|  |  |
| --- | --- |
| **项目名称：** | **多面体投放算法分析方案** |
|  |  |
| **文 件 名：** | **多面体投放算法分析方案** |
|  |  |
| **文件编号：** | **14D0001-001-001** |

**版本：V.1.0\_\_ 版次：\_1\_**

**制作：王文赛**

**日期： 2014年4月**

# 修改记录

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **修改简述** | **修改者** | **修改时间** |
| 1 | 初始文档 | 王文赛 | 2014-04-09 |
| 2 | 详细描述算法设计 | 王文赛 | 2014-06-13 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 目录

[修改记录 i](#_Toc394363554)

[目录 ii](#_Toc394363555)

[一、 引言 1](#_Toc394363556)

[一.1 编写目的 1](#_Toc394363557)

[一.2 目前理论及技术存在问题 1](#_Toc394363558)

[一.2.1 针对骨料生成的理论和技术问题 1](#_Toc394363559)

[一.2.2 针对重叠检测算法的理论和技术问题 2](#_Toc394363560)

[一.3 参考资料 2](#_Toc394363561)

[一.4 开发环境 2](#_Toc394363562)

[一.4.1 硬件环境： 2](#_Toc394363563)

[一.5 开发的方式 2](#_Toc394363564)

[一.5.1 ADS 3](#_Toc394363565)

[一.5.2 ObjectARX 3](#_Toc394363566)

[一.5.3 AutoLISP 3](#_Toc394363567)

[一.5.4 AutoCAD.NET 3](#_Toc394363568)

[二、 设计目标 4](#_Toc394363569)

[二.1 2D投放目标 4](#_Toc394363570)

[三、 2D投放模型分析 5](#_Toc394363571)

[三.1 多边形面积计算 5](#_Toc394363572)

[三.2 越界检测算法 5](#_Toc394363573)

[三.2.1 矩形边界检测 5](#_Toc394363574)

[三.2.2 圆形边界检测 6](#_Toc394363575)

[三.2.3 等腰梯形的边界检测 8](#_Toc394363576)

[三.3 相交检测算法 9](#_Toc394363577)

[三.3.1 包围盒法 9](#_Toc394363578)

[三.3.2 分离轴测试 11](#_Toc394363579)

[三.4 填充率分析 13](#_Toc394363580)

# 引言

## 编写目的

供开发人员参考，量化项目管理。

## 目前理论及技术存在问题

### 针对骨料生成的理论和技术问题

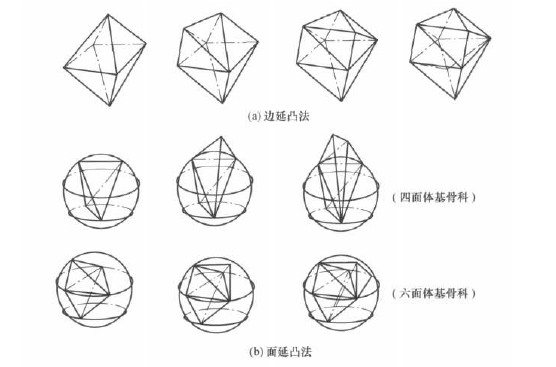
#### 基骨料的构造方法

凸形骨料空间形状复杂，因此很难在三维空间直接生成实现，传统算法是首先在一个虚拟的球体内部构造基骨料，然后通过对基骨料的面或边进行延拓以得到一个成品骨料。对于基骨料一般构造成四面体或六面体。四面体的构造主要是在球面上随机生成4个点组成四面体，通过控制四面体的体积大于某个值，使生成的随机四面体不至于太小，以利于后续的进一步延拓。对于基骨料为六面体的生成方法是首先在过球心的任一平面内生成一个内接正三角形，然后再在该平面所分的上下球面上各随机取一点，组成一个外表面都为三角形的六面体。为了降低对基骨料的延凸次数和限制畸形骨料的产生，上、下两点的选取位置必须符合程序中给定的限制值。

