

# 一种基于三维 Delaunay 三角化的 曲面重建算法

袁 方, 唐 杰, 武港山

(南京大学 软件新技术国家重点实验室, 江苏 南京 210093)

**摘 要:**提出一种基于三维 Delaunay 三角化的区域增长式曲面重建方法。该方法以空间点云的 Delaunay 三角化为基础, 结合局部区域增长的曲面构造, 较以往方法具有人为参与更少、适用范围更广的优点。算法采用增量式插入点的方式构建空间 Delaunay 划分, 采用广度优先算法, 以外接圆最小为准则从 Delaunay 三角化得到的四面体中抽取合适的三角片构成曲面。该算法的设计无须计算原始点集的法矢, 且孔洞系数对重建的结果影响很小, 重建出的三角网格面更符合原始曲面的几何特征。无论待建曲面是否是封闭曲面, 本算法均可获得较好的重建效果。

**关键词:**曲面重建; 三维 Delaunay 三角化; 三角网格; 广度优先

**中图分类号:** TP391

**文献标识码:** A

**文章编号:** 1673-629X(2011)10-0014-05

## A Geometric Spread Approach of 3-D Reconstruction

YUAN Fang, TANG Jie, WU Gang-shan

(National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)

**Abstract:** It introduces a method of 3D surface reconstruction. As many other algorithms of this kind, it produces a piecewise linear approximation of a surface  $S$  from a finite, sufficiently dense, subset of its points. It is based on the 3D Delaunay triangulation of points, and uses the local area incremental algorithm to build the triangular mesh. Build up the 3D Delaunay triangulation through incremental insertion of points, and use breadth-first algorithm to pick the appropriate facets to reconstruct surfaces. The algorithm doesn't need pre-processing of input data and nearly user input independent. Present some results of the method, which turn out to be good on surfaces with and without boundaries.

**Key words:** surface reconstruction; 3D Delaunay triangulation; triangular mesh; breadth-first

## 0 引 言

在三维模型重构中, 一个基本的问题是, 如何从物体表面的散乱三维数据点快速构建物体表面的三角网格模型一称之为曲面重建问题。即, 给定一个点集, 点集中的点分布在一个未知的曲面上, 根据给定点集构造出该未知曲面的分段线性近似曲面。

曲面重建问题在计算机图形学和计算几何学上已经得到了广泛的关注并被大量的研究<sup>[1]</sup>。在计算机图形学中, 曲面重建的算法通常分为三类: 基于距离场等值面抽取的曲面重建算法、基于三维 Delaunay 三角划分的曲面重建算法和基于局部区域增长的重建算法。

H. Hooper 的重建算法<sup>[2]</sup>属于第一种: 由  $K$  邻近点估计切平面, 法向一致性调整后构造距离场, 最后由优化的 Marching Cube 算法<sup>[3]</sup>提取等值面获得重建曲面。该算法涉及到复杂的法向一致性调整和等值面抽取, 因此相当耗时。Green and Sibson<sup>[4]</sup>实现了二维空间的 Voronoi 图的计算及 Delaunay 三角化<sup>[5]</sup>后, Bowyer<sup>[6]</sup>和 Watson<sup>[7]</sup>将二维 Delaunay 三角化推广到高维, 进而出现不少利用 Delaunay 三角化重建表面的算法, 如: Edelsbrunner<sup>[8]</sup>的  $\alpha$ -shape 算法, 他扩展了凸包的概念, 以  $\alpha$  球界定 Delaunay 三角化后的凸包, 删除包围球或外接圆半径大于  $\alpha$  的四面体、三角形和边获得重建表面。但该算法很难事先估计合适的  $\alpha$  半径。另外 Raphaëlle Chaine<sup>[9]</sup>的 SHRINK 算法也是基于三维的 Delaunay 三角化, 他构造一张伪表面, 该伪表面从空间三维点集的 Delaunay 三角化闭包开始收缩, 最终逼近原始曲面。该算法在闭合曲面的重建方面表现优秀, 但是在重建非闭合曲面时效果并不理想, 而且不管待重建曲面是否是闭合曲面, 该算法都需要输入合

收稿日期: 2011-03-06; 修回日期: 2011-06-15

基金项目: 国家高技术研究发展计划(863)(2007AA06A402)

