计算机图形学和虚拟环境读书报告

本文为了将两本教材的相关内容进行整合，没有完全按照课本顺序编排，在每一个小节的标题后附上了两本书对应的章节。

一、概论

I. 计算机图形学的历史

计算机图形学是一个年轻且充满活力的学科，其历史可以追溯到 1963 年推出的经典绘图系统 Sketchpad。Sketchpad 是一个交互系统，用户可以使用光笔在屏幕上绘图，这标志着图形对象的创立，使计算机从图片处理中分离出来。

1962 年，第一款游戏《星球大战》诞生；1973 年，第一款具有图形用户界面（GUI）的操作系统 Alto 问世。这些标志性事件预示着一个新兴的计算机学科即将登上历史舞台。到了 80 年代，随着个人电脑的普及和处理能力的显著提高，计算机图形学逐渐走入了大众视野。1985 年，第一个图形学标准 GKS 问世，但它是基于 2D 的。1988 年，PHIGS 扩展了这一标准，涵盖了 3D 图形，并成为新的标准。1992 年，OpenGL 诞生，成为图形学中的重要基础，类似于计算机底层的汇编语言，并一直沿用至今。

如今，图形学有广泛的应用场景，主要可以分为渲染、几何和模拟三个方向，其中渲染是核心，即如何在屏幕上呈现一个虚拟世界。在计算机中，通过程序和代码构建的虚拟世界呈现在光栅界面上，这样的世界就成为了虚拟环境。虚拟环境由内容、几何和动态特性三部分组成。在这样的世界中，物体之间可以互相干涉，人类处于世界之外，接受来自这个虚拟世界的信息。实现虚拟环境的设备包括各种显示器、虚拟现实头盔、定位的数据手套和 CAVE 系统。对于人类而言，最重要的是这些设备能否提供真实的体验。

在计算机图形学中，真实感是一个复杂的概念。从几何角度来看，真实感体现在与实际物体的相似度；从光照角度来看，光照结果会因位置不同而变化；从行为角度来看，动画角色的动作和行为需要与人类相符，以产生共鸣。

这里涉及到美学层面的争论：何为真实？现实主义认为“艺术模仿自然”，唯美主义则认为“自然模仿艺术”。印象主义则提供了更多的借鉴，关注事物的抽象特质而非本身。真实感通常意味着一定的帧率，需要足够的渲染效率。然而，由于计算能力的限制，实时渲染的实现并不容易。在虚拟现实（VR）中，当进行运动时，场景的卡顿会影响真实感。这种妥协促使了渲染领域的不断进步，如今实时路径追踪已在工业界得到应用，未来将更加广泛。

虚拟环境的魅力在于其沉浸感。尽管人们知道眼前的世界是虚假的，但仍会感到置身其中。这种沉浸感依赖于人类的视觉特性。人类的视觉系统是虚拟的，能够感知光的范围有限，并且有一系列的过滤机制。人类的认知模型会影响对信息的处理和完善，从而形成对三维空间的透视感。

图形学的基本脉络已经搭建完成：如何构建一个虚拟世界？首先需要数学形式表达，然后描述空间中的物体位置关系，进而引入渲染方程，描述物体表面特性和光线作用，最后探讨计算机动画和模拟仿真。

II. 数学基础

数学是图形学的基础。从线性代数到微积分，再到概率论，构成了图形学的基石。书中介绍的数学知识主要是对线性代数的推广。

在三维数学中，三维坐标可以表示一个点或两点间的差值。在此基础上，定义了一系列向量运算：向量的加法、减法、模、点乘和叉乘。这些运算构成了矢量代数的基本，也是图形学的地基。

为了描述向量的方向，我们通常使用规范化的方法，将其转换为三个方向余弦。由于直角坐标系在方向表示上不直观，因此球坐标系也是一种重要的方向表示手段。球面上的立体

角是一个类似于角度的量度。

图形学中的变换是所有运算的核心，最常见的是仿射变换。仿射变换具有线性性，但无法用常见的矩阵表示平移变换。为解决这个问题，引入了齐次坐标，推广到三维向量中的 w-coord，实现一系列变换的复合。

我们已经可以表示平移、缩放、旋转、错切和镜像等常见变换。通过将变换矩阵相乘，可以合成这些变换；根据几何意义，逆变换也可以方便地求出。唯一的麻烦是旋转变换，绕坐标轴的旋转可以直接用矩阵表示，但绕任意轴的旋转则需要罗德里格斯旋转公式，这显得过于繁复。为此，四元数在图形学中得到了应用。

四元数是对复数的推广，可以进行加法、数乘、模、逆和共轭运算。四元数还可以进行叉积，这种运算的性质保证了四元数在旋转上的可操作性。同时，四元数可以进行插值和样条运算，实现非常平滑的效果。