该构造方法，相对比较简单，只能确保骨料的粒径大小，不能确保骨料的尖锐度，而且生成的六面体形状比较单一，不能真实的虚拟出显示骨料形状。

#### 骨料的生长方式

当完成构造过程的第一步，即基骨料生成后，对基骨料有两种方式进行延凸 ，将其分别称之为边延凸法和面延凸法。边延凸法每次以延凸后的最长边为延凸对象，面延凸法每次以延凸后的最大面为延凸对象。其骨料生长方式如图2.3-1所示。边延凸和面延凸均有整体凸性限制，即新生成点与原凸多面体组成的新多面体如果失去凸性，则舍弃该点，重新生成新的延凸点。毫无疑问，该法需要频繁判断和舍弃，新生成的顶点不能一次到位，使得程序效率低下。



2.3-1 骨料生长方式

### 针对重叠检测算法的理论和技术问题

图形游戏中常用包围盒法进行碰撞检测，优点是速度快，缺点是精确度非常低。采用分离轴测试法进行碰撞检测的优点是简单、易于实现且结果可靠，但缺点是对于复杂结构进行分离轴测试法检测相交效率较低。

本系统采用划区域投放法并行投放，并采用包围盒法做快速初次判断，再以分离轴测试法进行精确判断，大幅提升了碰撞检测的效率上界。

## 参考资料

《基于matlab-混凝土二维细观结构数值模拟骨料随机投放》

《三维空间任意多面体体积的一种坐标计算法》

《蒙特卡罗法生成混凝土随机骨料模型的ANSYS实现》

《一种快速的三维凸型混凝土骨料随机投放算法》

《混凝土细观分析中随机多边形骨料生成方法》

## 开发环境

### 硬件环境：

内存：4G 或更大

CPU：Intel(R) Core(TM) i5-241 或更高配置

硬盘：50GB或更大

显卡：显卡缓存在512M以上

显示器：1,024x768 VGA  真彩色

#### 软件环境

操作系统：Window 7 32或64位系统 或 Window 8

开发工具：Visual Studio 2008 + Auto Cad 2010 或 Visual Studio 2012 + Auto Cad2014

## 开发的方式

AutoCad第一版本于1982年11月推出，其二次开发也在不断的涌现（见图1.3.1 开发方式时序图，图1.3-2开发方式原理）



图1.3-1 开发时序图



图1.3-2开发方式原理

### ADS

ADS(AutoCAD Development System)是AutoCAD的C语言开发系统，ADS本质上是一组可以用C语言编写AutoCAD应用程序的头文件和目标库，直接利用该用户熟悉的各种流行的C语言编辑器，将应用程序编译成可执行文件并在AutoCAD环境下运行，这种可以在AutoCAD环境中直接运行的可执行文件叫做ADS应用程序。ADS由于其速度快，又采用结构化编程体系，因而很适合于高强度的数据处理。

### ObjectARX

ObjectARX(AutoCAD Runtime eXtension)是一种崭新的开发AutoCAD应用程序的工具，以C++ 为编程语言，采用先进的面向对象的编程原理，提供可以AutoCAD直接交互的开发环境，能使用户方便快捷地开发出高效简洁的AutoCAD应用程序。ObjectARX能够对AutoCAD的所有事务进行完整的，先进的，面向对象的设计与开发，并且开发的应用程序速度快，集成度高，稳定性更强。

### AutoLISP

AutoLISP基于简单易学而有功能强大的LISP(List Processer)编程语言，由于AutoCAD具有内置的LISP解释器，AutoCAD能够直接读取AutoLISP代码，无需编译。通过创建AutoLISP程序，可以向AutoCAD添加专用命令。