作者简介: 袁 方(1986-), 男, 硕士研究生, 研究方向为计算机图形学、三维建模; 唐 杰, 博士, 副教授, CCF 会员, 研究方向为计算机图形学、三维建模; 武港山, 博导, CCF 会员, 研究方向为多媒体技术。

适的孔洞系数,孔洞系数也是很难事先估计的。近來国内也有利用基于投影的局部二维 Delaunay 三角化重建表面,但这些算法都涉及到复杂的网格拼接和法向一致化调整,大大增加了表面重建的计算量。另一类表面重建算法是基于增量式的思想,从一个基准三角面片开始,递增的扩展获取整个表面,如 Bernardini<sup>[10]</sup> 的 Ball Pivoting 算法,但这类算法常常需要对原始点集做预处理,并且需要额外的输入,因此并没有得到广泛的应用。

Hoppe 等人的算法使用的是曲面拟合的方式,重建出的曲面并不严格地过原始点集,因此一些尖锐特征无法很好的重现;且对一些复杂模型的表面(如下文图 3 中的模型表面),拟合式的算法无法重建出正确的曲面。基于三维 Delaunay 三角化的重建算法基本上都能很好地重现原始曲面的尖锐特征,且在时间复杂度上要优于拟合式的重建算法,但是现有的这类算法通常无法很好地重建非闭合曲面。

文中提出了一种基于三维 Delaunay 三角化的曲面抽取算法。该算法先将三维点集做三维 Delaunay 三角化,得到划分三维空间的四面体集合以及组成四面体的有向三角片,从这些有向三角片中选出外接圆半径最小的三角片作为种子三角片,使用广度优先原则递增地蔓延式地获得整个曲面。该算法相比 SHRINK 算法无需精确地估计孔洞系数,只要孔洞系数落入一个合理的区间即可,且在非闭合曲面重建方面效果更佳;而相比 Ball Pivoting 算法则省去了数据的预处理和对合适的输入球半径的估计。

## 1 基于三维 Delaunay 三角划分的曲面重建

### 1.1 三维 Delaunay 三角划分

二维平面上的 Delaunay 三角化的严格定义是:一个三角集合  $T = \{T_i\}$ , 其中  $i = 1, 2, \dots, N$ , 称为平面上不共线有限离散点集  $P$  的三角化, 若其满足: (1)  $T$  的顶点集为  $P$ ; (2)  $T$  的并集为  $P$  的凸包; (3)  $T$  中任意两个三角形的交集为空集, 或为  $P$  中的点, 或为以  $P$  中两点为端点的一条边; (4)  $T$  中每个三角形的外接圆的内部除三角形的三个顶点外不包含点集  $P$  中的其他点; 则称其为离散点集  $P$  的 Delaunay 三角化。二维 Delaunay 三角化的规则可以简称为“空圆准则”。

扩展到三维空间, Delaunay 三角化的规则变为“空球准则”, 即划分空间的每个四面体的外接球内部不包含其他顶点。三维 Delaunay 三角化又称为 Delaunay 四面体化。

二维 Bowyer/Watson 算法通过先生成给定区域的大三角化(亦可能从包围盒开始), 然后递增地向其中插入新点。当新插入点落在一些现有三角片的外接

圆内时, 这些三角片就不再是 Delaunay 三角片了, 首先将其删除。由于剩余三角片依然还都是 Delaunay 三角片, 只需在插入点周围形成的空多边形内增加新的三角片, 分别连接插入顶点与空多边形各顶点, 即可形成新的 Delaunay 三角化。重复上述操作直至所有顶点均插入到三角化当中。而三维的 Delaunay 三角化算法复杂性远远超过二维算法, 只有少数二维算法可直接推广到三维算法, 如抛物线投影算法, 但其计算复杂度相当高并不适合实际应用。因此三维的 Delaunay 三角化常常需要用到第三方计算几何算法库如: CGAL (Computational Geometry Algorithms Library), 它是欧洲和以色列的八个科研机构在欧盟项目框架下联合开发的 C++ 几何算法库。