III. 渲染方程

为了引入渲染方程，首先需要从物理层面解释光线的传播。光是一种辐射能量的聚合体，假设光具有独立的波长、稳定的能量分布和真空中光运动的媒介。在单位面积上某个方向的通量记作 Luminance。

我们可以用光通量来描述单位时间流过表面的能量，即 Radiosity。在此基础上对立体角求微分，得到 Radiance。换言之，Radiance 就是单位时间单位面积上的光通量。如果两个面片之间有一定的倾斜角度，需要引入投影连线和法线的夹角，从而定义一个面片之间的形状因子。这个形状因子只与夹角和距离有关，是一个定值。考虑某个面片的光通量，由自身发出的通量和其他所有面片乘上形状因子与反射率构成。这种形式可以转换为一个矩阵方程，称为辐射度方程。

在物体表面上，最核心的话题是反射量和反射方式。极端情况下，漫反射均匀地散射入射光线；另一极端是高光反射，以镜面方式反射入射光。BRDF 函数描述了这种分布。BRDF 是对确定入射光线下不同出射光线辐射度的描述。引入 BRDF 后，可以引入渲染方程。渲染方程由两项构成：自身发出的光和总的反射光。反射光是对所有入射光的累加，需要在球面上对 BRDF 和入射 Radiance 之积乘上 Cosine 值对每个立体角累计求和，即积分。

解决渲染方程的过程贯穿图形学的核心。栅格化方法是最古老且高效的解决策略，但效果有限。Whitted 风格的光线追踪可以实现真实的渲染，但算力消耗过高，通常用于离线渲染。Path Tracing 因其复杂性得到了广泛应用，但由于蒙特卡罗积分的采样操作，仍不能实时实现。直到最近，实时 Path Tracing 在复杂度和真实性之间达成了良好妥协，应用越来越广泛。

IV. 颜色 【黑书 4】

颜色是虚拟与真实的分界线，历来为哲学家所探讨。一个有趣的悖论是色盲悖论：如果一个人与生俱来是色盲，那么即便他认知的世界与他人不同，他看到的依然是他所认知的世界。这引发了关于认识论的争论，但我们关注的是看到的是否是真实的。

从光的角度看，不同波长的光具有不同的密度，形成一种分布函数。在视锥细胞的过滤效应下，形成了人类对颜色的感知。因此，人对颜色的观察具有一种积分的形式，可以用 Delta 函数描述单色光。人类有三种视锥细胞：L 型、M 型和 S 型。三种细胞有各自的反应函数，对不同波长下的光分布积分形成的三元实数对(l,m,s)描述了颜色。用三个固定的阶跃函数替换自然界中的分布，得到一个由三个颜色反应基函数构成的线性空间，任意颜色都可以在此空间中找到。

接下来引入颜色匹配函数。根据线性性，我们尝试找到每种基色的强度权重，通过加权

获取目标颜色的匹配。最常用的匹配是 CIE-RGB 系统，得到的权重即为(r, g, b)三个参数。将这些颜色投影，得到色度空间，并定义亮度。但自然的 RGB 空间存在负数区域，不易表示，XYZ 空间解决了这个问题。XYZ 空间的 X 和 Z 为零亮度，形成了完美的 CIE-XYZ 色度。

在 CRT 显示器上，每个像素点由 R、G、B 三种荧光色点构成，组合成显示颜色。荧光点刷新频率很高，可实现 60fps 的刷新率。不同像素数目用分辨率描述，分辨率不足会导致像素走样。内存中有一个帧缓存区，刷新到屏幕中。由于 RGB 值是控制光束电压值的描述，而电压与发光强度非线性关系，处理时需要加入 Gamma 校正，通常取 Gamma 为 2.2。

了解颜色空间后，我们可以更深入地探讨场景的构造。

二、光栅化

光栅化是最早应用于渲染的技术，也是实时渲染的核心方法，无论何时都将被广泛应用。光栅化从相机的视点出发，通过对视锥体中的物体进行裁剪和转换，将其投影到屏幕上，然后将其离散化为一系列像素，经过着色、纹理映射等过程，最终呈现出完整图像。掌握光栅化技术的关键在于对渲染流水线的理解和应用。

渲染流水线大致分为三个阶段。第一个阶段是 CPU 的系统调用阶段，向 GPU 发送渲染指令，这个过程称为 Draw Call。第二个阶段是几何处理阶段，包括模型和视图变换，顶点着色，之后进行投影和裁剪，最终映射到屏幕上。此时，屏幕上显示的是一系列三角形顶点。光栅化阶段将这些顶点转换为像素，首先设置三角形，然后进行三角形遍历，对片元着色器进行着色，最终混合不同的颜色，得到最终的图像结果。