### AutoCAD.NET

从AutoCAD2006开始，Autodesk为其开发增加了.NET API。.NET API提供了一系列托管的外包类，使得开发人员可以在Microsoft.NET Framework下，使用任何支持.NET的语言，如VB.NET,C#和Managed C++等对AutoCAD进行二次开发。其优点是完全面向对象，方便易用，是相对比较理想的AutoCAD二次开发工具。

# 设计目标

## 2D投放目标

生成随机凸多边形，并随机投放在给定的矩形容器中，，且满足：

1.多边形为存在外接圆的凸多边形

2.多边形外接圆、内角大小应可以控制在指定范围内

3.多边形在容器中服从均匀分布

4.多边形之间不能相交、重合或互相包含

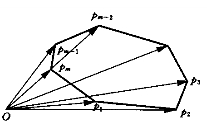
5.多边形不能超出容器边界

6.多边形的填充率，即所投放的满足要求的多边形的面积总和占矩形容器面积的百分比，达到60%±5%。

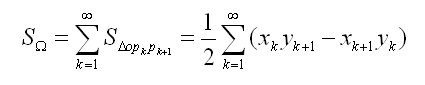
# 2D投放模型分析

## 多边形面积计算

设Ω是m边形(如下图)，顶点1343722850_1102沿边界正向排列，坐标依次为1343722878_8349



建立Ω的多边形区域向量图。由图知坐标原点与多边形任意相邻的两个顶点构成一个三角形，而三角形的面积可由三个顶点构成的两个平面向量的外积求得。由此可得任意多边形的面积公式：



## 越界检测算法

对每一个多边形，判断其所有顶点坐标是否均在容器所围区内。因此越界检测简化为一个坐标点是否在容器所围成的区域内的问题。

### 矩形边界检测

一个点相对于矩形的边界有如图2.1所示的几种情况。



图2.1 点相对于矩形边界的位置

1. **点Pa在矩形P0P1P2P3内**

Pa在四条直线X=X0、X=X2、Y=y0、Y=y1围成的内部区域，坐标可表示为Xa>X0且Xa<X2且ya>y1且ya<y0

1. **点Pb在矩形P0P1P2P3边界上**

Pb在四条直线X=X0、X=X2、Y=y0、Y=y1围成的四条线段上，四条线段分别为P0P1，P1P2，P2P3，P0P3，

Pb在P0P1上坐标可以表示为：Xb == X0 且 yb>y1 且yb<y0

Pb在P2P3上坐标可以表示为：Xb == X2 且 yb>y1 且yb<y0

Pb在P1P2上坐标可以表示为：yb == y1 且 xb>x0 且xb<x2

Pb在P0P3上坐标可以表示为：yb == y0 且 xb>x0 且xb<x2

综合表示为：

（Xb == X0 且 yb>y1 且yb<y0）或（Xb == X2 且 yb>y1 且yb<y0）或（yb == y1 且 xb>x0 且xb<x2）或（yb == y0 且 xb>x0 且xb<x2）

