计算机图形学
Computer Graphics

张思容
zhangsirong@buaa.edu.cn

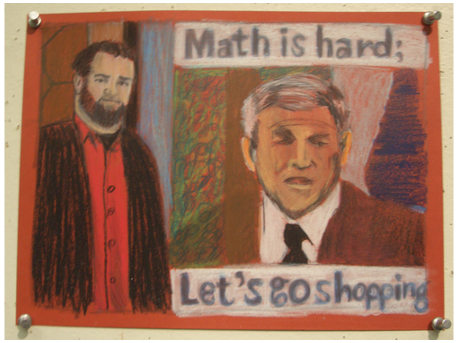
数学与系统科学学院，北京航空航天大学
School of Mathematics and Systems Science, Beihang University

October 11, 2012

引言

自我介绍

- 张思容: Ph.D. 几何分析，医学图像分析；
- 办公时间：周二(12pm-2pm)或预约。 图书馆西配楼501
- 联系方式：134-3920-1025. zhangsirong@buaa.edu.cn
- 欢迎大家学期中提建议和问题，不要最后要成绩!



关于课程：
选修课与研究?
数学还是计算机?
学生需要与难易?

Chapter 1: 计算机图形学简介(2D)

什么是计算机图形学?
计算机图形学
课程大纲
计算机图形学的应用及例子

乌龟画图：LOGO 语言
基本命令的实现
程序与例子

分形
递归画图
分形的性质

2D 仿射变换

2D 仿射几何的画图语言：

计算机图形学与几何建模

- 字面解释： 计算机+图形学
  "计算机画图"
- 计算机：硬件+软件(设备, 编程语言)
- 图形即几何：仿射几何，射影几何，
  解析几何, 微分几何...
- 内容：简单图形, 复杂几何对象，真实图像；(科学数据，计算机艺术，虚拟现实...)
- 课程重点： 计算机编程能力； 几何建模方法；

计算机图形学相关的学科和概念

图形 vs 图像

- 图形：有几何学模型, 可以有不同表示形式; 主要用于计算机辅助几何设计(制造)
- 图像：是bitmap 数据(矩阵), 数学模型: 未知(随机模型?). 主要用于图像处理，模式识别。

相关学科：

- 计算机图形学衍生的分支：
  计算机辅助几何设计CAGD, 计算机绘图drawing, 虚拟现实, 计算机动画，计算机艺术。。。
- 图像科学：图像处理与分析, 模式识别, 人工智能;
- 计算机视觉Computer Vision:
- 计算几何：

课程内容

预备要求：解析几何，线性代数, 计算成熟性, 耐心!
主要目标： 了解计算机图形学基本内容：（计算机编程能力！！！）
学习几何建模方法，（研究基础）
教学参考书:

- (教材)计算机图形学与几何造型导论。Ronald Goldman，邓建松等译，清华大学出版社。
- (教材）计算机图形学基础教程：孙家广,胡事民. 清华大学出版社.
- (参考教材)计算机图形学（opengl版)，Hearn. 电子工业出版社。
- Graphics Gems I-V: 图形学宝藏： 适合课程设计;
- 语言：C 语言大全 Schildt;
  Matlab guide： Higham，SIAM
  openGL 编程指南：红宝书。

课程安排

教学日历：16周： 9/11-12/25

- 第一章：介绍：3周
- 第二章：图形学基础：4周
- 第三章：真实感图形学 3周
- 第四章：几何模型：曲线与曲面 4周
- 课程设计答辩 1周

考核：

- 作业3-4次(包含上机作业)
- 课程设计：1-2人
- 成绩：作业60+课程设计30+课堂参与10=100分

插曲：

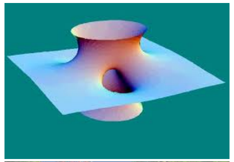
我的图形学课程设计：
万花筒
开花

历史和应用

- 起源： 1962，MIT实验室 Sutherland 博士论文：交互式SketchPad. 法国雷诺汽车公司：Bezier曲面；
- 技术发展： CGI, GKS, PHIGS; openGL：VHML，SIGGRAPH.
- 计算机更新： 八十年代： APPLE II, 九十年代：DOS到WINDOWS; 二十一世纪：Ipod到Ipad(Iphone);
- 应用技术：医学图像；

应用领域

- 生活： UI:人机界面
- 工业：CAGD: 工业设计（Bezier 曲线曲面);
- 娱乐：计算机艺术和动画。
- 科学研究：可视化, 医学图像;



乌龟画图：LOGO 语言

简单的画图语言： LOGO (1966，适合儿童学习，也有高级的几何主题！turtle geometry)
基本命令：

- FORWARD N
- MOVE N
- TURN A
- RESIZE S

其他命令：循环，控制。

MATLAB 实现

数学表达： 乌龟坐标p+方向w

- FORWARD N： $P = P + nw$, 画线；
- MOVE N: $P = P + nw$， 不画线；
- TURN A: $w = w * R_A$,R_A是旋转矩阵；
- RESIZE S: $w = s * w$.