I. 相机模型

在计算机图形学中，常见的相机模型主要有两类：针孔相机模型和透镜相机模型。针孔相机模型是光栅化中常用的模型，而透镜相机模型则用于需要景深效果的场景。

一个相机模型可以通过三个向量来描述：位置（Position）、上方向（Up）和视线方向（Look at）。这三个自由度完全确定了相机的位置和方向。针孔相机类似于小孔成像，视域为一个视锥体，其视野角（Field of View, FOV）决定了视锥体的张角。透镜相机模型则使用光线传播的参数方程来描述，在焦点处光线汇聚最强。景深（Depth of Field）用于描述能够清晰成像的范围，越离焦点远，模糊程度越高。这种模型可以推广到运动模糊，通过子像素分割来实现。

II. 投影

投影过程中有几个关键参数：视平面距离（从相机到投影平面的距离）、近平面和远平面（限定投影区域的前后边界）以及视口的长宽比，这些参数共同定义了视锥体。

投影有平行投影和透视投影两种。平行投影的视锥体是一个长方体，无法表现远近关系；透视投影则将所有直线汇聚到一个点，能够显示物体的远近关系，更接近人眼的视觉感受。

投影矩阵用于将物体坐标转换到标准化设备坐标。平行投影矩阵通过缩放将所有点映射到[-1,1]范围内；透视投影矩阵通过相似三角形将点转换到标准化视锥体。最后，通过视口变换，将标准化坐标转换到屏幕坐标。

III. 裁剪

投影后需要裁剪不可见的部分。裁剪算法有很多，常见的有 Sutherland-Cohen 算法和 Sutherland-Hodgman 算法。

Sutherland-Cohen 算法用于裁剪线段，通过判断线段端点是否在窗口内或外来决定裁

剪。Sutherland-Hodgman 算法用于裁剪多边形，通过边和裁剪平面的位置关系裁剪顶点，形成新的多边形。

在三维空间中，可以通过投影空间的裁剪和齐次坐标中的裁剪来实现。通过比较坐标和齐次坐标的大小关系，判断点是否需要裁剪。

IV. 可见性确定

确定可见性是渲染的重要步骤，涉及背面剔除和深度比较。背面剔除通过法向量和视线方向的点积判断面片是否朝向相机。Z-buffer 算法是最常用的可见性算法，通过深度缓存维护每个像素的深度值，比较新像素的深度决定是否更新。

画家算法按深度从远到近渲染物体，虽然简单，但无法处理复杂的重叠关系。BSP 树（Binary Space Partitioning Tree）通过递归分割空间，建立一个二叉树结构，用于高效的可见性判断和渲染。

V. 图形的光栅化

光栅化是将几何图形转换为像素的过程，常用的算法有 DDA（Digital Differential Analyzer）和 Bresenham 算法。Bresenham 算法通过整数递推避免浮点运算，效率更高。

多边形的扫描转换通过扫描线算法实现，通过扫描线和多边形的交点判断像素的内外关系，逐行着色。边缘填充算法则通过对多边形边界进行操作，实现区域的着色。

VI. 反走样技术

光栅化过程中，走样（aliasing）是一个常见问题。反走样（antialiasing）技术通过提高采样率和滤波来减少走样。

MSAA（Multisample Anti-Aliasing）通过对每个像素进行多次采样，并将结果平均化，实现抗锯齿效果。对于复杂的几何体，通过计算覆盖面积来确定灰度值，从而得到平滑的边缘。

VII. 着色

物体表面与光线的作用决定了最终的渲染效果。常见的材质模型有漫反射（diffuse）、镜面反射（specular）和光泽反射（glossy）。Blinn-Phong 模型将光线的作用分为环境光（ambient）、漫反射光（diffuse）和镜面反射光（specular），通过法向量和光线方向的关系计算出射光强度。

基于物理的渲染（PBR）考虑了光线的反射和折射，通过 Fresnel 项等物理定律，得到更加真实的渲染效果。

VIII. 纹理映射

纹理映射是将图片贴到几何体表面，使其具有丰富的颜色和细节。通过几何体顶点的纹理坐标，在纹理图像中采样得到像素颜色。

在三角形面片上，通过重心坐标插值得到内部点的纹理坐标，再通过双线性插值得到最终的颜色。纹理过滤技术如线性过滤和各向异性过滤可以提高纹理映射的质量。

以上是光栅化过程中的关键步骤和技术，从相机模型、投影、裁剪、可见性确定到最终的光栅化和着色，每一步都至关重要，共同构成了图形学中光栅化渲染的完整流程。

三、光线追踪

光线追踪是离线渲染的核心方法，追求光线追踪的渲染结果一直是离线渲染的重要目标。最原始的光线追踪算法由 Whitted 提出，被称为 Whitted-style Ray Tracing。在引入辐射度量学后，更加真实的 Path Tracing 得到了广泛应用，此外还有 BDPT、MLT 等方法。另一种思路是使用 Photon Mapping，这类方法在处理 Caustic 物体时效果较好。

