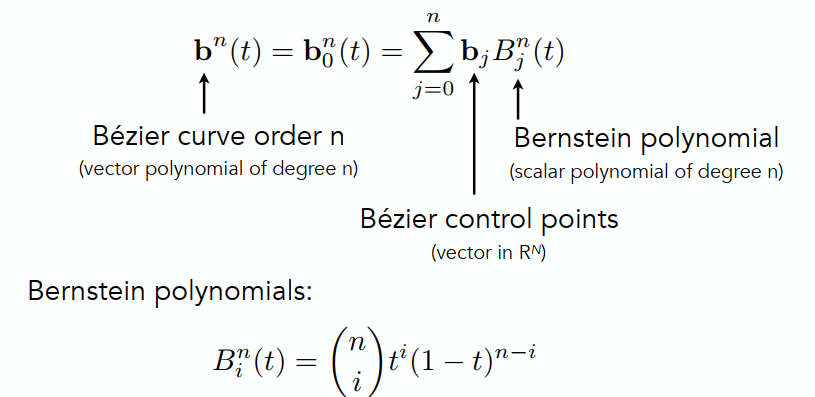
de Casteljau algorithm gives a pyramid of coefficients

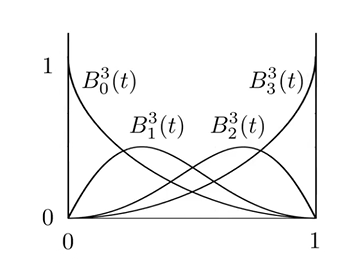


Example: quadratic Bézier curve from three points

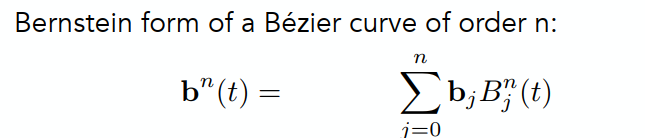


Bernstein form of a Bézier curve of order n(n+1个点，n阶贝塞尔曲线可用n阶伯恩斯坦多项式描述)





(与二项展开式类似)



Example: assume n = 3 and we are in R3

i.e. we could have control points in 3D such as:（3D贝塞尔曲线的区别就是b是一个三维向量）

b0 = (0, 2, 3), b1 = (2, 3, 5), b2 = (6, 7, 9), b3 = (3, 4, 5)

These points define a Bezier curve in 3D that is a cubic polynomial in t:

bn(t) = b0(1-t)3+ b13t(1-t) 2+ b23t2(1-t)2 + b3t3

## 3、Properties of Bézier Curves

Interpolates endpoints(插值端点)

--For cubic Bézier:b(0) = b0; b(1) = b3(起点在b0，终点在b3)

Tangent to end segments(与末端线段相切)

--Cubic case: b‘ (0) = 3(b1-b0); b’ (1) = 3(b3 -b2)(这个3仅仅是因为有四个控制点的三次贝塞尔曲线)

Affine transformation property

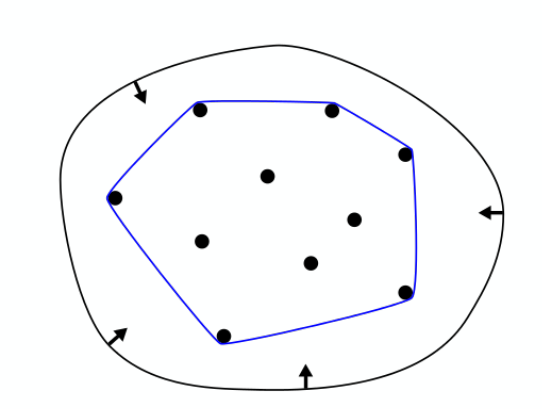
--Transform curve by transforming control points(对给定顶点画出的贝塞尔曲线，与该些顶点做完仿射变换的再画出的贝塞尔曲线是一样的)(注意是仿射变换而不是投影变换)

Convex hull property(凸包性质)

--Curve is within convex hull of control points(曲线位于控制点的凸包内)