### 1.2 曲面的构建

空间点集经过 Delaunay 四面体化后得到大量的三角片, 基于三维 Delaunay 三角化的曲面重建算法一般都是采用一些规则来去除冗余的三角片, 剩下的三角片即组成了待建曲面的三角网格面。Raphaëlle Chaine<sup>[9]</sup> 的 SHRINK 算法即属于这一类, 它以空间点集的三角化闭包为初始三角网格面, 若网格上的三角片不满足 Gabriel 性质或三角片外接圆半径大于用户输入的孔洞系数, 则用其所在四面体中的其他三个三角片代替该三角片。如此迭代进行, 若待建曲面是闭合物体的表面, 则当网格曲面不能再收缩时, 即为最终结果曲面; 如果待建曲面是有边界的曲面, 则最终共三点且方向相反的两个三角片都保留下来时, 其中一个三角片才被保留下来作为结果曲面的一部分。

重建曲面中的三角片始终是从三维 Delaunay 三角化的三角片中得来的, 因此从这些三角片中抽取合适的三角片来组成待建曲面也是一个可行的办法。Amenta<sup>[11]</sup> 的 Power Crust 算法是这方面较著名的算法, 但该算法要判定空间单元的内外性, 因此根本无法用来重建非闭合的曲面。

## 2 算法概述

采用去除冗余三角片方式的重建算法中, 最重要的显然就是去除规则的设定, 然而现有的这类算法的冗余三角片去除规则都无法保证重建效果, 例如 SHRINK 算法的重建结果中常常有四面体的存在。通过严格控制新加入三角片和已有三角片之间的拓扑关系, 采用抽取三角片方式的重建算法可以很好地避免这种情况的发生。文中算法先将空间点集做三维 Delaunay 三角化, 得到组成四面体的有向三角片集合, 三角片的方向指向四面体内部。构建曲面时采用抽取合适三角片的方式, 抽取出的三角片应当是包含所有原始点的、流形的(即每条边只能被不超过两个三角

片包含,三角片和三角片仅在边上相交)并且三角片方向一致的。

### 2.1 包含有向三角片的三维 Delaunay 三角化

文中使用 Edelsbrunner 和 Shah 的 Incremental flip<sup>[12]</sup>算法来做三维 Delaunay 三角化。Incremental flip 算法可以描述如下:  $S$  是三维空间中的点集,令  $4 \leq i \leq n$ ,假设  $S$  中的前  $i$  个点的 Delaunay 四面体化已经构建好,称为  $D(i-1)$ ,添加第  $i$  个点到  $D(i-1)$  中,对新加入的四面体做相应调整使其满足“空球准则”,得到  $D(i)$ ,重复这一过程直到  $i = n$ 。

#### 2.1.1 数据结构

点 point 的数据结构:三个 double 型数表示点的  $X$ 、 $Y$ 、 $Z$  坐标。

四面体 tetrahedron 的数据结构:四个指向顶点的指针;四个指向邻接四面体的指针,无相邻四面体时相应指针为空;四个指向所包含的面的指针。设组成四面体四个点为  $v_0, v_1, v_2, v_3$ ,则四个有向三角片分别为  $f_0(v_0, v_1, v_2), f_1(v_0, v_3, v_1), f_2(v_1, v_3, v_2), f_3(v_2, v_3, v_0)$ 。

三角片 triface 的数据结构:一个指向所属四面体的指针;int 型整数 loc(取值范围 0~3),用来标记是所属四面体的第几个有向三角片。该数据结构同时被设计用来表示有向边,因此它还包含了一个 int 型整数 ver(取值范围 0~5),用来标记是所属三角片的第几条有向边。

#### 2.1.2 算法步骤

步骤 1:读入所有点,选取前四个不共面的点构成第一个四面体。

步骤 2:取新的点,判断该点的空间位置,若该点在所有已有四面体的外部,则执行步骤 3;若该点在某一个四面体内部,则执行步骤 4;若该点落在某一个三角片上,则执行步骤 5;若该点落在某一条边上,则执行步骤 6;若和已有顶点重合,则丢掉该点,重新执行步骤 2。

步骤 3:找到离该点最近的三角片,以该点和最近三角片构建新的四面体。转到步骤 7。

步骤 4:将点在其内部的四面体分裂成为四个新的四面体,转到步骤 7。

步骤 5:将点在其上的三角片,和与该三角片共三点方向相反的三角片所在的四面体分别分裂为三个新的四面体。转到步骤 7。

步骤 6:将所有包含点在其上的边的四面体分别分裂为两个新的四面体。转到步骤 7。

步骤 7:使用 Lawson<sup>[13]</sup>算法检查是否有不符合“空球准则”的四面体的存在,若有,则加以调整。转到步骤 8。

步骤 8:如果所有点都已加入,则算法结束,否则

转到步骤 2。

### 2.2 区域增长式的三角网格重构

为了能够最终得到一个尽量光顺的曲面,在增长式地抽取下一个有向三角片时似乎应选择和当前有向三角片法向之间的夹角最接近 0 的有向三角片。但是如果这样,可能会发生以下情况(见图 1 和图 2)。