I. 光线和绘画隐喻

在图形学中，我们将光线抽象成一条射线，从一个点出发并不断延伸。把人眼类比成一个摄像机，当我们看向场景时，许多光线从不同像素出发汇聚到眼睛。如果摄像机是一个画家，光线则汇聚到粗糙的画布上的不同位置。

亚里士多德认为，视觉的产生需要光这一媒介。我们向各个方向发出一系列光，当光线到达世界中的对象时，我们就看到了物体。这一概念称为光线投射，是光线追踪的核心。

假设我们将画布划分成 M×N 的网格，让(0,0)位于左下角，(M-1,N-1)位于右上角。可以找到一种映射，将(x,y)映射到网格上。每个小网格对应一个矩形区域，代表某一块物体的颜色混合结果。取整个矩形区域的中点，与人眼的位置构成向量作为光线矢量，并将其投射出去。此时光线会与场景中的许多物体相交，得到一系列交点。比如与一个球心在原点的球面相交，可以很容易求解方程。尽管这种做法耗时较多，但它是正确的。每条光线代表一个前进方向上的颜色，即它所代表的像素颜色。最后，只需将这个颜色对应放在窗口上，即完成了一幅画的绘制。眼中映射出的世界风景，代表了画上的每一个像素。

海德格尔说，“人诗意的栖居于大地之上”。不仅仅是人，图形学的风景也充满诗意，因为它在屏幕上呈现的风景越真实，越贴近自然的本真，就越富有吸引力。柏拉图提出的三重世界中，理念世界充满意义和价值，现实世界是对理念世界的投影，而画家所作是对现实世界的描摹。可是画本身也是另一个敞开的世界。在画家描述的世界中，人和世界接触，参与其中。图形学也是一样，在屏幕上，人和丰富又充满魅力的世界相接轨，这就是图形学的魅力所在。

II. Whitted-Style Ray Tracing

1980 年，Whitted 将递归式的光线追踪引入图形学，开创了光线追踪技术的先河，因为光线追踪本身得到的是完全正确的结果。

这个过程首先从产生光线开始。从相机出发，向网格化的平面中每个网格发出一系列光线，这些光线会与物体发生碰撞。对于物体来说，由于 BRDF 的特性，光线的作用多种多样。比如，透明材质的物体，光线可以越过表面继续传播，但会发生偏折；金属性光泽的物体，光线碰撞后会反射并向其他方向传播。此时，物体相当于一个次级光源。从次级光源开始传输光线与从初始光源传输光线是完全一样的子问题，可以递归处理。

一般的 BRDF 形式复杂，有时可以用 Lambert 模型和 Blinn-Phong 模型近似。Diffuse 物体可以继续传输，Specular 物体只向某一方向反射。透明表面可以使用 Snell 定律，根据入射方向和法线方向计算出射方向。物体表面可能有不同传输情况，将这些传输结果相加即可得到物体的最终着色。从辐射度角度看，由于反射和折射，光线会不断衰减。可以假定一个最大弹射次数，以达到较好的结果。

III. 光线求交

要正确完成上述算法，光线和物体的求交是不可避免的工作。图形学中，这一过程涉及与多边形求交，首先是与三角形求交。

三角形确定一个平面，用点法式表示。将光线坐标代入点法式，确定光线传输距离，然后判断交点是否在三角形内。Moller Trumbore 优化了这一过程，使用重心坐标描述三角形

方程，通过求解三元线性方程组找到公式。

对于多边形，同样可以求出交点，麻烦在于判断交点是否在多边形内。可以先将多边形和交点投影到一个坐标面，一般选择夹角最小的坐标面，以保证信息丢失最少，同时防止积聚现象。

接下来，按逆时针方向选择边界点。在多边形内的点一定在所有边构成的直线同侧，可根据某些算法计算所有边解析式，将待确定点坐标代入，判断最终符号是否相同。与每个平面都进行计算复杂度高，因此可考虑用盒子确定范围。光线与某个平面相交首先需要与包围它的盒子相交，而与盒子求交复杂性低于平面求交。对于三维盒子，光线需要进出，只需解与六个面相交得到的 t 大小，即可确定是否相交。