1. 点Pc**在矩形P0P1P2P3边界外**

Pa在四条直线X=X0、X=X2、Y=y0、Y=y1围成的内部的外部区域，坐标可表示为Xa<X0或Xa>X2或ya<y1或ya>y0

### 圆形边界检测

一个点相对于圆形的边界有如图2.2所示的几种情况。



图2.2点相对于圆形边界的位置

1. **点Pc在圆内**

Pc在以Pr为圆形，R为半径围成的内部区域。即Pc到Pr的距离小于半径R，可表示为

点Pb在圆的边界上

Pb在以Pr为圆形，R为半径围成的内部区域。即Pb到Pr的距离小于半径R，可表示为

点Pd在圆的边界外

Pd在以Pr为圆形，R为半径围成的内部区域。即Pd到Pr的距离小于半径R，可表示为

### 等腰梯形的边界检测

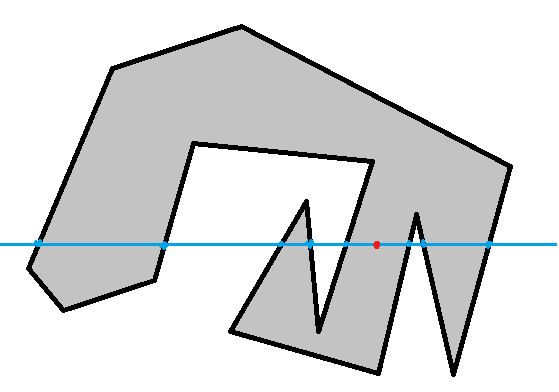


图2.3 点在凹多边形内部

如图2.3显示一个具有14条边的凹多边形。判断红点Pa（Xa,Ya）是否在多边形内。测试点Pa

的Y坐标与每一个点进行比较，得到一个测试点所在的行与多边形的交点列表，如果测试点的两边点的个数都是奇数个则该测试点在多边形内；否则在多边形外。在上图中有8条边与测试点所在的行相交，而有6条边没有相交。测试点左边有5个交点，右边有三个交点，它们都是奇数，所以该测试点在多边形内。

结论：已知测试点Pa（Xa,Ya），直线Y=Ya与多边形的各边按逆时针（或顺时针）排序组成的点集合为U。如果Pa两侧点的个数是奇数，那么Pa点在多边形内部；否则在多边形外。

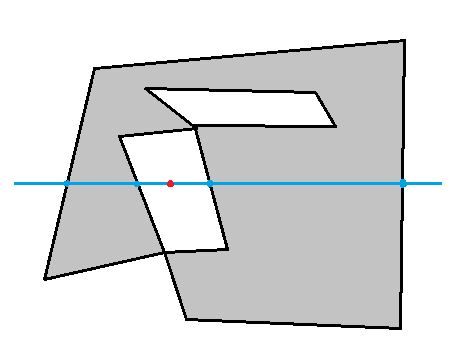


图2.4 点在自交多边形内部图形

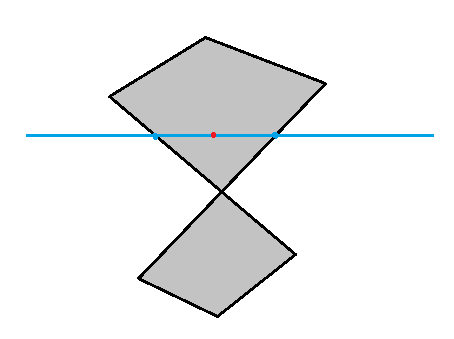


图2.5 点在没有重叠但有两条边相交的多边形内

图2.4，图2.5所示的情况，根据上述结论，也能得出正确的结果。因此可用此结论进行判断一个点是否在梯形内部。

## 相交检测算法

### 包围盒法

包围盒算法是一种求解离散点集最优包围空间的方法。基本思想是用体积稍大且特性简单的几何体（称为包围盒）来近似地代替复杂的几何对象。

最常见的包围盒算法有AABB包围盒（Axis-aligned bounding box），包围球（Sphere）， 方向包围盒OBB（Oriented bounding box）以及固定方向凸包FDH（Fixed directions hulls或k-DOP）。

AABB是应用最早的包围盒。它被定义为包含该对象，且边平行于坐标轴的最小六面体。故描述一个AABB，仅需六个标量。AABB构造比较简单，存储空间小，但紧密性差，尤其对不规则几何形体，冗余空间很大，当对象旋转时，无法对其进行相应的旋转。处理对象是刚性并且是凸的，不适合包含软体变形的复杂的虚拟环境情况。

对象的包围球被定义为包含该对象的最小的球体。确定包围球，首先需分别计算组成对象的基本几何元素集合中所有元素的顶点的x，y，z坐标的均值以确定包围球的球心，再由球心与三个最大值坐标所确定的点间的距离确定半径r。包围球的碰撞检测主要是比较两球间半径和与球心距离的大小。