MATLAB 实现 $p = [0, 0]; w = [0, 1]$;

- FORWARD N： $newp = p + dist * w; drawline(p, newp); p = newp;$
- MOVE N: $p = p + dist * w$
- TURN A: $w = w * [cos(\theta)sin(\theta); -sin(\theta)cos(\theta)]$;
- RESIZE S: $w = s * w$.

简单画图程序

- 正多边形：Polygon N
  REPEAT N次
  FORWARD 1
  TURN $2\pi/N$
- 星形：Star N
  REPEAT N次
  FORWARD 1
  TURN $4\pi/N$
- 螺旋形：Spiral N
  REPEAT N次
  FORWARD 1
  TURN $A = 4\pi/5$
  RESIZE $S = 0.9$

复杂例子

- 车轮： 需要三角计算：边长与半径的关系；
  花：计算对角线长度；
- 几何变换： 平移图形 SHIFT N；
  旋转图形 SPIN A
  伸缩 SCALE S
- 迭代与递归：分形；
  Sierpingski 地垫：

乌龟画图LOGO: 从简单到复杂

唯一的画图命令：FORWARD N.
迭代画图：无限重复

- POLY(长度，角度)
  REPEAT FOREVER
  FORWARD 长度
  TURN 角度
- SPIRAL(长度，角度，伸缩)：同上

Lemma (循环引理)
对简单的形状的无限重复得到的形状是基本相似的； 特别的：*POLY*生成的顶点在一个共同的圆周上；*SPRIRAL*在一个共同的螺线上。

见计算机例子。
注：必须用递归得到新的形状！

递归与分形 I:

分形地垫： Sierpinski 三角形

1. 大三角形的角点处画一个小三角形；
2. 移动到另一个角点画一个小三角形；
3. 移动到另一个角点画一个小三角形；
4. 回到起点和起始方向；

MATLAB 实现

function frac_gask(n)
if $n == 0$ $tPoly(3, 2 * \pi/3)$
else for $t = 1 : 3$
$w = tResize(w, 0.5); frac\_gask(n - 1);$
$w = tResize(w, 2);$
$p = tMove(p, w, 1); w = tTurn(w, 2/3 * \pi);$
end end
end

推广：原始形状可以是其他形状等，或递归多边形地垫；

递归与分形 II:

分形曲线：(bump curve) Koch 曲线

1. 画原始 bump curve：
   FORWARD 1 TURN $\pi/3$
   FORWARD 1 TURN $-2\pi/3$
   FORWARD 1 TURN $\pi/3$
   FORWARD 1
2. 用递归调用代替每一个FORWARD 命令。

MATLAB 实现

function frac_bump(n)
if $n == 0$ $p = tForward(p, w, 1);$
else $w = tResize(w, 1/3);$ frac_bump(n-1);
$w = tTurn(w, \pi/3);$ frac_bump(n-1);
$w = tTurn(w, -2 * \pi/3);$ frac_bump(n-1);
$w = tTurn(w, \pi/3);$ frac_bump(n-1);
w=tResize(w,3); end end

注:可以改成其他初始形状。

分形的维数

维数的定义：

- 简单形状： 线段，正方形，立方体；
- 形式定义： 将几何物体线段，面等N等分后,得到的小几何体个数为E； 则维数定义为$D = \log E / \log N$.
  注：验证简单形状；
- 类似定义: $S = 1/N$伸缩因子， $D = -\log E / \log S$。
  特别：对数可以是任何底！

分形的维数：

- Sierpinski 三角形： $D = \log 3 / \log 2 \sim 1.585$
- Koch 曲线： $D = \log 4 / \log 3 \sim 1.262$
- 类似定义: $D = -\log R / \log S$， 其中R是递归调用次数。

思考：Cantor 集的维数?