IV. 加速结构

光线追踪算法中，耗时最长的是光线和平面求交过程。如果直接遍历，每次光线传输都需遍历场景中所有平面，这一开销难以接受。

一种思路是将空间场景划分。使用这种结构的包括空间八叉树、KD-Tree 和 BSP 树。八叉树将空间分成等大小的 8 份，不断递归划分，形成树状结构。BSP-Tree 前面已介绍，这里不赘述。KD-Tree 依次按水平或竖直方向划分，每个节点先水平分成两份，再在两个子节点中竖直分成两份，构成 KD-Tree。因此，KD-Tree 对空间划分同时很好保持了二叉树性质。KD-Tree 应用存在问题。使用 KD-Tree 意味着需切割物体，增加求解复杂性。KD-Tree 对场景划分均匀，对信息分布均匀场景无足轻重，但对物体分布不均匀场景则难以接受。

BVH 是目前常用的一种加速结构。它递归对盒子进行一系列划分，对场景求交。比如，对三角形网格构成的物体，可根据最长轴方向划分，以中位数三角形为划分标准。这样，场景形成嵌套盒子。盒子之间可能交叠，但物体仅出现在一个盒子里，最终形成树状结构。比起 KD-Tree，BVH 实现容易且效率较好，但每加入一个新物体需重新计算，无法实时更新。

V. Path Tracing

路径追踪是基于辐射度量学发展的改进光线追踪，是现代离线渲染的又一里程碑。首先考虑渲染方程 $L(p,w)=Lr(p,w)+\int f(p,wi,wo)L(p,wr)\cos\theta\, dw$。假设积分符号为 L=Lr+RLr+R^2Lr+……，即考虑多次弹射光线传输。只考虑前几项，即完成了较好的近似。

由于涉及积分运算，直接求解较难，图形学中一般使用 Monto-Carlo 积分。将求解过程转换为采样过程是 Monto-Carlo 方法精髓。但需考虑 BRDF 性质，对任意入射光线，出射光线方向随机。使用 N 条光线采样，出射光线按指数级增长，开销难以接受。因此，可直接选取 N=1。视点到像素可取多条光线，取平均，最终结果相当于遍历大多数情况。同时，常用 Russian Roulette 方法，对表面按某概率打出一条光线，确保递归终点期望不变。

为了缩减方差，即去掉噪点，Arvo 和 Kirk 引入重要性采样。具体来说，生成光线时不使用均匀分布作为概率分布函数，而是根据具体平面性质采样，得到较小方差。由此发展出 MLT 等方法。BDPT，双向路径追踪是另一种优化思路。场景中光源可能只占很小部分，从相机出发的光线不易打到光源，概率较小。为解决此问题，同时从光源出发发射光线，让二者交汇。此方法速度慢，但在某些场景表现较好。

VI. Photon Mapping

如果将光子看成一束能量，包含碰撞交点位置和方向，可建立一颗 KD-Tree 描述这些光子，构成光子图。Jensen 创建了两个光子图，焦散光子图和全局光子图。

焦散光子图标识路径，对漫反射表面，当光线碰撞时可能产生特别图案。光子从光源出发，与表面碰撞按 BRDF 反射，直到遇到漫反射表面，形成焦散光子图。全局光子图解决光

源光子分配问题。光子图沿任意路径，密度较低，用于间接反射。它是 Rendering Equation 的全局解。渲染时，将 Rendering Equation 拆解成光源贡献、漫反射焦散成分、镜面反射辐射度。第一项用全局光子图光亮度估计，第二项从焦散光子图读取，第三项由路径追踪求解，完成 Photon Mapping。

与 Path Tracing 不同，光子映射给出有偏估计，但光子数足够大时收敛到正确结果，即一致估计。

四、几何学

CAD（计算机辅助设计）一直以来都是图形学的重要应用方向，尤其在三维造型方面，它涵盖了图形学中的一个重要范畴。几何学可以大致分为两个主要部分：一个是场景构造，包括 CSG 树、边界表示法和空间八叉树；另一个是参数化曲线曲面的生成，如 Bezier 曲线和样条（Spline）曲线。

I. CSG 树【白书 10、黑书 18】

将物体看作一系列点的集合，可以将集合运算应用于物体之间的关系上，这就是 CSG 树（Constructive Solid Geometry Tree）的基本思想。集合运算包括交、并、差等操作，例如在一个立方体上开一个圆柱形的孔，是立方体和圆柱体做差运算的结果。为了避免出现悬边和悬挂面问题，引入了正则化的集合运算，即先进行运算再求闭包。

在构建一个三维物体时，可以使用 CSG 树来表示操作。树的叶子节点是构成几何体的基本物体，称为基本体素；非叶子节点表示集合操作。通过遍历这棵树，可以得到物体的构造过程。CSG 树一般由五个元素构成：操作类型、坐标变换、基本体、左子树和右子树。建立 CSG 树后，可以递归地求解几何性质。例如，求物体的重心时，可以通过深度优先搜索（DFS）序统计每个小立方体的重心，并将它们组合起来。