OBB是较为常用的包围盒类型。它是包含该对象且相对于坐标轴方向任意的最小的长方体。OBB最大特点是它的方向的任意性，这使得它可以根据被包围对象的形状特点尽可能紧密的包围对象，但同时也使得它的相交测试变得复杂。OBB包围盒比AABB包围盒和包围球更加紧密地逼近物体，能比较显著地减少包围体的个数，从而避免了大量包围体之间的相交检测。但OBB之间的相交检测比AABB或包围球体之间的相交检测更费时。

FDH（k-DOP）是一种特殊的凸包，继承了AABB简单性的特点，但其要具备良好的空间紧密度，必须使用足够多的固定方向。被定义为包含该对象且它的所有面的法向量都取自一个固定的方向(k个向量)集合的凸包。FDH比其他包围体更紧密地包围原物体，创建的层次树也就有更少的节点，求交检测时就会减少更多的冗余计算，但相互间的求交运算较为复杂。

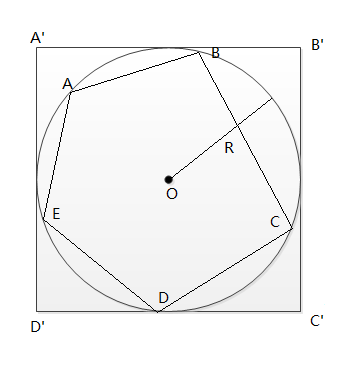


图3.1包围盒法检测示意图

### 分离轴测试

相交检测根据分离轴定理（Separating Axis Theorem）采用分离轴测试(Separate Axis Testing)算法。

分离轴测试(Separate Axis Testing)是空间碰撞检测中一个很常用、很有效的判断方法。它的原理来源于集合分析中的“分离超平面定理”（separating hyper-plane theorem）：如果两个集合A和B不相交，那么必定存在一个分离超平面P，并使得A和B分别位于P的不相同的两侧。此定理对于空间数据中的凸集合和凹集合均适用。当数据集是凸集时这里的分离超平面就是一个空间的平面；当数据集是凹集时，这里的分离超平面则不再是一个空间的平面，而是一个空间的曲面。

这个方法是碰撞检测的核心，也非常快并且非常可靠，因为计算中没有使用除法操作，下面(如图3.2 矩形分离轴测试)给出一个简单的基于两个矩形的碰撞检测的例子。

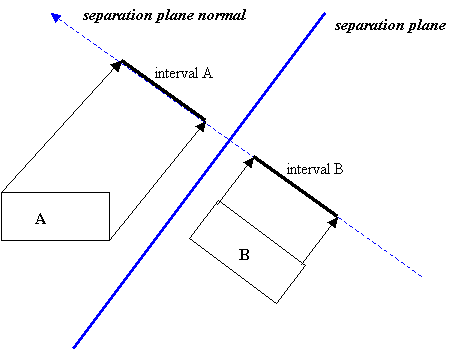


图3.2 矩形分离轴测试

算法试图在两个物体之间找到一个合适平面，如果这个平面存在，那么物体就没有相交。

为了测试物体是否是分开的，简单的方法是投影这个物体到平面的法线上，并比较两者之间的间距看二者是否重叠。显然有无数的平面可以用来分割两个物体，但已证明只需要使用一部分平面来进行测试，对于矩形从上图中可以看出平面的法线为矩形 B的长轴。

对于矩形来说需要测试的分割平面是那些法线等于两个矩形的轴向的平面。因此对于两个矩形来说，只需要测试4个分割平面即可。在这四个平面里，一旦发现一个分割平面可以分割矩形那么就可以断定这两个矩形是不相交的。如果四个平面都不能分割矩形，那么这两个矩形一定是相交的，也就是出现了碰撞。