## 4、BTW: What’s a Convex Hull

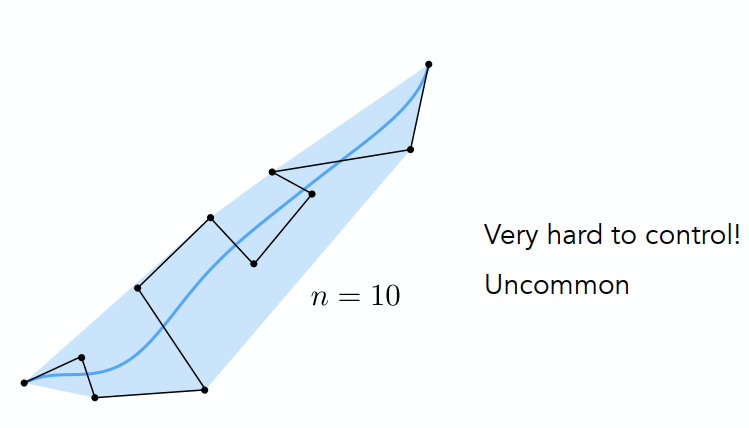
凸包：能够包围一系列给定的几何形体的最小的凸多边形



更直观来讲，如上图，在一个板子上打好多钉子，用一个橡皮筋拉大包住所有钉子后放手，让橡皮筋自然收缩，被卡在的一圈钉子组成的多边形即为凸包。

## 5、Piecewise Bézier Curves(逐段贝塞尔曲线)

Higher-Order Bézier Curves?(高阶贝塞尔曲线)



当控制点的数量过多，就可能导致某些地方弯曲不到我想要的程度，很难去控制局部。

Piecewise Bézier Curves(逐段贝塞尔曲线)

Instead, chain many low-order Bézier curve(将低阶的贝塞尔曲线连接起来)

Piecewise cubic Bézier the most common technique(最常见的方法是每四个顶点定义一条贝塞尔曲线)



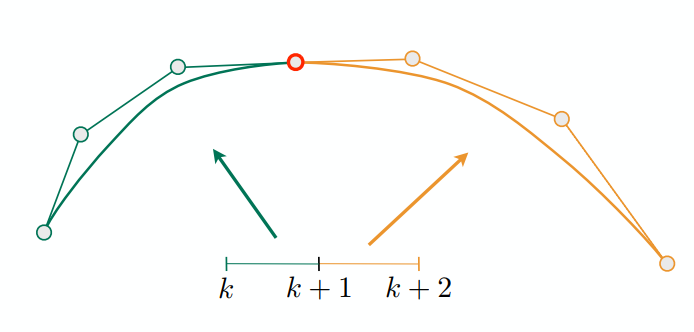
将贝塞尔曲线每4个点做一次分段，其中1号和3号点是正常插入的位置点，而2号和4号则是被当做了“控制手柄”。

一个关于分段贝塞尔曲线的例子：[Bezier Curve Edit (hws.edu)](https://math.hws.edu/eck/cs424/notes2013/canvas/bezier.html)

### Piecewise Bézier Curve – Continuity

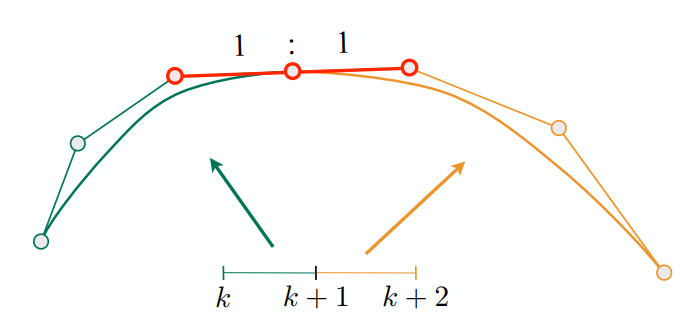
C0 continuity:an=b0 (C0连续)

第一段的终点等于第二段的起点，这种点叫C0连续。



C1 continuity:an=b0=1/2(an-1+ b1) (C1连续)

第一段的终点等于第二段的起点（公共点），且公共点相邻的左右两个控制点和这个公共点之间距离相等且三点共线（方向相反、等距、共线）。



## 6、Other types of splines

Spline(样条)