另一个问题是光线投射，即显示物体的精确边界。从视点出发，向每个像素投射光线，可以判断最近的可视交点。为解决这个问题，可以先对每个基本体素求交，再对这些交点递归运算，最终找到距离视点最近的交点，得到光线投射的结果。

CSG 树简洁有效，适合表示工程中的基本元素变化，但它无法对物体表面进行修改，且某些操作效率较低，因此常使用边界表示法。

II. 边界表示法【白书 10、黑书 8】

三维物体可以用一个可定向二维流形曲面的三维空间集合来表示，这实际上是将二维流形同构到平面图上。具体来说，这种同构是一个映射，图中的每条边对应物体的一条边，每个顶点形成一个循环。

边界表示法用边界表示物体，对于三维物体则是由外表面的一系列平面构成。顶点、棱边、表面之间相互连接，共有 9 种表示方法。在这种表示法中，物体与物体的关系信息称为拓扑信息，物体本身的特性称为几何信息，二者共同构成物体的信息。

边界表示法常用的数据结构是翼边数据结构，由 Baumgart 提出。翼边结构由边构成，其数据结构包括两个顶点的指针、所在曲线的信息和两个环指针。这两个环分别指边的左侧和右侧的绕环，这是由于多边形的循环性决定的。同时，还需存储左侧循环的第一条边、最后一条边、右侧循环的第一条边、最后一条边的共 4 个指针。

基于翼边数据结构，提出了半边数据结构。半边数据结构考虑到两个循环的不便于维护，将一条边拆分成两部分，确保每条边只位于一个循环内。从拓扑结构上，半边数据结构分为体-面-环-半边-顶点五个层次，通过循环相互沟通。

Sweep 运算（扫掠）是造型中常用的方法，包括平移式、旋转式和广义式三种。平移式

扫掠将平面沿某方向移动产生新几何体，旋转式扫掠将平面绕轴旋转产生几何体，广义式扫掠将平面区域沿任意轨迹移动。

欧拉运算常用于数据结构生成。常见的欧拉运算包括：mvaf 表示输入顶点构造物体；mev 表示输入点连接两个点；mef 表示输入边构造邻接面；kemr 表示删除桥边建立新内环；kfmrh 表示删除原有内环再构造新内环，相当于开通孔。集合运算和局部运算也常用于边界表示法中。

III. 八叉树表示法

边界表示中物体的集合运算相当复杂，效率低下，因此八叉树用于空间划分成为一种策略。八叉树将空间分成 8 份均匀的小格子，递归细分，构成场景的八叉树。每个节点有一个标记：EMPTY（空）、PARTIAL（部分）、FULL（满）。EMPTY 和 FULL 为叶子节点无需继续细分，PARTIAL 则继续递归，直到分割的小立方体边长为单位长度。

建立八叉树后，物体的集合运算转换为对八叉树的结构变换，只需修改树形结构即可实现并交差。隐藏线和隐藏面消除也变得简单，因为八叉树的元素已按空间顺序排列，可在线性时间内找到隐藏线面。

八叉树的主要缺点是占用大量存储空间，一个普通物体的八叉树可能有数千个节点，每个节点需 8 个指针域，开销相当昂贵。可以使用编码优化的一维数组存储八叉树。

IV. 多项式和开花

多项式是参数化构型的基础，对多项式性质的深入讨论，有助于建立丰富多彩的几何世界。多项式一般写成Σat^n 形式，对于线性变换 f(t)，由于 f 是仿射变换，所以可以保持在线性映射中的不变性。

仿射变换的性质可推广到多个参数上，定义多仿射变换为关于 t1,t2,...,tn 的多元函数，含有常数项和一元一次项、二元二次项等的线性组合。例如，三变量仿射映射为 f(t1,t2,t3)=c0+c1t1+c2t2+c3t3+c4t1t2+c5t1t3+c6t2t3+c7t1t2t3。

引入开花概念，任意一个多仿射映射有且仅有一个多项式与之对应，即开花定理。对于 k 元仿射变换,假设 t^q 项的系数为 c(q),对应开花多重映射中,每项的系数为 c(q)/C(k,q)。

V. Bezier 曲线

从 de Casteljau 算法的角度定义 Bezier 曲线。假设有四个控制点，分别为 f(r,r,r)、f(r,r,s)、f(r,s,s)、f(s,s,s)，按照层级插值运算求解 f(t,t,t)的过程即为构造 Bezier 曲线的一个点的过程。

Bezier 曲线具有一定数学形式：F(t)=ΣB(n,i)(t)p，其中 B(n,i)是 Bernstein 基函数，B(n,i)(t)=C(n,i)t^i(1-t)^(n-i)。Bezier 曲线端点是左右两个控制点，端点处切线与 P1P2、Pn-1Pn 平行。曲线具有凸包性、保凸性和变差缩减性。