类似可将该算法推广到一般凸多边形，只是需要测试的平面数量不同。分割平面在每个多边形边的垂直方向上有一个法线。如图3.3，两个分割平面用于测试，在红色的平面上两个间隔是重叠的。然而在蓝色的平面上间隔是不重叠的，因此，蓝色的平面的是分割平面，因此这两个三角形是不相交的。

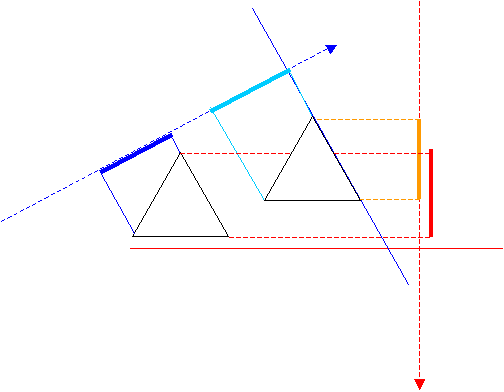


图3.3 两个分割平面分离轴测试

对于两个相交的由多面体组成的凸集，它们之间的相交方式可以归结为以下几种：面－面、面－边、面－点、边－边、边－点、点－点相交，而对于其中的点，又可以将它们处理为退化的边，这样一来就可以简化两个凸集间的相交方式为：面－面、面－边、边－边相交。对于上述三种简化的相交方式，我们就可以逐条考虑进而找到所有潜存的分离轴，这样我们就可以得到两个凸集之间潜在的分离平面可能存在于以下位置：

1.凸集A中每个多边形所在的平面。

2.凸集B中每个多边形所在的平面。

3.凸集A中的每条边和凸集B中的每条边之间的公共垂面。

而对应的分离轴的方向为：

1. 凸集A中每个多边形的法向量。

2. 凸集B中每个多边形的法向量。

3. 凸集A中的每条边和凸集B中的每条边之间的公共垂面的法向量，即两条边方向向量的叉乘方向。

假设凸集A中含有FA个面、EA条边，凸集中含有FB个面、EB条边，这样就存在FA + FB + EA \* EB条潜在的分离轴，某些情况下这个数量是巨大的，因而在实际情况中我们往往需要再根据具体几何体元的空间特征对其进行简化。比如，对于两个OBB之间的相交测试，由于OBB的六个面中两两相对的面是平行的，这样我们就可以将面－面之间的分离轴数目减少一半，同样也可以将边－边之间的分离轴数目减少一半，如此一来情况就简单了很多。

    总结上述内容，可以得到两个凸集之间用分离轴测试进行相交判断主要有以下几个步骤：

   1．根据凸集的具体情况分析它们之间潜在的所有分离轴，并针对每条分离轴进行分离测试。

   2．对于每条分离轴L，做投影计算得到d、RA、RB，并根据它们之间的关系得到分离关系。若d > RA +RB则分离平面存在，即两个集合之间不相交，此时可以及时退出判断程序；反之，这两个集合在此分离轴下没有分离，但却也不能说明两个集合相交。

   3．若对于所有潜在的分离轴判断的结果无法证实分离平面的存在，那么这两个凸集必定相交。

## 填充率分析

填充率R由所有已投放的多边形面积之和Sp除以容器总面积Sc得到。

由多次实验得到，本系统在一分钟内，以边长为1的四边分别与容器的边界平行的单位正方形随机投放到矩形容器中，可达到98%的填充率。

以随机多边形投放，一般一分钟内可达到65%的填充率，这已经是相同课题目前可以达到的较高水准。调整多边形外接圆直径的最小值后填则充率最高可达70%。

在满足多边形均匀分布的情况下，填充率与算法运行时间成正比。显然，随着填充率的增大，投放成功的概率呈减小趋势，这使得填充率在随即投放算法下是有上界的，且此上界因受多边形边数、形状、大小、坐标、容器大小和形状等因素的影响而变化。