分形的性质

Proposition (连续不可微)

一般的分形曲线是处处连续，处处不可微的。
例子：Koch曲线，分段连续但无处可微。

Proposition (吸引子)

一般的分形不依赖与初始形状。

- Sierpingski 地垫： 四边形，五边形，线段；
- Koch 曲线： 正方形凸起，正方形（回到终点位置)，

思考： 分形是迭代系统的不动点！！！

几何与变换

变换生成复杂的几何图形：

- 简单形状： 点，线段，正方形，立方体；
- LOGO语言：平移，旋转，均匀伸缩； （对线段变换+对对象做共形变换！）
  刚体变换：保距变换。共形变换：保角变换。
- 图形学的变换：仿射变换：平移，旋转，一般伸缩和错切变换（shear）
  注：错切变换把正方形变成平行四边形；
- 仿射变换：affine transform: 保维数（点到点+线到线）和沿直线和面积比例关系。

注：变换的两个用途：生成几何图形（模型变换)；对几何对象进行变换得到新的对象；
还有视图变换，参见后面的射影变换。

共形变换的数学表示

变换的基本数学对象：点P和向量v,(内部是二维数组坐标表示)
共形变换：

- 平移：$P_1 = P_0 + w$
  严格的 $v = Id * v, P_1 = Id * P_0 + w$
- 旋转：$v_1 = Rot(\theta) * v_0$,$Rot(\theta)$是旋转矩阵。
  一般的关于任意Q点旋转
  有$v_1 = Rot(\theta) * v_0, P_1 = Q + Rot(\theta) * (P_0 - Q)$
- 均匀伸缩： $v_1 = Scale(s) * v_0$,$Scale(s) = [s, 0; 0, s]$是伸缩矩阵。
  一般的关于任意Q点伸缩
  有$v_1 = Scale(s) * v_0, P_1 = Q + Scale(s) * (P_0 - Q)$.
- 注：特别$Q = 0$,则向量和点的旋转和伸缩是一样的。

仿射变换仅仅把旋转和伸缩矩阵推广为任意非奇异线性变换！

仿射变换的表示

变换的基本数学对象：点P和向量v,(内部是二维数组坐标表示)
一般的仿射变换：$v_1 = M * v, P_1 = M * P_0 + w$，$M$是非奇异的线性变换；

- 仿射变换是关于向量的线性变换，但不是点的线性变换；
- 抽象定义：保持点和向量加法的变换
  $A(P + v) = A(P) + A(v)$
- 矩阵形式：$A(P) = M * P + w$
- 几何解释： 线性变换（保维数）+点和向量加法（保比例）

仿射坐标表示：不能用2×2矩阵，用高维！

- 点$P \to (x, y, 1)$; 向量$v \to (v_x, v_y, 0)$
  原因：线性变换+平移
- 矩阵形式：$M_1 = [M, w; 0, 1]$
  $P_1^* = M_1 * P_0^*, v_1^* = M_1 * v_0^*$

一般的仿射变换的表示

共形变换：

- 平移： $Trans(w) = [Idw; 0, 1]$
- 旋转： $Rot(Q, \theta) = [Rot(\theta)Q * (Id - Rot(\theta)); 0, 1]$
- 均匀伸缩： $Scale(Q, s) = [Scale(s), (1 - s)Q; 0, 1]$.

一般情形：矩阵有六个未知数！

- 一点和两个线性无关向量：P, v, w,求方程的逆！
  解释：P, v, w组成的平行四边形的变换决定仿射变换；
- 例子：非均匀伸缩：沿两个方向的伸缩系数不一样； $v, w \to sv, tw$;
  错切：$v, w \to v, w + sv$;
- 不共线三点决定仿射变换：$Affine((P, Q, S); (A, B, C)$.

简单例子

LOGO的缺点：局部几何，无记忆，只能画一个线段；
仿射几何的画图语言：CODO：可以记住很多点，连接任意一对点。

- 对象：点，向量，线段，变换；
- 几何形状构造：$Vector(P, Q), Line(P, Q)$fi$Lines$œ
  变换$Transform(X, M)$,M是仿射变换矩阵；
- 基本变换：向量$Rot(\theta), Scale(s)$;
  点$Trans(v), Rot(Q, \theta), Scale(Q, s), Scale\Phi Q, w, s)$
  $Affine(P_1, P_2, P_3; Q_1; Q_2; Q_3)$
  矩阵：复合$M_1 * M_2$;逆 $Invert(M)$
- 显示：$DISPLAY(X_1, X_2, \ldots, X_n$
  注：点，线段的显示不一样的；
- 说明：CODO没有实现向量的加法等，（可以加入).

例子：多边形；

完全实现：OPENGL语言和其他

OpenGl 实现：
参考：Nate Robins opengl 教程；
问题：使用MATLAB实现二维的仿射几何?