​ --a continuous curve constructed so as to pass through a given set of points and have a certain number of continuous derivatives.(通过给定的一系列控制点构成的一条连续曲线并具有一定数量的连续导数。)

​ --In short, a curve under control(简而言之，曲线处于可控状态)

B-splines(基函数样条)

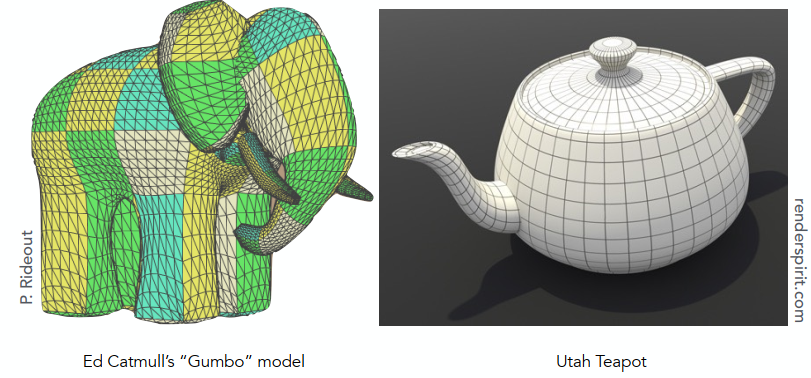
​ --Short for basis splines

​ --Require more information than Bezier curves

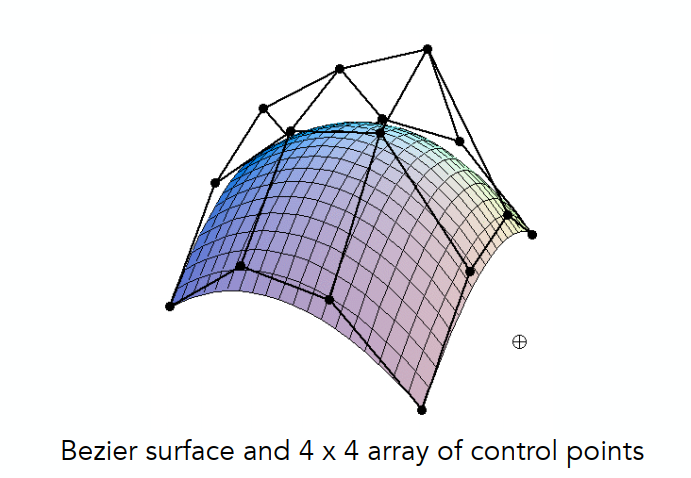
​ --Satisfy all important properties that Bézier curves have (i.e. superset)