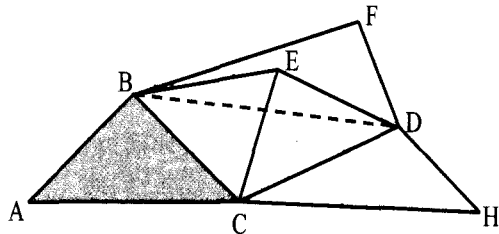


图 1 三角片的延伸示意

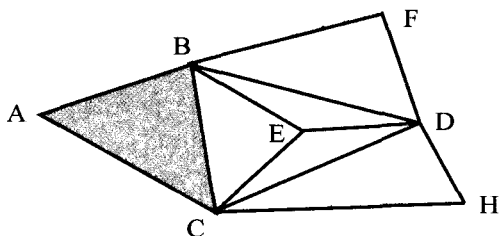


图 2 空间三角片在近似平面的投影

图 1 中阴影三角片 ABC 为当前处理三角片,挑选以边 BC 相邻的三角片时遇到四面体 BCDE,按照夹角最接近 0 的原则,应该选择三角片 CDB。而当处理三角片 CBD 时,则应该选入三角片 BDF 和 CHD,如此,点 E 最终可能不会包含在结果三角网格曲面中,而这无疑是不符合原始点集的几何特征的。因此在挑选下一有向三角片时,和当前有向三角片夹角最接近 0 并不适合作为首要原则。

在曲面重建的过程中,经常会用到一个近似,即:在一阶临近点的范围内,将曲面近似地看作平面。H. Hoop 的重建算法中用 K 邻近点估计切平面时即是基于这一近似。文中算法也使用这一近似。重新来看图 1 中的情况,把所有点和边投影到一个近似的平面上,如图 2 所示。点 E 落在了三角形 BCD 内部,而显然,三角形 BCE、CDE、BDE 加起来相比三角形 BCD 能够表达更多的几何特征信息。

因此,为了更好地重现原始点集的几何特征,且在参考了 Ball Pivoting 算法之后,文中认为在使用增长式曲面构造方法时,应以邻接有向三角片中外接圆最小的有向三角片作为下一个选入的有向三角片。当然,选入的有向三角片的法向和当前处理的三角片的法向之间的夹角还应满足一定条件,以保证结果曲面中的三角片的方向的一致性,文中算法以相邻有向三角片法向间夹角不大于  $2\pi/3$  作为条件。

#### 2.2.1 算法步骤

在三维 Delaunay 三角化的数据结构的基础上,建

立于广度优先处理的队列,其中存放的是待扩展的三角片的有向边。有向边构成的有向环即表示了已有三角网格的当前边界,有向环向外扩展,直到环中所有的边变为边界边(待建曲面是有边界的曲面),或者环变为空(待建曲面为闭合物体表面)时,重建过程完成。

以下是相关伪代码:

Algorithm DelaunayChoose

```

1: chooseFirstTriangle(); // 找到外接圆最小的三角片作为起始三角片,三角片三条有向边构成最初的有向环,将三条有向边放入队列
2: while( 队列非空)
3:  $e_{ij} = \text{pop\_front}()$ ; // 从队列头弹出一条边作为处理对象
4:  $\sigma_k = \text{findTheSmallestPoint}(e_{ij})$ ; // 找到满足条件的三角片中的第三个点
5: if( not_used( $\sigma_k$ ) )
6: join( $e_{ij}, \sigma_k$ ); // 结果曲面中加入新的三角片,且将两条新边加入到队列尾部
7: else if( on_front( $\sigma_k$ ) )
8: join( $e_{ij}, \sigma_k$ );
9: if( hasCoincidentEdge( $e_{ik}$ ) ) glue( $e_{ik}, e_{ki}$ );
10: if( hasCoincidentEdge( $e_{kj}$ ) ) glue( $e_{kj}, e_{jk}$ );
11: else
12: mark_as_boundary( $e_{ij}$ );