生成 Bezier 曲线可使用 de Casteljau 方法或公式展开。实际开发中，常使用三阶 Bezier 曲线，涉及连续性问题。在微分几何中，常见的连续性定义有：数值上连续（G0 连续），切向相同（G1 连续），切向相同且大小相等（C1 连续），类似的可定义二阶导数的几何连续（G2 连续）。

VI. B 样条

B 样条没有优雅的几何形式，但与 Bezier 曲线一样，是一系列函数的线性组合。B 样条的基函数定义为：B(i,k)(t)=(t-t(i))/(t(i+k-1)-t(i))B(i,k-1)(t)+(t(i+k)-t)/(t(i+k)-t(i+1))B(i+1,k-1)(t)。B 样条的基函数是分段函数，只在[t(i),t(i+k)]范围

内取正数，其他地方为0。

B样条定义为 ΣPiB(i,k)(t)，也是从一系列控制点出发的样条曲线，具有保凸性、变差缩减性和连续性，但不一定经过控制多边形的端点。渲染B样条可使用de Boor算法，插入操作可用Oslo算法，具有良好可操纵性。l重节点的连续阶不低于k-1-l，使得曲线具有良好的连续性。

VII. 参数化曲面

可用Bezier Surface或B样条曲面完成曲面的参数化。Bernstein基函数表示的曲面形式为 F(t,u)=Σ ΣB(t)B(u)p。交换指标看，可视为对一系列 Bezier 曲线求和，并按Bernstein基函数分配权重。

参数化曲面具有良好性质：四个端点是控制平面的四个端点，边界线是四个边界控制点构成的Bezier曲线。端点的切平面是角落的四个三角形所在平面，并具有凸包性，曲面落在控制平面内。用切线的连续性可实现曲面拼接。

使用B样条基函数也可定义B样条曲面，构成对控制曲面的逼近，具有良好连续性。对于给定曲面，使用Hermite基函数作双三次多项式插值，可得到双三次孔斯曲面，具有许多有趣性质。

五、动画

动画，或称模拟，一直是研究的热门领域。动画是图形学中最广为人知的应用之一，从迪士尼卡通到电影中的各种特效，动画在不同行业中都有广泛应用。某些动画专注于如何模拟最真实的物理效果，如布料和流体的运动。物理基础的动画是动画研究的核心领域之一，质点弹簧系统是其中最简单的系统。对于骨骼和关节系统的模拟，低层运动控制实现起来相对容易，但过程较为复杂；反向运动学则简化了这个过程。粒子系统可以模拟大量粒子的群体行为，而在形变模拟中，常使用Morphing技术。

I. 动画的发展

计算机图形学与艺术密不可分，动画尤其如此，旨在创造一个充满想象的世界。1993年，《侏罗纪公园》使用了大量计算机特效，给观众留下深刻印象。1996年，《玩具总动员》作为首部完全由计算机动画制作的电影，取得了巨大成功。

无论是传统卡通动画还是现代动画，其原理都是利用人类的视觉暂留效果。起初，由于在关键帧之间需要大量手工绘制中间帧，成本很高，计算机插值技术因此应运而生，最早在1964年的贝尔实验室尝试使用，标志着新时代的到来。70年代，随着图形学基础理论的迅速发展，光照模型、纹理映射和三维造型技术相继问世。80年代，商业动画软件大量涌现，动画开始大放异彩并得到广泛应用。Flash技术的出现更是让动画进入千家万户，成为计算机发展史上最具动感的元素之一。

如今，计算机动画已经无处不在。在广告业，动画能做出夸张的表演吸引注意；在影视业，动画不仅用于特效设计，还成为一种独特的艺术形式；在工业界，动画使得装配和几何造型从静态变为动态。

II. 低层运动控制方法

低层运动主要指对底层参数的控制，而非行为本身。无论是低层控制还是高层控制，最终都作用于参数上，但低层控制需要人工操作复杂的数据。

关键帧最早用于运动控制，这一概念在早期动画中得到了普及，最早用于帧与帧之间的插值。具体来说，给出一系列关键参数，然后在时间轴上对这些参数进行插值，从而实现运

动控制。设定好运动轨迹后，可以按照轨迹进行动画制作。通常以弧长为参数，进行迭代积分。许多情况下，用曲线表示运动状态，即弧长对时间的导数变化。在旋转问题中，常使用四元数插值。具体来说，借助 Slerp(q1, q2, u)=sin(1-u)θ/sinθq1 + sin uθ/sinθq2，当u∈[0,1]时，得到一系列值作为插值结果。若多个关键帧的插值需保持较高连续性，可以将样条曲线推广到四元数空间中。