# (二)Bézier Surfaces(贝塞尔曲面)

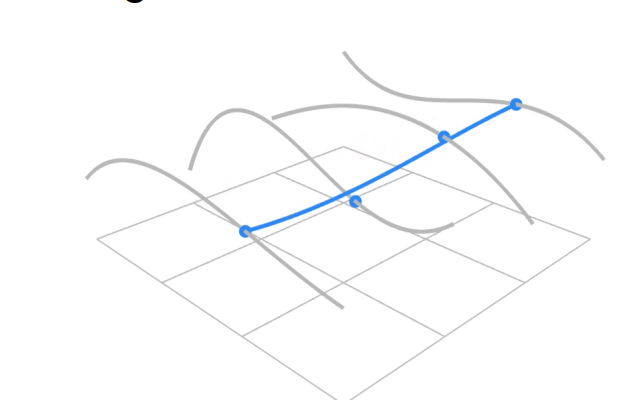
## Extend Bézier curves to surfaces



## Bicubic Bézier Surface Patch



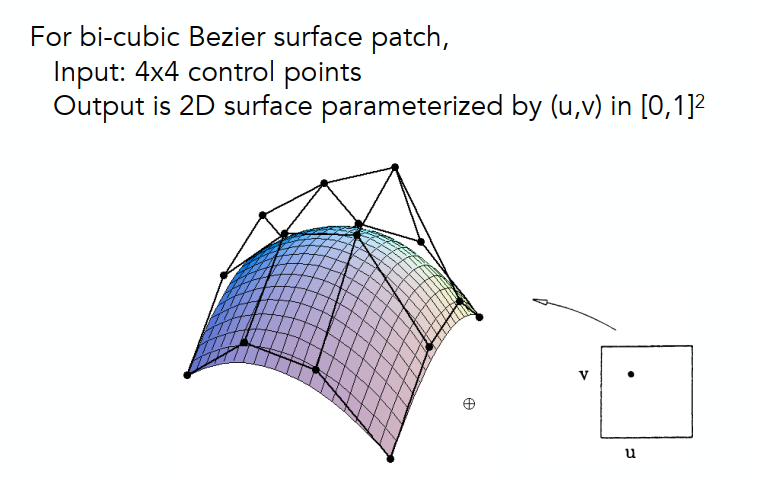
## Visualizing Bicubic Bézier Surface Patch



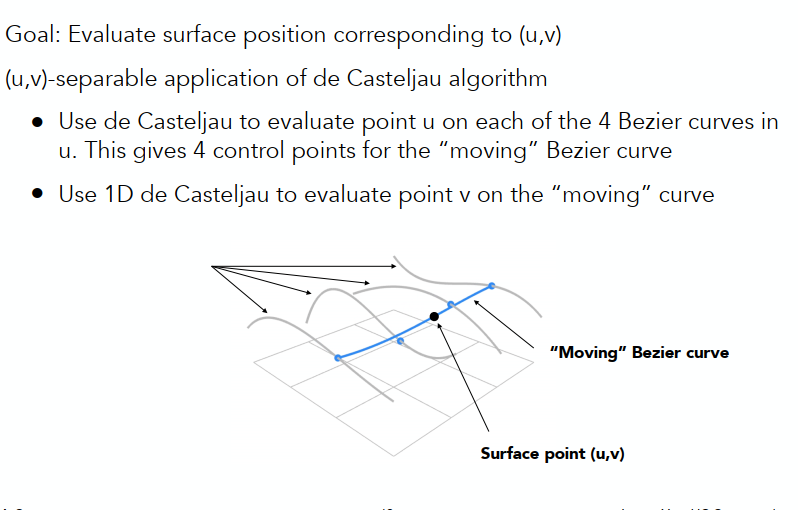
在两个方向上分别运用贝塞尔曲线。

首先在一个方向上做贝塞尔曲线，然后把得到的点作为控制点，再做一次贝塞尔曲线即可。

## Evaluating Bézier Surfaces

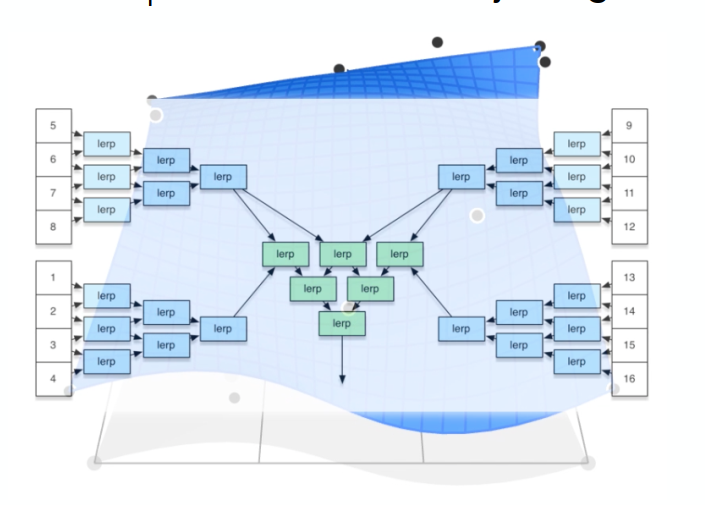


## Evaluating Surface Position For Parameters (u,v)



可以通过一个u方向和一个v方向做一个映射关系。

## Method: Separable 1D de Casteljau Algorithm



## Mesh Operations: Geometry Processing

Mesh subdivision网格细分：用更多的三角形表示更完美的曲面

Mesh simplification网格简化：虽然三角形越多曲面越光滑，但是过多的曲面映像内存。需要找个折中的范围

Mesh regularization网格正规化：不至于出现特别尖特别长的三角形，最好是所有三角形都长得差不多