```

上述算法中,函数 hasCoincident() 用来判断有向环中的一条边在有向环中是否存在顶点相同但是方向相反的边。函数 findTheSmallestPoint() 用来找到与当前有向三角片以边  $e_{ij}$  相邻的、其法向与当前有向三角片的法向夹角小于  $2\pi/3$  的、外接圆半径最小的三角片中的第三个顶点。

### 2.2.2 join 和 glue 操作

执行 findTheSmallestPoint() 找到对应的  $\sigma_k$  后,如果  $\sigma_k$  不被包含在已有的三角网格中,则执行 join 操作,即将三角片  $(\sigma_i, \sigma_k, \sigma_j)$  加入到三角网格中去,且将边  $e_{ik}, e_{kj}$  插入有向环内,同时加入到队列中。

若  $\sigma_k$  是已经使用过的点,且不是在有向环上,则将边  $e_{ij}$  标记为边界边。

若  $\sigma_k$  在有向环之上,首先要查看如果加入三角片  $(\sigma_i, \sigma_k, \sigma_j)$  会不会产生非流形的情况,如果没有,则执行 join 操作。这个 join 操作执行完了之后可能会产生一对或者两对对偶边(顶点相同,方向相反的两条边),如果产生对偶边,则执行 glue 操作将这一对有向边从有向环中删去。

CPU E6550 @ 2.33GHz (2 CPUs) 的处理器,3.25GB 的内存,Microsoft Windows XP Professional(5.1)平台。用 SHRINK 算法和文中算法的效果做了比较,表1是相关数据和结果。可见文中算法的用时基本上和 SHRINK 算法在同一数量级上。

表1 模型参数及实验结果

模型	hzi	test_knot	dinosaur
顶点数	736	1280	23984
SHRINK 算法重建时间(s)	0.281	0.406	10.563
文中算法重建时间(s)	0.313	0.562	13.078

实验效果如图3~图5所示(以下图中左侧为原始点集,中间为本文算法的重建效果,右侧为 SHRINK 算法重建结果)。

由图3中看到,SHRINK 算法在重建开放曲面的时候效果并不是很理想,有孔洞存在,且还有些三角片构成了四面体,即有非流形的情况存在;文中的算法重建出的结果则没有这些问题,而且边界更光顺。

由图4可以看到,SHRINK 算法在重建时需要精确地估计孔洞系数以控制三角片的删除,当输入的孔洞系数在1.2左右的时候,其效果才能达到文中算法的效果;而文中算法输入的孔洞系数只要大于1都会得到相当好的重建效果。

图5表明了文中算法在大数据量模型的重构方面亦有相当好的效果。

## 4 结束语

文中提出了一种基于三维 Delaunay 三角剖分的区域增长式曲面重建算法。算法以外接圆半径最小作为挑选三角片的准则,重建出的曲面较其它算法有适用范围更广、人为参与更少、更好地保持了原始点集的几何特征等等优点。最后通过实验证明了文中算法的有效性。

三维曲面重建在计算机图形学中并不是一个新的研究课题,但是目前为止还没有通用的效果很好的重建算法。进一步的工作可以考虑将文中算法并行化,利用多核或者众核系统提高曲面重建速度。

### 参考文献:

- [1] Mencl R, Müller H. Interpolation and approximation of surfaces from three-dimensional scattered data points [R]. [s.l.]:Euro-graphics,1998:51-67.
- [2] Hoppe H, DeRose T, Duchamp T, et al. Surface reconstruction from unorganized points[J]. Comput. Graphics,1992,26(2):71-78.