III. Morphing 和空间变形技术

动画中常见物体逐渐变成另一物体的效果来源于 Morphing 技术。1982 年，Brigham 制作了一部短片展示了从女人变成山猫的过程，这是早期的工作之一。Morphing 指从一个状态逐渐过渡到另一个状态，中间帧需保持关键帧特性，需要指定源对象和目标对象的对应关系。空间变形则在保持拓扑结构不变的情况下产生某些扭曲，生成新的物体。

二维多边形变换涉及顶点的对应关系和插值。若 f(t)表示从 A 到 B 的变换，且 f(0)=A, f(t0)=B，最简单的方法是构造线性插值，但容易出现收缩和扭结现象。对夹角和边长进行插值可达到较好的效果。二维图像变形技术复杂但有用。过程需要首先计算出几何元，如网格节点和线段,然后构造两个满射,将第一个图映射到第二个图,再做线性插值和颜色插值，最终得到逼真效果。网格形变是早期应用的技术，但复杂，1992 年引入基于线对的特征指定方法，通过变换让直线发生变形。

三维物体有更多自由度，可实现非常逼真的效果。若两个多边形拓扑结构相同，可轻松插值，但如何找到这种同构性？Kent 等人提出了合并拓扑结构的方法，将物体投影到单位球上,再合并拓扑结构,最后映射回来。Lerios 等人则提出了用体表示实现的三维 morphing。将物体嵌入空间中,使用参数曲面对空间变形,可使物体随之形变。对三维物体,可用 Bezier 超曲面构造 Bezier 体，形成 FFD 块，实现自由变形，这种方法即 FFD 方法。

元球是特殊的隐式曲面,适合描绘光滑物体。多个元球相互融合的过程可刻画复杂动画，既具光滑性又节省空间。

IV. 过程动画

过程动画用过程描述物体的运动和形变，常基于数学模型或物理规律，体现复杂性。

粒子系统用许多微小粒子构建复杂系统，在系统中模拟大量粒子的统计规律，生成复杂效果。粒子有生长和死亡过程，在生命周期中不断形变。Reeves 为粒子系统给出绘制算法，假设不与几何模型相交，覆盖像素的光亮度贡献来自点光源，效率较高。

群体动画描述动物群体运动，即相互靠近同时避免碰撞。Reynolds 提出三条定律匹配群体运动。

布料渲染一直是热点话题。最早的布料通过悬挂在约束点上利用悬链线计算，但无法模拟褶皱。现多用物理方法模拟，考虑材料的各种参数。

水体渲染最简单的方法是用正弦波设置参数，也可用波动理论模拟传播过程。

V. 骨骼动画

抽象人体模型时，会发现控制运动的是许多关节。给人体模型绑上骨骼，控制骨骼就变为控制关节，只需关注运动学模型和动力学模型。

在关节链结构中，刚体的连接点是关节，连接关节的刚体是链杆，起点是根节点，末端是末端影响器。关节链结构的可能形态向量空间即状态空间，由系统自由度决定。许多关节链结构有约束连接，构成复杂模型。

可用 DH 表示法描述关节链结构，由链杆长度、相邻链杆距离、扭角和夹角四参数构成。关节链控制可用正向控制或反向控制。对骨架系统，常构造骨架树描述关节链关系。在骨架

树中，运动控制可用节点对构造单链，在树上更新，逆向运动学可推广。

在骨架上建立肌肉和皮肤，可实现丰富效果。如脸部表情动画，可通过肌肉变化营造各种表情。

VI. 物理动画

传统动画模拟真实物理系统需动画师精准把握生活细节，物理规则推演可实现逼真效果。

刚体运动常见，通过碰撞检测和响应可实现刚体系统。用动力学方程模拟关节链运动，使动画与场景交互。核心问题是碰撞检测，包围盒、八叉树等场景构造方法可加速检测。

质点-弹簧系统常用于描述塑性物体形变。Weil 将其用于布料悬挂过程，后推广至各领域。也有使用连续弹性系统或引入振动系统实现物理模拟。

流体力学是力学重要分支，图形学中用流体运动偏微分方程求数值解，得出流体形状和位置变化。

六、结语

图形学世界丰富多彩，将数据呈现在屏幕上本身是一种创举，是一个全新的世界。无数人通过全新方式表达自己，带来丰富多彩的媒体世界。

渲染、几何和动画共同构成这个多彩世界。有的人为营造虚拟世界的真实感拼尽全力，有的人为构造完美曲线不断努力，这就是图形学的魅力所在。虽然图形学投入大，产出不一定匹配，但只要有人真正热爱图形学，这个世界将更加五彩斑斓。