## 3 实验结果及性能分析

文中实验的实验环境采用 Intel® Core™2 Duo



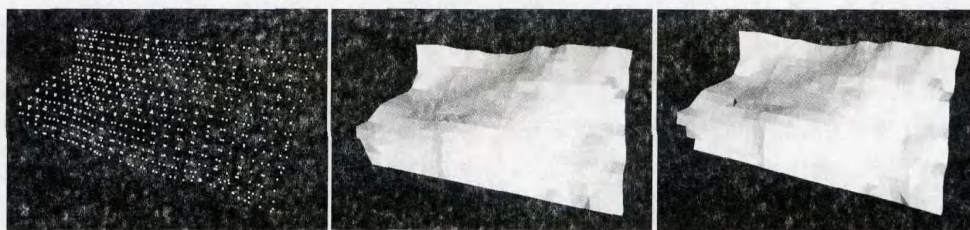


图 3 曲面点集 hz1 的重建结果(输入的孔洞系数为 2)

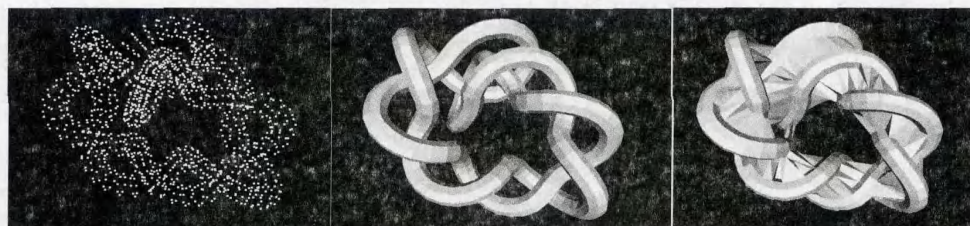


图 4 管道表面点集 test\_knot 的重建结果(输入的孔洞系数为 2)



图 5 恐龙模型表面点集 dinosaur 的重建结果(输入的孔洞系数为 3)

- [3] Lorensen W E, Cline H E. Marching Cubes: A high resolution 3D surface construction algorithm [J]. Computer Graphics, 1987, 21(4): 163-169.
- [4] Green P J, Sibson R. Computing Dirichlet tessellations in the plane [J]. The Computer Journal, 1978, 21(2): 168-173.
- [5] O'Rourke J. Computational Geometry in C[M]. 北京:机械工业出版社, 2005: 161-165.
- [6] Bowyer A. Computing Dirichlet tessellations [J]. The Computer Journal, 1981, 24(2): 162-166.
- [7] Watson D F. Computing the n-dimensional delaunay tessellation with application to Voronoi polytopes [J]. The Computer Journal, 1981, 24(2): 167-172.
- [8] Edelsbrunner H, Mücke E P. Three-dimensional alpha shapes [J]. ACM Transaction on Graphics, 1994, 13(1): 43-72.
- [9] Chaine R. A geometric convection approach of 3-D reconstruction [C]//Eurographics Symposium on Geometry Processing. [s. l.]: [s. n.], 2003.
- [10] Bernardini F, Mittleman J, Rushmeier H, et al. The Ball-Pivoting Algorithm for Surface Reconstruction [J]. IEEE Transactions on Visualization and Computer Graphics, 1999, 5(4): 349-359.
- [11] Amenta N, Bern M, Kamvysselis M. A new Voronoi-based surface reconstruction algorithm [C]//Proceeding of SIGGRAPH'98. Danvers: Assison-Wsley Publishing Company, 1992: 415-421.
- [12] Edelsbrunner H, Shah N R. Incremental topological flipping works for regular triangulations [J]. Algorithmica, 1996, 15(3): 223-241.
- [13] Lawson C L. Generation of a triangular grid with application to contour plotting [R]. California: Institute of Technology, Jet Pollution Laboratory, 1972.

(上接第 13 页)

- [8] Benkmann R, Kappeler Uwe-Philipp, Zweigle O, et al. Resolving Inconsistencies using Multi-agent Sensor Systems [C]//Proceedings of the 8th Conference on Autonomous Robot Systems and Competition: Robotica 08. Aveiro, Portugal: [s. n.], 2008: 93-98.
- [9] Riedmiller M, Merke A. Using Machine Learning Techniques in Complex Multi-Agent Domains [M]//Adaptivity and Learning. [s. l.]: Springer, 2003.
- [10] 季秀才, 崔连虎, 郑志强. 混合分布式任务分配机制在足球机器人系统中的应用研究 [J]. 计算机应用, 2008, 28(3): 706-709.
- [11] 陈万米, 张冰, 朱明. 智能足球机器人系统 [M]. 北京: 清华大学出版社, 2009: 120-142.
- [12] 宋鑫坤, 陈万米, 徐昱琳, 等. 非结构化场景下移动机器人 FastSLAM 应用研究 [J]. 计算机技术与发展, 2010, 20(2): 95-99.